Master Thesis Artificial Intelligence

Skewed Selective Acquisition: Sampling Bias in Active Learning and its Influence on Operational Classification Performance

Active Learning, Machine Learning, Class Imbalance, Sample Selection Bias, Classification

by

Alec Flesher-Clark

first supervisor

Georg Krempl

second supervisor

Arno Siebes

external supervision by

Paul Merkx
Jasper van Vliet
António Pereira Barata

Submitted to
UTRECHT UNIVERSITY
on September 23, 2022

For further questions contact:
Alec Flesher-Clark: a.c.flesher-clark@students.uu.nl, 1625985

Abstract

Active learning aims to provide supervised learning models with highly informative and succinct data, but can also introduce sampling bias as the resulting labelled dataset often no longer follows the original data distribution. Due to sampling bias, the training data no longer represents the environment that the model will operate in, which can either be harmful or helpful to classifier performance. One form of sampling bias is class bias, where the labelled dataset no longer follows the class distribution of the population data. Through experimenting on 15 different binary-classification datasets, this thesis studied active learning sampling bias through class bias and its relation to operational classification performance.

First, this thesis investigated four main factors in the active learning cycle that might influence active learning sampling bias in labelled training data. The chosen factors were the choice of active learning algorithm, the machine learning classifier, the level of class imbalance in the unlabelled data pool and the class ratio of the initial training set. All factors had an effect on sampling bias and on classifier performance, with varying degrees of severity. The level of class imbalance had the largest influence on active learning algorithms introducing more sampling bias.

Afterwards, experiments were conducted using three active learning debiasing methods: hierarchical sampling, QUerying Informative and Representative Instances (QUIRE) and Active Learning By Learning (ALBL). These query strategies were compared in terms of trained classifier performance and sampling bias mitigation. In these experiments, utilizing informative-based query strategies like uncertainty sampling led to the highest amount of sampling bias but also the highest performance. While the three debiasing methods resulted in less sampling bias, they were generally outperformed by uncertainty sampling. Of these methods, hierarchical sampling performed the best, achieving a performance which was marginally worse than uncertainty sampling. These results suggest that using active learning algorithms that introduce sampling bias can boost performance, especially in high class imbalance situations. However, careful consideration should be taken before implementing more bias introducing active learning algorithms. In cases where sampling bias in training data is harmful, either by raising issues of fairness or by yielding a shortsighted classifier, using a well-performing but more representative-based active learning method like hierarchical sampling or density-weighted sampling is recommended.

Contents

1 Introduction

Within the Dutch government, the Human Environment and Transport Inspectorate (ILT) is the supervising authority of the Ministry of Human Infrastructure and Water Management. The ILT aims to uphold environmental and transportation related laws in the Netherlands through inspecting cases which might be non-compliant to these laws. In recent years, they have created a research and development team called the Innovation and Data lab (IDlab). This IDlab seeks to apply machine learning and data science techniques to ILT problems, having ILT inspectors work together with models designed by the IDlab to move towards a more data- and risk-driven inspectorate. In practice, this means that ILT inspectors with assistance of IDlab created models select companies or objects to inspect for non-compliance from a target population. To apply different supervised learning techniques, the IDlab requires labelled data. Through various data sources, commercially gathered but also through much data gathered by the government, the ILT has access to a large amount of unlabelled data related to transport, infrastructure and human environment. However, there is significant cost to labelling this data, as the only ways of labelling ILT data is often through an actual inspection of a possible risk to environmental or transportation law or through inspectors' expert judgement of a label for a data entry through experience of similar historical data entries. When inspections are conducted, they are costly both in time and in resources, and are therefore predominantly applied to cases classified by inspectors as having high risk. To reduce the amount of labelled data required for training a supervised model, the IDlab have begun to apply active learning methods to ILT cases. Active learning determines the datapoints which have the highest information value and queries an oracle [1], a human user, to label these instances through inspections.

Through deploying active learning in a supervised classification data pipeline, the IDlab is faced with a choice between information value-based and risk-based inspections. Information value-based inspections can be seen as inspections purely conducted with the aim of improving the data fed to the machine learning classifier, thereby improving the classifiers performance. Risk-based inspections are then conducted when the classifier determines a possible violation. Using machine learning and active learning, information value-based inspections are conducted by active learning techniques aiming to find the most informative instances to inspect. A machine learning model is trained on the resulting labelled dataset, and is then applied to predict non-compliant individuals and aid with risk-based inspections. Both types of inspections ultimately contribute to improving the classification performance of both information value-based and risk-based inspections by acquiring new labelled data. The difficulty lies in the level of priority each type of inspection should be given at a certain time when a model is deployed. With the cost of inspections, risk-based inspections are conducted much more often by the ILT. In recent years, the IDlab aims to introduce active learning methods in the information-based targeting to ensure that data instances with the highest information value are inspected and labelled accordingly. Currently, the IDlab has applied active learning to two domains, a webscrape pipeline for monitoring online trading through text classification and a pipeline for classifying ship waste dumping in the North Sea.

1.1 Problem Description

When applying active learning to the ship waste discharging pipeline [2], the various active learning algorithms exhibited interesting behaviour. To properly describe this behaviour and to motivate the focus of this research, first a brief overview of the detection of ship waste dumping is required. The IDlab aims to classify ships that are practicing the activity of ”zeezwaaien”, a Dutch term which represents a ship discharging of waste and various chemical residues in open sea. The practice of zeezwaaien is regulated by law in the second appendix of the International Convention for the Prevention of Pollution from Ships (MARPOL) agreement [3]. Previous research at the IDlab [4] introduced an anomaly classification model to detect which ships practice zeezwaaien. The research was done as a master’s thesis and implemented a supervised machine learning model, trained on vessel Automatic Identification System (AIS) data, which classifies zeezwaaien en route for tanker ships. The aim was to classify ships en route as inspectors could then possibly intervene if a risk of zeezwaaien is detected. Ships performing so-called looping trajectories, where ships sail out from and return to the same port, were found to be more likely to discharge ship waste. Looping trajectories were found by experts to be the highest indicator of zeezwaaien behaviour, as ships would often only go out to open waters to discharge ship waste and then return to the same port. The model was trained on different distributions of the labelled dataset, all of which exhibited high class imbalance, as around 2% was labelled as zeezwaaien behaviour and the other 98% as normal behaviour.

Experts can estimate labels for any machine learning problem relatively inexpensively. However, labelling through expert judgement can sometimes introduce social problems, where some groups are discriminated against through human sample selection bias. The ILT wants to acquire labels through actual inspections, making use of a labelling budget every year. Because the cost of actual inspections is high, the ILT has a desire for machine learning classifiers to have high performance in categorizing anomalous behaviour in different cases while being limited to few labelled data instances. Active learning seems to be an appropriate solution to this problem. Therefore, the IDlab has researched the implementation of various active learning methods in the zeezwaaien detection pipeline [2]. Most of the implemented methods resulted in a significant improvement in the machine learning model’s learning rate when compared to randomly sampling to create the training data.

However, when looking at the distribution of the two classes in the data sampled by active learning techniques, instances of the minority class were queried almost exclusively during the first 10 query iteration, and much more frequently during the first 25. During these first query iterations, because of the prevalence of the minority class, the information value-based method of active learning mimics the risk-based targeting of the minority class in anomaly detection. When the active learning algorithms had queried 50 instances, most methods would result in a labelled set with 40% zeezwaaien instances and 60% normal behaviour instances. Looking at the evolution of the label distribution over time during the application of the different active learning query strategies, it is evident that the active learning algorithms introduce a form of sample selection or sampling bias to the sampled dataset. We refer to the distribution of the labels as an introduced sampling bias because the dataset no longer follows the 98 : 2 class distribution of the population. This poses a potential problem to machine learning models trained using biased datasets, as the machine learning classifiers may have difficulty when applied to unbiased real life situations. Sampling bias in labelled datasets can also negatively impact the fairness of the machine learning classifier trained on it. In some cases where human data is used, sampling bias in the labelled dataset can lead to discrimination of certain social groups. However, in high class imbalance cases like the zeezwaaien research, through sampling bias in the labelled dataset the resulting data distribution of the labelled dataset is more balanced and seems to leverage the extreme class imbalance in the population data distribution. These findings raise questions regarding the role of sampling bias in active learning pipelines and their influence on the classification performance of machine learning models trained with these biased datasets. Various sampling bias mitigation methods in active learning exist, but the question is whether the tradeoff between the representativeness in the sampled dataset and the performance of the machine learning classifier is too severe.

Therefore, the problem this research will explore is the role of active learning sampling bias and various sampling bias mitigation techniques on classification performance of operational machine learning classifiers. It will study sampling bias through looking at class bias in the training data: the difference in class distribution between the labelled dataset and the original data distribution. After giving an overview of the background in section 2 and related work within the field of sampling bias and active learning in section 2.6.3, an exploratory study of the various factors that influence sampling bias will be conducted. Following this study, a comparison of various active learning sampling bias mitigation algorithms will be given. Experiments on these specific algorithms will show whether mitigating sampling bias in the training data through active learning influences the learning rate and classification performance of machine learning classifiers.

1.2 Research Questions

Given the potential problems active learning sampling bias may pose for machine learning models, and the need to explore the influence of various sampling bias mitigation techniques on operational classification performance, this thesis aims to answer the following main research question through answering the various accompanied subquestions:

Given varying degrees of class imbalance in the data distribution, how do various Active Learning sampling bias mitigation techniques influence operational classification performance?

- What factors in the active learning cycle influence active learning sampling bias in the labelled dataset?
- How do different active learning sampling bias mitigation techniques compare when looking at operational classification performance?
- Will the influence of active learning sampling bias mitigation techniques yield similar results on the same dataset with varying degrees of class imbalance?

2 Background

This section provides a literary background for machine learning, active learning, bias and bias mitigation. A short description of machine learning and the various machine learning techniques is given in section 2.1. Afterwards, section 2.2 provides an overview of the general active learning pipeline and how active learning and machine learning interact in querying and classifying data instances. In order to understand the sampling bias problem and the distinction between class imbalance, sampling bias in active learning query strategies and sampling bias in human sampling, section 2.3 aims to clarify these phenomena and where they occur in an active learning pipeline. After describing sampling bias in the active learning pipeline, a more in depth overview of all specific types of active learning querying strategies is given in the following section 2.4. Section 2.5 discusses the literature on the various existing active learning bias reduction techniques. Finally, section 2.6 describes the various existing and relevant evaluation metrics for machine learning and active learning methods.

2.1 Machine Learning

Machine learning is a subset of the field of Artificial Intelligence which concerns itself with the idea that computer algorithms can learn and improve from experience and can apply their experience to new cases, somewhat mimicking human behaviour. A general overview of machine learning is given by Tom M. Mitchell in his book on the subject [5]. This section will briefly discuss the different types of machine learning and will offer a more in depth description of the type of machine learning used for this thesis, namely supervised machine learning.

In machine learning, algorithms are trained on the datasets they receive as input in order to learn patterns in the data. When faced with new data, machine learning algorithms use the learned patterns to predict the form of the new data. An illustrative example is the application of machine learning in categorizing data into classes, called *classification*. When you would like to predict whether an email in your mailbox is either spam or not-spam, a machine learning classifier can be trained on data instances with various features representing emails and labels containing their class, either spam or not-spam. Specifically, this is a case of binary classification as there are only two classes of data which the machine learning algorithm must distinguish between. Through training, the algorithm can then be applied to new unlabelled instances to predict whether the new instance is spam or not spam. A machine learning classifier does so by learning a border called the *decision boundary* in the feature space of its input dataset. All new instances to one side of the decision boundary are then classified as spam while instances on the other side are classified as non-spam.

The spam classification example is part of machine learning's subfield of *supervised learning*. Supervised learning is concerned with training machine learning models on labelled data and is one of the most popular applications of machine learning. Various research has been conducted on the application of supervised learning in different fields, such as cancer prognosis and detection [6], text classification [7] and the aforementioned detection of spam [8]. While supervised machine learning has been proven to work well in a multitude of domains, supervised models are highly reliant on the quality of data they receive. They also require all data instances of training and test data to be labelled with their corresponding class, which is usually done by a human annotator. The process of labelling thousands of instances is a tedious process, which active learning (see section 2.2) seeks to diminish. Specifically, this thesis will focus on the application of active learning to reduce the amount of labelled data required for training well-performing machine learning binary classifiers, as the zeezwaaien case researched by the IDlab is also a binary classification problem.

Machine learning can also be applied to unlabelled data, in which case it is referred to as *unsupervised learning*. In these problems, the goal of machine learning algorithms is to analyze and organize the patterns of the input data. The input data can then be transformed to group various types of data into clusters [9], in order to predict in which cluster new data belongs based on its features. Some active learning algorithms make use of clustering and other unsupervised learning techniques. Some of these algorithms are described in section 2.5.

The third and final main form of machine learning is called *reinforcement learning.* The main idea of reinforcement learning is that a reinforcement learning algorithm, sometimes referred to as an agent, acts according to certain predetermined rewards. The agent will aim to maximize its received reward over time by learning from experience through the various actions it can perform. An example of an intelligent agent using reinforcement is of an agent traversing through a maze. The agent can take actions to move through the maze and must maximize the reward of making it through the maze as quickly as possible. Over iterations of traversing the maze, the agent will learn the most optimal path to reach its maximal reward. There exist some parallels in active learning techniques and reinforcement learning, particularly the concept of exploration and exploitation. For a definition of these terms and how they relate both to reinforcement learning and active learning, see section 2.5.2.

2.2 Active Learning

As illustrated in the previous section, supervised machine learning requires labelled data. However, as is the case in the IDlab's machine learning application for ILT related problems, labelling data can be a costly venture, both in time and resources. Active learning, a subfield of machine learning, seeks to alleviate this cost. Active learning aims to achieve high machine learning classifier performance with fewer training labels required by letting the active learning algorithm choose which instances in the data provide the greatest contribution to the learning rate of the machine learning classifier and should therefore be labelled and added to the training dataset. Burr Settles [1] offers a succinct and prefatory overview of active learning, various active learning scenarios and the main active learning techniques. This section will introduce the main concepts of active learning, the goal of implementing active learning in a machine learning prediction problem, as well as the general active learning pipeline. Figure 1 depicts a schematic for a typical data pipeline using active learning for determining which data instances should be labelled.

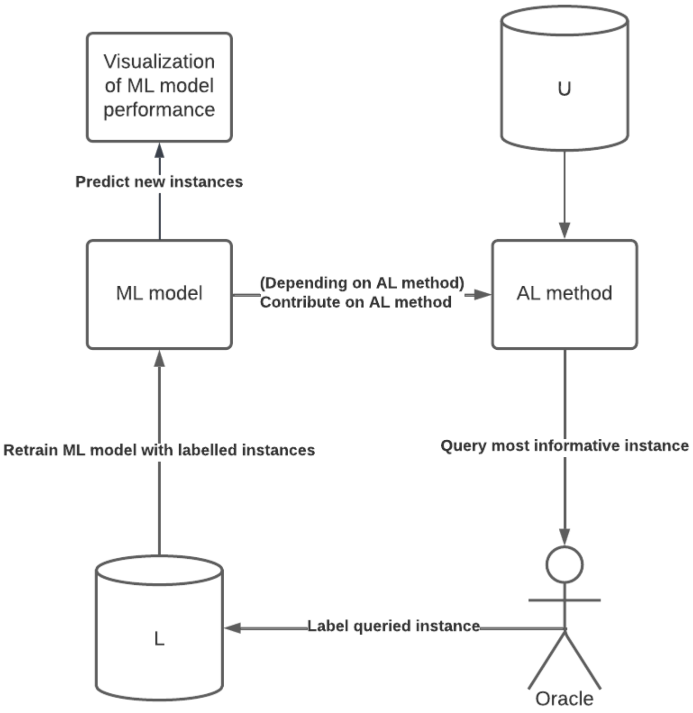


Figure 1: Standard pipeline for implementing active learning (AL) in data acquisition for Machine Learning (ML) prediction,

The pipeline starts by providing an active learning method with unlabelled data, in figure 1 referred to as U. Depending on the problem, U might be preprocessed or transformed in order to remove redundant features, empty fields in the data or other preprocessing steps will be necessary. The active learning method or *query strategy*, often aided by a pretrained machine learning model, aims to locate the most informative data instances in the unlabelled data it receives as input. The instance with the highest determined utility score is then provided to a human annotator called an *oracle*, who determines which label belongs to each instance through their expertise. The resulting labels are provided to the labelled L, which is used as the training dataset for the machine learning model. L can also be used as the test set for evaluating the performance of the machine learning model, but care must be taken to separate L into training and test datasets in order to prevent the machine learning model from overfitting to the data. Iteratively, the active learning query strategy selects more data to be labelled, until a certain predefined stopping criteria or *labelling budget*, a certain amount of labels, is met. Throughout the labelling process, a *learning curve* is generated to keep track of the machine learning model's performance over query iterations. The chosen evaluation metric varies depending on the type of machine learning problem, a more detailed overview of which is given in 2.6.

There are three main data-related scenarios of the active learning pipeline:

- **Pool-based sampling** is the scenario when U consists of a large pool of unlabelled instances. The chosen query strategy will have to evaluate the entire pool before selecting the best query. This form of active learning will be utilized for researching active learning sampling bias.
- **Stream-based sampling** is a form of active learning when unlabelled data instances arrive as input for the query strategy one at a time from a data stream. The query strategy then decides whether to query or discard the instance.
- **Membership query synthesis** is the last active learning scenario. In membership query synthesis, the active learning algorithm may request labels from the oracle from any unlabelled instance in the input space. The learner may also request labels from instances the learner generates, rather than those sampled from some underlying natural distribution.

Active learning has seen a variety of applications. Zhou et al. [10] researched the use active learning and semi-supervised learning for content-based image retrieval, a case where labelling instances is ordinarily time-consuming. They found that implementing active learning improved the image retrieval performance. The field of text classification has also seen widespread implementation of active learning techniques, as the manual annotation of text corpora is a tedious and time-consuming process. The field has seen successful implementations of active learning in e.g. part-of-speech tagging [11][12] and parsing [13]. However, a survey conducted by Tomanek and Olson [14] sheds light on the difficulties of practical implementation of active learning. The survey was conducted using participants who were involved in text annotation intended for machine learning for a myriad of natural language processing (NLP) tasks. The results of the survey indicated that 20% of participants had ever used active learning for NLP tasks, as most participants were sceptical that active learning would reduce the overall annotation time, as there currently is no concrete consensus on this matter. Settles offers a critical look into the problems faced in the practical applications of active learning techniques [15], but concludes with an optimistic view of the direction active learning is heading.

This thesis aims to contribute to the research on active learning in practice, by researching one of it's difficulties: sampling bias. As sampling bias in active learning is at the forefront of this thesis, the following section will formalize and distinguish sampling bias from class imbalance in active learning. Since the query strategy is the core of active learning, the next section will provide an overview of the different active learning query strategies.

2.3 Class Imbalance and Sample Selection Bias

Within an active learning pipeline, both class imbalance and sample selection bias, or sampling bias might occur. *Class imbalance* occurs when one class is overrepresented in the data, referred to as the majority class. The other underrepresented class, or classes in multiclass classification, is called the minority class.

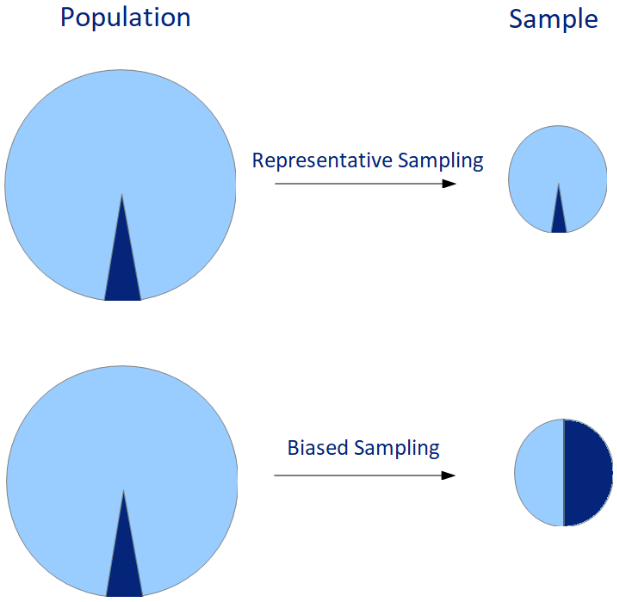


Figure 2: Difference between representative sampling and sampling which introduces sampling bias through class bias. Colours are the two classes in a binary classification dataset.

Sampling bias, as seen in figure 2, is the phenomenon when data taken from some population does not follow the same distribution as the population. In figure 2, even though the sampled dataset no longer contains class imbalance, the dataset does contain sampling bias because its class distribution is different from the population data. This form of sampling bias is called *class bias*, which is the form of sampling bias this thesis will study. The distinction between multiple types of sampling bias is given in section 2.3.3. The rest of this section is structured as follows. First, more detailed definitions of class imbalance and sample selection bias, introduced by both humans and active learning algorithms, will be given using an example of an active learning pipeline. Afterwards, the effect of class imbalance and sampling bias on machine learning classifiers will be given. After providing the distinction between class imbalance and sampling bias, this section will summarize prior research on the effect of sampling bias on machine learning classifiers.

Figure 3 shows a more detailed and realistic active learning pipeline where the data is preprocessed before being fed to the main active learning cycle. When comparing this figure to figure 1, the difference lies in the steps taken before feeding unlabelled data to the active learning cycle. These steps include the preprocessing of the population data pool and the selection of labelled instances to pretrain a machine learning classifier.

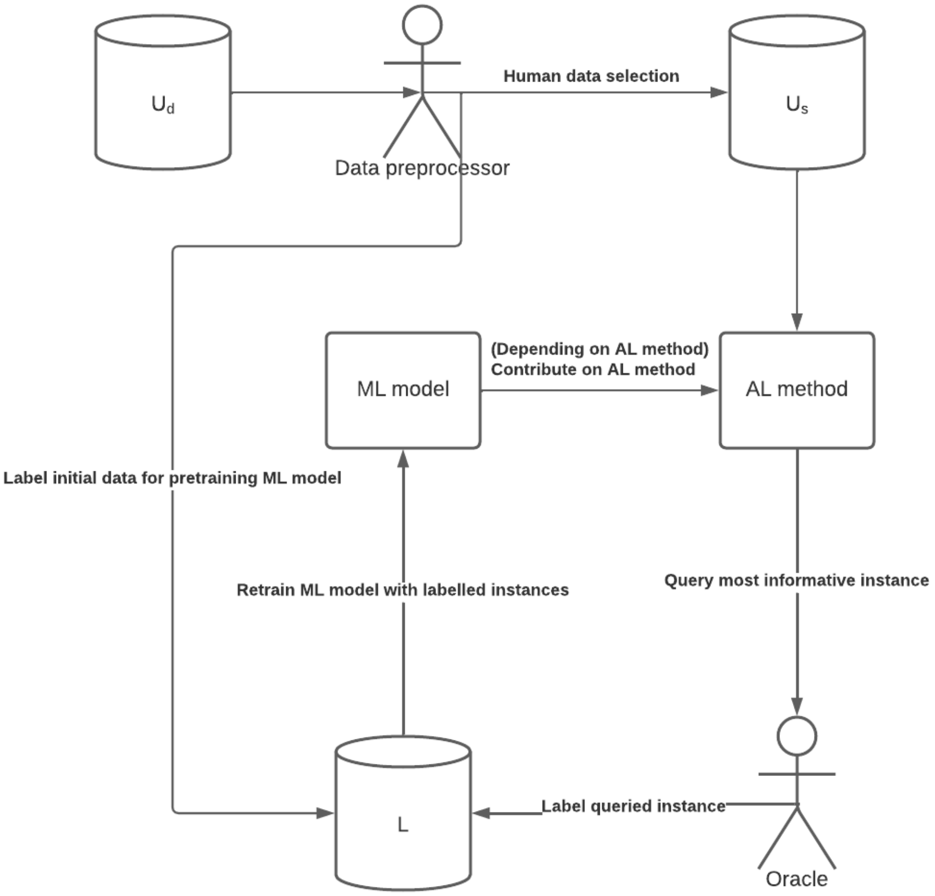


Figure 3: Class imbalance and sampling bias in an active learning (AL) and machine learning (ML) classification pipeline.

2.3.1 Class Imbalance

In the pipeline, *class imbalance* can occur in all three different datasets: U_d, U_s and L. The level of class imbalance can vary depending on the type of data problem. In the case of the supervised spam detection example in section 2.1, the non-spam class is the majority class, as it has many more instances than the spam class, the minority class. If a dataset has high class imbalance, it is analogous to having a low class ratio. Class ratio is defined as:

$$class\ ratio = \#_{MinorityClass}/(\#_{Minority\ Class} + \#_{Majority\ Class}) \tag{1}$$

Perfectly balanced class balance in binary classification is seen as a dataset with a 50 : 50 or 0.5 class ratio. In figure 3, all datasets might exhibit a different class ratio, thus some datasets will have more class imbalance than others. Class imbalance in L can influence the machine learning classifier trained on this dataset. A survey on this impact was conducted by Japkowitz and Stephen [16]. Similarily, class imbalance in U_s can have an impact on active learning querying [17]. Various active learning techniques exist to address class imbalance in the U_s dataset, such as oversampling the minority class through an active learning algorithm called VIRTUAL [18]. More detail of these techniques is given in the related work section, subsection 3.3.

2.3.2 Human Sampling Bias

To understand why U_s might have a differing class imbalance ratio than U_d, knowledge of data and data preprocessing is necessary. In many machine learning and active learning cases, the original data pool might be unstructured, contain missing values or have redundant features. Specifically for ILT related cases, inspectors might take subsets of the data based on their experience with and recognition of non-compliant data instances. ILT inspectors might also have to make choices due to a lack of available data. An example of this might be that ILT inspectors are forced through lack of data to exclude certain ships in their dataset, making the resulting classifier biased as the classifier is trained on data that does not represent the real world situation. This problem of human sample selection bias cannot be fixed through active learning methods, but through operational choices in data selection. Before feeding data to a machine learning classifier, a developer might want to preprocess the data. In figure 3, the data preprocessor alters the original data pool U_d to create a new unlabelled pool U_s and to initialize the labelled dataset L. U_s might have different data and distributions if the preprocessor decides to resample the data in U_d i.e. by undersampling instances of the majority class or oversampling the minority class. L might also have a differing data and class distribution from U_d, as the initial labelled data is a smaller subset (usually around $10 - 20$ instances) of the original data. Only if the data preprocessor has taken measures to ensure that L initially is representative of the original distribution of U_d, will the two distributions be the same. If a dataset, like U_s or L, has a data distribution which, through manipulation of a human preprocessor, differs from the original data distribution, the dataset is said to exhibit *(human) sample selection bias* or *human sampling bias.* This thesis will refer to this phenomenon as human sampling bias and the sampling bias introduced by active learning algorithms simply as sampling bias, to distinguish the two and to avoid further confusion.

2.3.3 Sampling Bias through Active Learning

In figure 3, the chosen active learning method can also influence the distribution of the labelled dataset L. Similar to human sampling bias, the process of active learning selective acquisition can introduce its own form of *sampling bias* [19]. As active learning methods pose queries of unlabelled data instances the method deems to be the most informative to the oracle, the distribution of the resulting labelled dataset

L might not be representative of the distribution of the population distribution U_d. As will be discussed in section 2.4, especially information-based query strategies are prone to introduce sampling bias as they can repeatedly select redundant examples and outliers in the feature space [20]. Ever since the first active learning algorithms were introduced, the concept of sampling bias has been noted but never truly formalized. Farquhar et al. formalize active learning sampling bias in machine learning classification [21], which they refer to as statistical bias. In order to provide clarity on sampling bias and its effect on machine learning classifier optimization, a brief summary of their formalization is required:

In supervised learning, classifiers aim to find a decision rule which, given a predetermined loss function, minimizes the *population risk*. The population risk can be seen as the expectation over a continuous loss function. This risk cannot be calculated, but estimated through looking at the empirical distribution for some dataset of N points drawn from the population. This gives the unbiased and consistent estimator of the population risk which is called the *empirical risk*:

$$\hat{R} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(y_n, f_\theta(x_n)) \tag{2}$$

Here, $\mathcal{L}(y_m, f_\theta(x_n)$ is the loss function using the class labels y_n and the outcome of the decision rule learned by the classifier $f_\theta(x_n)$. However, once an active learning algorithm is applied to the population, we cannot evaluate $\hat{R}$ directly because not all N datapoints are labelled, only a labelled subset L. We then evaluate the *sub-sample empirical risk*:

$$\bar{R} = \frac{1}{L} \sum_{l=1}^{L} \mathcal{L}(y_l, f_\theta(x_l)) \tag{3}$$

This estimator is biased as the L instances are not drawn independent and identically distributed from the population distribution. Because active learning algorithms use equation 3 for estimating the population risk, active learning algorithms suffer from sampling bias. Only when L is sampled according to the population distribution U_d is this estimator unbiased.

Sampling bias through active learning can be further subdivided into two types [22]. *Class bias* (see figure 2) occurs when the class distribution of the labelled training set is disproportionate to the original class distribution. *Feature bias* occurs when outliers and non-representative instances of the feature space are repeatedly selected by active learning methods. This means that classifiers trained on this data will have a limited and inaccurate view of the feature space of the instances, which can lead to accuracy problems of the classifier. This thesis will strictly be focused on class bias through active learning selective acquisition and will refer to class bias when referring to sampling bias, section 5.4 of the methodology gives the full reasoning behind this decision. The full effects and situations in which sampling bias can harm or help classifier performance are given in the next section 2.3.4. Apart from the choice of query strategy, there is a lack of clarification on which factors influence sampling bias through active learning. This thesis will therefore aim to bring new insights into these factors through empirical experimentation.

2.3.4 The Effects of Sampling Bias on Machine Learning Classifiers

Depending on the situation, sampling bias in the labelled dataset can either be harmful or helpful to the performance of the machine learning classifier [21]. This section will discuss both sides of sampling bias, while illustrating the positive and negative effect of this type of bias through examples.

Sampling bias in the training data can sometimes have *positive* effects on classifier learning. Sampling bias in the training data can lead to an improvement classification performance, given class imbalance in the original dataset [23]. In high class imbalance situations such as the IDlab study on zeezwaaien [2], active learning introduced sampling bias by oversampling the minority (or positive) class with respect to the original class distribution. This resulted in an increase in overall classification performance in terms of recall and precision (see section 2.6.1 for a description of these metrics), when compared to learning a classifier under the original imbalanced class distribution. In studies under which the goal is to correctly classify minority class instances, sampling bias through active learning can, therefore, be advantageous by providing a more balanced labelled dataset for training.

However, sampling bias can sometimes have *negative* effects towards learning classifiers. While it may be beneficial to oversample the minority class towards generating a more balanced training set, the choice of active learning algorithm, one of the factors which can produce sampling bias, impacts the distinct regions of the feature space which are targeted for sampling. This leads to scenarios in which the majority class sample is not independent and identically distributed with respect to the original *feature* distribution: the classifier is learned on a feature-biased sample which compromises the generalisation of the classification performance [24]. Putting it differently, the training data no longer represents the actual environment in which models learned from this data are meant to operate. To make it explicit, consider the following operational example. The IDlab wants to solve the problem of detecting which Netherlands-based companies are non-compliant to an environmental law, such as proper company waste disposal. In actuality, around 5% of companies are non-compliant to the law, whereas through using active learning a dataset of 50 instances is created in which around 40% of companies are non-compliant. Now, the sampled dataset no longer accurately reflects the actual situation. In addition, the queried minority class instances might be outliers in the data and are therefore not representative of the data distribution as a whole. The trained classifier might now predict non-compliance more often when in actuality there is no non-compliance, resulting in a loss of precision. In ILT inspection situations such as this one, the cost of misclassifying a compliant company as a non-compliant one is high, as it will waste already scarce inspection resources.

Another possible problem caused by sampling bias pertains to fairness in machine learning classification. When machine learning is applied to human data, certain features in the data can inadvertently cause the classifier to discriminate against certain groups. Sampling bias in the training data can either aggravate biases in the original data or create new biases through feature subset selection bias. This can be done e.g. by giving certain basic human features more weight while in actuality the problem is more complex. The problem of determining whether people are more likely to commit fraud is an example. When the population is sampled, either by humans or through active learning, it so happens that all people in the sampled dataset who commit tax fraud are of a certain ethnicity. Now the model trained on this data will conclude that all people of that ethnicity are more likely to commit tax fraud. Therefore, bias reduction is therefore often linked to improving fairness in machine learning predictions. Sampling bias is one of many different types of bias which can be present in machine learning tasks, an overview of all other forms of bias and various fairness improvement techniques is given by Mehrabi et al. [25].

To see what effects human sampling bias has on machine learning classifiers, Zadrozny [26] evaluated multiple types of supervised classifiers with and without sampling bias. Zadrozny concluded that what he terms only global classifiers, whose output depends on the distribution of the entire input space, are affected by sampling bias in their training data. Local classifiers, which are not dependent on the distribution of the input space, are not affected. Examples of global classifiers are decision tree learners, naive Bayes and soft margin SVM. Logistic regression, hard margin SVM and K-nearest neighbours are examples of local classifiers. Oommen et al. [27] experiment with varying degrees of class imbalance and sampling bias in training a maximum-likelihood logistic regression (MLLR) [28] classifier. Their conclusions bolster Zadrozny's conclusion by suggesting that MLLR, a global classifier, is affected by sampling bias. Adhering to these conclusions, this thesis will study the effects of debiasing active learning sampling bias on two global machine learning classifiers, one decision tree learner and a MLLR classifier.

When sampling bias is introduced to a training dataset, there are multiple methods to mitigate this form of bias. Recent years have seen many publications on sampling bias mitigation in machine learning, such as a classifier that adapts to levels of sample selection bias [29]. A detailed overview on sample bias correction is provided by Cortes et al. [30]. In contrast to reducing the effect of sampling bias during the training of a machine learning classifier, this thesis will study the effect of reducing sampling bias on machine learning classifiers by using certain active learning algorithms. If sampling bias is reduced in the labelled dataset through querying, no measures to mitigate this bias will be necessary at training time.

2.4 Active Learning Query Strategies

In the pool-based active learning scenario, there are multiple query strategies one can consider to use. Kumar and Gupta provide a survey on query strategies in this setting and divides these strategies into three main types [31]. Through this categorization, active learning query strategies can easily be distinguished from another. The main types are:

- **Information-based methods** aim to find the most informative instance in the unlabelled data pool. These methods can do so by looking at the uncertainty a machine learning classifier has when classifying these instances [32]. The instances that are closest to the classifier's learned decision boundary are seen as the most informative. The benefit of using query strategies under this category are that their choices for queries are quite intuitive and understandable. The main disadvantage of information-based methods is that they do not account for relationships among individual instances and that they consider each instance as independent and identically distributed (i.i.d.). Therefore, the original distribution of instances is not represented in the choices of information-based methods, which can lead to sampling bias.

- **Representative-based methods** do look at the overall structure of the data when determining which instance to query to the oracle. These methods pick instances which best represents the input feature space. Depending on the method, different measures are chosen for representativeness, such as distance. In doing so, these methods address one of the main shortcomings of information-based methods and are less sensitive to sampling bias. However, these methods may require more queries in order to achieve the target decision boundary. Therefore, these methods converge slower to optimal machine learning performance than informative-based methods.

- **Informative- and Representative-based or hybrid methods** combine the two aforementioned types with the aim of addressing both their shortcomings. By combining the two, these methods aim to find the most instances which are the most informative but also those that represent the underlying data distribution. However, there exists a trade-off between these two goals. Hybrid methods of informative- and representative-based methods must optimize this trade-off through creating a balance of information-based and representative-based searching through the input feature space.

- **Other methods** are harder to categorize as there are many other ways to determine which instance to query. Some of these differing query strategies will be discussed in the following sections, like reinforcement learning inspired approaches 2.5.2 and querying instances based on the expected reduction of the generalization error 2.4.6.

The following subsections describe different noteworthy query strategies belonging to the described four main categories. The query strategies will be described in detail, as understanding the different perspectives on applying active learning is key to understanding which strategy is useful depending on the situation. Figure 4 shows a sketch of the different querying behaviour of informative-based, representative-based and hybrid methods. In the figure, which is inspired by a figure Kumar and Gupta's active learning survey [1], a clear distinction between the behaviour can be seen. To summarize, informative-based query strategies query instances closest to the decision boundary and can therefore classify outliers like the instance on the far right of the feature space, which leads to sampling bias. Representative-based methods query instances which represent a group of instances the best and hybrid methods combine both information value and representativeness when making selections.

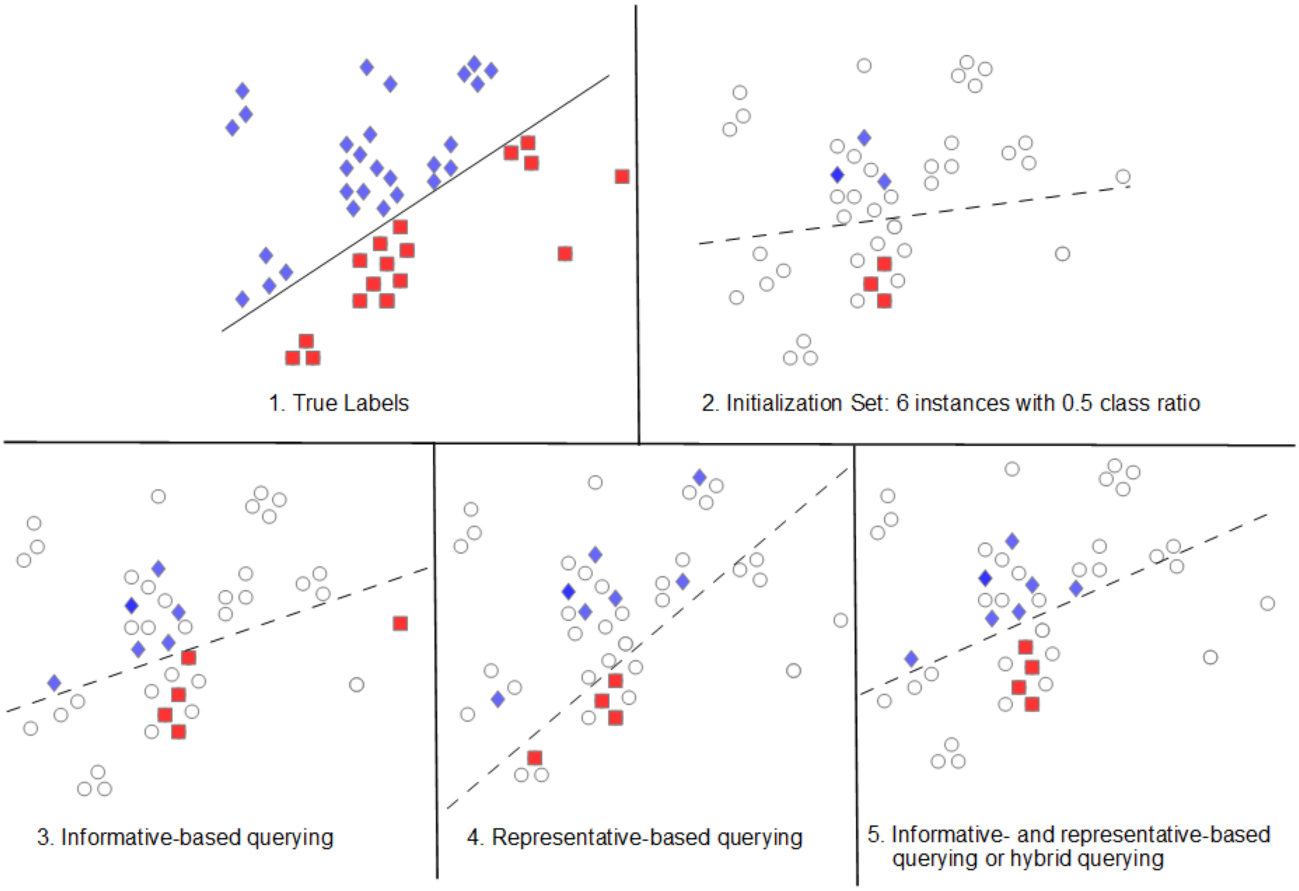

1. True Labels
2. Initialization Set: 6 instances with 0.5 class ratio
3. Informative-based querying
4. Representative-based querying
5. Informative- and representative-based querying or hybrid querying

Figure 4: Differences in querying behaviour between different query strategies. All different methods make 5 queries. Informative-based query strategies find the outlier instance on the far right the most informative because of its location on the decision boundary. This behaviour can result in sampling bias.

2.4.1 Uncertainty Sampling

Since since its introduction in 1994 by Lewis and Gale [32], uncertainty sampling has become one of the most popular query strategies. It is the quintessential information-based method, as it aims to query the most informative instances through finding the instances that a probabilistic machine learning classifier is most uncertain about. For binary classification, uncertainty sampling using least confidence as the uncertainty measure can be formalized by the following equation:

$$x^*_{LC} = argmax(1 - P_\theta(\hat{y}|x)) \tag{4}$$

Where x^*_{LC} is the instance the classifier is least confident about and and $\hat{y} = argmaxP_\theta(y|x)$, the label with the highest posterior probability under the machine learning model θ. In the binary classification case, instances are selected that have a posterior probability closest to 0.5 for being classified as the positive class. Being an information-based query strategy, uncertainty sampling can introduce sampling bias in the labelled dataset. This makes uncertainty sampling an interesting query strategy to compare to other query strategies whose aim is to mitigate sampling bias.

2.4.2 Query by Committee

Query by Committee (QBC) [33] is a more refined and theoretically-motivated information-based query strategy. The main idea of QBC is that a committee of machine learning models is maintained which are all trained on the labelled dataset L. The difference between the models lies in that each model has its own hypothesis based on the labelled data it has seen so far. These hypotheses are represented by different decision boundaries in the labelled data feature space. For maximum disagreement sampling, the instance which is considered the most informative is the instance that the different committee members disagree on the most. Therefore, it is important to have some measure of disagreement between the committee members.

The original idea for QBC sampling was introduced by Seung et al. [33], where they used the algorithm to evaluate the performance of perceptron learners. In the following years, many applications and modifications were conceived, such as the application of QBC for text categorization [34]. Modifications have been introduced to the measuring of the level of disagreement by Dagan and Engelson [35], who are first to implement QBC for efficiently training probabilistic classifiers and use vote entropy to measure the level of disagreement between committee members. Another popular measurement of disagreement is the use of Kullback-Leibler divergence [36], first proposed for use in QBC by McCallum and Nigam [37]. Another influential publication in QBC research is that of Abe and Mamitsuka [38]. Since QBC uses a committee of models, it can be treated as an ensemble method. Therefore Abe and Mamitsuka use the well-known ensemble methods of bagging and boosting [39] [40] to construct committees.

QBC sampling, through having a committee of models vote on the most informative instance, aims to select more informative instances than the single model uncertainty sampling approach. However, also being an information-based query strategy, QBC is as myopic as uncertainty sampling and can introduce heavy sampling bias. Uncertainty and QBC sampling are prone to outliers in the data, if these outliers are deemed the most informative or closest to the decision boundary by the query strategy.

2.4.3 Density-weighted Sampling

Unlike uncertainty and QBC sampling, density-weighted sampling considers the entire input feature space instead of looking at instances most near decision boundaries. However, it still makes use of an informative-based strategy like uncertainty sampling, before applying a density weight to the utility of instances. This makes density-weighted sampling less sensitive to outliers in the input data and an example of a informative- and representative-based query strategy or hybrid query strategy. A general density-weighted framework is described by Settles and Craven [41]. The key concept of density weighing is that instances with the highest utility are not only the instances the machine learning model is most uncertain about, but are also representative i.e. inhabit dense spaces of the input space. The method of determining the utility of instances is then a combination of an uncertainty measure and a measure of similarity to other instances, like a distance measure used in [42]. Using distance as a density weighing measure, the distance of one instance to all other instances is summed. The instance that has the shortest total distance is then seen as the most representative of the data distribution. However, in high class imbalance situations, using distance as a representativeness measure might not be ideal. This is due to the fact that the minority class is represented less and therefore is less densely available in the data. Other density-based approaches to sampling use clustering to select representative instances. These methods, like hierarchical sampling [24], density clustering in a stream-based active learning scenario [43] and active learning using density sampling [44] will be discussed in section 2.5.1.

In taking the spatial relationships of the input feature space into account, density-weighted sampling approaches query instances which adhere more closely to the underlying distribution of the unlabelled data. Consequently, density-weighted and other density-based approaches introduce less sampling bias in the labelled dataset than the myopic, information-based approaches like uncertainty sampling and QBC sampling. As density-weighted, QBC and uncertainty sampling are the three most popular forms of querying instances, the behaviour of these methods will be studied in various settings, to determine which factors can influence sampling bias.

2.4.4 Expected Error Reduction

Expected error reduction [45] is a decision-theoretic approach to instance querying where the goal is to reduce the generalization error of the machine learning model as much as possible. The expected future error of the model is estimated on the remaining unlabelled instances before querying the instance with the lowest *risk* i.e. the minimal expected future error. When using 0/1-loss for evaluation, the choice of instance to query would then be according to the following formula:

$$x^*_{0/1} = argmin(\sum_i P_\theta(y_i|x) * (\sum_{u=1}^{U} 1 - P_{\theta^{+(x,y_i)}}(\hat{y}|x^{(u)}))) \tag{5}$$

Here, $\theta^{+(x,y_i)}$ denotes the new model after it has been retrained with the new training tuple (x, y_i) added to the labelled training dataset. However, when estimating the error, we do not know the true label for each unlabelled instance yet. This means the expectation over all possible labels under the current model is approximated. Expected error reduction depends heavily on the objective function in which the minimal risk is to be found.

Expected error reduction techniques have advantages in being near-optimal and independent of the class of machine learning model. However, expected error reduction techniques are the most computationally expensive query frameworks. This is due to the estimation of the expected future error over the unlabelled test set for each new query. To estimate the expected future error, the model needs to be incrementally retrained for each possible query labelling. This makes expected future error unattractive for implementation in practical active learning scenarios.

2.4.5 Probabilistic Sampling

Probabilistic active learning or PAL, introduced by Krempl et al. [46], combines information- and representative-based active learning. It takes inspiration from expected error reduction and aims to address its problem in computational cost. Similar to expected error reduction, PAL computes the expected classification performance change when giving an instance x a label y and selects the x which results in the highest expected improvement of performance. To estimate the performance change of labelling instances, PAL makes use of the smoothness assumption [47] to assume that instances that lie close to each other in the feature space have the same labels.

Using the smoothing assumption, the impact of an additional label depends on the labels of instances in its neighbourhood. Then in order to determine which instance will be queried, for each instance the *label statistics* $ls = (n, \hat{p})$ are calculated where n expresses the absolute quantity of labelled information, obtained through counting the similar labelled instances for pre-clustered or categorical data. $\hat{p}$ denotes the posterior estimate of an instance x's label as the number of positive labels in its neighbourhood over n. The label statistics are used as parameters to model the true posterior distribution and true label as Beta-distributed random variables, as their actual values are unknown. With these two variables, the expected performance gain is calculated through weighing the label statistics and predicted label realisation by the density of the instance's neighbourhood. Through this process, PAL becomes a combination of an information- and representative-based query strategy, as the neighbourhood of instance's is taken into account.

PAL is more efficient than expected error reduction, as the machine learning model does not need to be retrained every iteration. However, because it makes use of the smoothness assumption, PAL might not work in some cases when instances are not in similar areas of the feature space. Since the introduction of PAL, Krempl et al. have published some variations to probabilistic querying.

Optimised probablistic active learning (OPAL) [48] optimises the label selection for instances based on the minimisation of misclassification loss, making it a cost-sensitive version of probablistic active learning. Section 3.2 provides a more in depth overview of cost-sensitivity in active learning approaches. It also optimises the calculation of the probablistic gain of an instance, making it a more efficient than PAL but is myopic. To remove myopicity from OPAL, Krempl et al. propose a non-myopic extension of OPAL where for each estimated label the possible remaining number of similar label acquisitions under a given labelling budget is considered.

Recently, Kottke et al. proposed another modification to probabilistic active learning. Their proposed method, called Expected Probalistic Gain for Active Learning or xPAL [49], is a Bayesian approach to PAL, generalized to handle multi-class situations. It introduces a conjugate prior distribution (instead of PAL's prior of 1) to determine the class posterior.

2.4.6 Expected Model Change

Similar to probabilistic active learning and expected error reduction, expected model change is another type of decision-theoretic approach to active learning. The difference between expected model change and other decision-theoretic approaches lies in its calculation of the impact of labelling on the machine learning model, instead of on the classification performance. An example of a query strategy based on expected model change is the Expected Gradient Length (EGL) model [50]. This algorithm queries the instance which would result in the highest change in the currently used machine learning model if its label were known. EGL requires a gradient-based machine learning model to be utilized properly, as the change imparted by labelling an instance can be measured by the length of the training gradient. The largest change to the model corresponds to the training gradient with the largest magnitude.

Similar to expected error reduction, EGL is computationally expensive if the feature space and set of labels is large. EGL can also be misguided by the size of an instance's feature and requires feature scaling. Large features result in a training gradient with a high training gradient which can lead the algorithm to draw incorrect conclusions on the most informative instance.

2.5 Active Learning Debiasing Methods

This section will elaborate on different active learning algorithms used for reducing sampling bias in the training data. All the active learning query strategies discussed below, like density-weighted sampling in section 2.4.3, are focused on selecting representative instances. These techniques are all representative of different perspectives on instance selection, which makes it interesting to compare their influence on operational classification performance. Most of the discussed techniques are not purely representative-based, as they aim to balance querying based on representativeness and information value. Therefore, these debiasing techniques predominantly belong to the group of informative- and representative-based methods 2.4. This group of methods is ideal for researching the influence of debiasing sampling bias on performance, as they still aim to query informative instances while reducing sampling bias.

2.5.1 Hierarchical sampling

Hierarchical sampling [24] exploits clustered unlabelled data as it makes use of unsupervised hierarchical clustering techniques. The idea of hierarchical sampling is that representative instances are selected through these clusters. The clustering technique is applied to the unlabelled data before being fed to the active learning method, to provide the method with a binary tree of hierarchical clusters. In the binary tree, each node represents a subcluster of the hierarchically clustered data and every leaf represents an instance. An example of a binary tree representation of the clusters is shown in figure 5, taken from the original paper [24] by Dasgupta and Hsu.

Initially, just a single cluster of the entire unlabelled data is taken by pruning this tree, from which a predefined number of random instances are sent to the oracle to be labelled. After labelling, each node of the tree is checked to see how many sampled instances are present in it and the relative number of positive and negative class instances per node is updated. Afterwards, the partition of the binary tree is replaced by its two child nodes, in figure 5 $\{1\}$ will be replaced by $\{2, 3\}$ as they are more pure (containing more instances of a single class). Then random samples from these two clusters will be queried, with more random samples drawn from the more impure cluster. This process continues until a stopping criterion is reached, usually a labelling budget. An interesting feature of hierarchical sampling is that once the stopping criterion is reached, every in the last-viewed partition gets assigned the majority label of the points queried from it. This means that the entire data is labelled by the end of the hierarchical active sampling process. The downside of this is that stopping early can result in a large amount of incorrectly labelled instances.

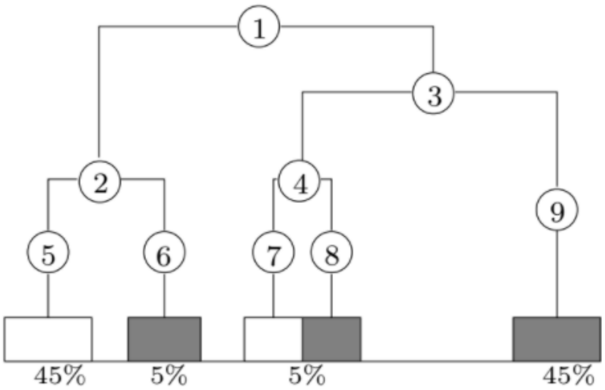


Figure 5: Binary tree representation of hierarchical clusters.

A recent variation on hierarchical sampling is the active learning through density peak clustering (ALEC) algorithm [44]. There are multiple differences between ALEC and hierarchical sampling. Firstly, ALEC uses a tree structure which represents the relationships between nodes better than the binary tree used in hierarchical sampling. The tree grows by adding new nodes instead of pruning a finished tree. The algorithm keeps track of the density of instances through the following formula:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{6}$$

Where d_{ij} is the Manhattan distance between two instances, d_c is the cutoff distance and χ, and

$$\chi(x) = \begin{cases} 1 & x < 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Instances with the highest ρ_i and lowest distance to other instances with high density are selected as the cluster centers. The cluster-centers becomes the root nodes and parent-child nodes among instances are also defined by these density and distance to other high density instances. This process iteratively creates new nodes for the master tree. A deterministic query selection process is applied to sample the $\sqrt{N}$ instances with the highest density and the lowest distance to other high density instances. After the query step, new clusters are found and new nodes are added to the master tree.

2.5.2 Exploration/Exploitation and Thompson Sampling in Active Learning

Taking inspiration from reinforcement learning techniques [51], it is possible to adapt exploration and exploitation to querying instances. In active learning, exploration during querying is seen as searching for large regions in the input feature space the machine learning classifier would incorrectly classify [52]. On the other hand, exploitation in querying is similar to uncertainty sampling, in that exploiting the feature space is querying the instances close to the decision boundary. Consequently, exploitation is a form of information-based querying while exploration pertains to representative-based querying. The balancing of exploration and exploitation in a query strategy then categorizes the strategy as an information- and representative-based query strategy. Reinforcement learning research, in which this balancing act is also a prevailing problem, has proposed variety of solutions of which some can be applied to active learning querying.

A way of balancing exploration and exploitation is to view the sequential decision problem of deciding which instance to query as a *multi-armed bandit problem.* In the multi-armed bandit problem, a gambler must iteratively choose one out of k non-identical slot machines to play. Each of these slot machines has its own distribution of rewards, which is unknown to the gambler. Only through choosing a machine does the gambler receive a reward, which he keeps track of. The goal of the gambler is to maximize his total reward over the entire sequence. Adapting the multi-armed bandit problem to active learning, the choice of slot machine has been interpreted as the choice of different hypotheses through different query strategies [53], whereas the reward is made analogous to the machine learning algorithms test accuracy or other performance metric (discussed in section 2.6). Active Learning By Learning (ALBL) [54] is such a query strategy, using different query strategies as arms and a chosen performance metric as reward.

Bouneffouf et al. [55] address balancing exploration and exploitation through a contextual multi-armed bandit problem. Viewing active learning as a contextual multi-armed bandit problem, each arm is a cluster of instances in the space and the different features of the cluster are the context of the arms. At each query step, the active learning algorithm is presented the different context vectors of each arm. The algorithm keeps track of the history of chosen arms and their corresponding rewards. Using Thompson sampling for contextual bandit problems [56], the query strategy models its past observation as triplet of $(c_t, r_t, b_c(t))$. c_t represents the previously chosen cluster, r_t the chosen cluster's reward and $b_c(t)$ the current context vector. The Bayesian update rule is used to calculate the posterior probability of rewards given this triplet. The cluster (or arm) that maximizes this expected reward is then chosen, from which a random instance is given to the oracle to query.

2.5.3 Informative and Representative Querying in Active Learning

Huang et al. propose a query strategy that measures and combines informativeness and representativeness through prediction uncertainty. They name their proposed framework active learning by QUerying Informative and Representative Examples or QUIRE [57]. In the QUIRE algorithm, how informative an instance is depends on the classifier's prediction uncertainty on labelled data, favouring instances closest to the decision boundary. This makes selecting informative instances equivalent to uncertainty sampling. How representative an instance is depends on the classifier's prediction uncertainty on the unlabelled data. The idea behind this is when an instance is considered to be representative, it is expected to share a large similarity with many of the other unlabelled instances in the pool. To calculate the classifier's prediction on the unlabelled data, the class labels of the remaining unlabelled instances needs to be estimated using a kernel, making this a semi-supervised approach. To approximate these labels, the label assignments of the already labelled instances and the cluster structure of the unlabelled data are used. QUIRE uses a min-max view of active learning, which searches for instances which result in the smallest value for the chosen loss function of the machine learning classifier, regardless of that instance's class label. The drawback of using QUIRE is that it is susceptible to certain assumptions of the input data, as for the calculation of representativeness it requires use of the manifold assumption [58].

In order to combine information- and representative-based active learning without violating preexisting assumptions and constraints on the input data, Du et al. propose their own querying framework [59] and show that it can sometimes achieve better performance than QUIRE. Similar to QUIRE, their query strategies' aim is to query instances that contain the most information for the classifier and are representative of the unlabelled dataset. However, an extra goal of the framework is to query instances which offer as little redundancy as possible in the labelled dataset. The framework uses a tradeoff parameter that balances informative- and representative-based querying. To query informative instances, the framework computes an uncertainty vector for all instances. Representative samples are selected using two-sample discrepancy, where similarity between samples is measured using a posterior probability with the Radial Basis Function. The posterior probability represents the importance of instances to the classifier. Two instances are then seen as similar if they have the same impact on the classifier.

2.5.4 Weighted Active Learning and Plain and Levelled Unbiased Risk Estimation

Applying corrective weights to the queried and labelled instances in the labelled training set helps correct sampling bias. The loss function of the machine learning classifier then makes use of these individual weights, to optimize and direct the focus of the classifier on representative instances. Previously, importance weighing techniques have been used in online active learning scenario's, like in the Importance

Weighted Active Learning (IWAL) algorithm by Beygelzimer et al. [60]

Apart from formalizing sampling bias in their paper, Farquhar et al. use corrective weighing in a pool-based active learning scenario to address sampling bias. They propose two different unbiased population risk estimators to implement on top of existing query strategies [21]. These risk estimators are Plain Unbiased Risk Estimation ($\bar{R}_{PURE}$) and Levelled Unbiased Risk Estimation ($\bar{R}_{LURE}$). Both risk estimators use corrective weighing techniques to weigh each instance according to how probable they are to be sampled next by the query strategy. In order to do this, the estimators require a proposal distribution, akin to a query selection strategy which adds probabilities to each instance in the data pool. Recall that most original active learning techniques estimate the population risk according to equation 3.

The first variation on risk estimation Farquhar et al. introduce is $\bar{R}_{PURE}$. It takes the total amount of labelled datapoints M, the total amount of datapoints N and the loss from an instance $\mathcal{L}_{i_m}$. To weigh instances in risk estimation, $\bar{R}_{PURE}$ uses the probability mass for each instance m as being the next to be sampled, once the labelled training dataset $\mathcal{D}_{train}$ contains $m-1$ instances. The acquisition proposal distribution over instances depends on the instances sampled so far because active learning is applied to the unlabelled data pool. The weight of an instance is then calculated using the following formula:

$$w_m \equiv \frac{1}{Nq(i_m; i_{1:m-1}, \mathcal{D}_{pool})} \tag{8}$$

The complete formula for $\bar{R}_{PURE}$ is as follows:

$$\bar{R}_{PURE} = \frac{1}{M}\sum_{m=1}^{M} a_m \tag{9}$$

Where:

$$a_m = w_m \mathcal{L}_{i_m} + \frac{1}{N}\sum_{t=1}^{m-1} \mathcal{L}_{i_t} \tag{10}$$

So $\bar{R}_{PURE}$ estimates the population risk as the sum of the weighted loss of each sampled instance and the sum of the loss of each previously sampled instance weighed by $\frac{1}{N}$.

$\bar{R}_{PURE}$ has some disadvantages. Say we use a uniform proposal distribution for our selection strategy, equivalent to randomly selecting instances or passive learning. Consequently, as the size of the training set approaches the size of the full data pool, the weights of each instance fail to become uniform. The second and better proven risk estimator is $\bar{R}_{LURE}$, fixes this problem. $\bar{R}_{LURE}$ is calculated using different weights v_m according to the following equation:

$$\bar{R}_{LURE} = \frac{1}{M}\sum_{m=1}^{M} v_m \mathcal{L}_{i_m} \tag{11}$$

Where:

$$v_m \equiv 1 + \frac{N-M}{N-m}\left(\frac{1}{(N-m+1)q(i_m; i_{1:m-1}, \mathcal{D}_{pool})} - 1\right) \tag{12}$$

Instead of w_m, using v_m the weight of instance m does not depend on the position it was sampled in, but only on the probability which which it was sampled [21]. This lowers the variance of the machine learning classifier optimized using $\bar{R}_{LURE}$.

Table 1 shows a summary of the active learning algorithms used in this thesis and whether they are categorized as informative-based, representative-based, hybrid or other query strategies.

Query strategy	Type of Query Strategy
Uncertainty sampling	Informative-based
Density-weighted sampling	Hybrid
QBC sampling	Informative-based
Hierarchical sampling	Hybrid
QUIRE	Hybrid
ALBL	Hybrid

Table 1: Different active learning query strategies used for experimentation and their corresponding category.

2.6 Evaluation Metrics

There are various stages in the active learning querying cycle in which the performance of sampling bias and active learning debiasing methods can be evaluated. At the forefront of active learning performance evaluation lies the performance of the machine learning classifier, trained on the queried and labelled dataset. Therefore, section 2.6.1 will discuss evaluation metrics for supervised machine learning classifiers. Section 2.6.2 will discuss how machine learning classifier performance is measured over query iterations in order to measure the influence of active learning query strategies on classification performance. Section 2.6.3 shows various evaluation metrics for the measurement of sampling bias and sampling bias reduction. These evaluation metrics shall be applied in the following sections detailing the experiments on active learning sampling bias.

2.6.1 Evaluation Metrics for Supervised Machine Learning

		Actual	
		1	0
Predicted	1	TP	FP
	0	FN	TN

Table 2: Confusion matrix depicting differences between predicted and actual class.

To understand how to evaluate the performance of machine learning classifiers, knowledge of the confusion (or error) matrix is necessary. Table 2 depicts such a matrix for a binary classification problem with two classes 0 and 1. The confusion matrix summarizes a machine learning classifiers prediction results on the test set. In the matrix, TP and TN represent the true positive and true negative predictions respectively. These can be seen as the cases where the classifier correctly predicted the positive and negative class. FP predictions are then the instances the classifier predicted as positive, when the ground truth label for the instances were negative. Finally, FN predictions are the instances the classifier predicted as negative when they in reality were positive.

The values from the confusion matrix are used to calculate a myriad of possible of evaluation metrics for quantifying classifier performance. A common performance measure is the *accuracy* of the classifier, which calculates the proportion of correctly predicted instances by dividing it by the total number of predicted instances:

$$Accuracy = \frac{TP + TN}{Total} \tag{13}$$

Antithetical to the accuracy, the *misclassification error* represents the proportion of incorrectly classified instances of all classified instances:

$$Misclassificationerror = \frac{FP + FN}{Total} \tag{14}$$

Therefore to improve classification performance, the goal is to have as high as possible classification accuracy or similarly having an as low as possible misclassification error. For general classification problems, improving the accuracy of the classifier or reducing the misclassification error are valid goals. However, in cases with extreme class imbalance, it is easier to achieve a high accuracy while not adequately predicting minority class instances. If the ratio of majority to minority class is 99 : 1, a classifier which classifies every instance as the majority class is 99% accurate [61]. Another more practical example where accuracy is a poor metric is cancer detection, in which it is more important for the classifier to correctly classify the minority class, instances where an individual has cancer, instead of being generally accurate on detecting mostly non-cancer cases. In these cases, *recall* is a more meaningful performance measure. The recall is calculated by looking at the proportion of correctly positively classified instances when compared to all actually positive instances. Hence, the recall is calculated using the following formula:

$$Recall = \frac{TP}{TotalPositive} \tag{15}$$

Precision is a performance measure which is similar to recall. The precision depicts the amount of instances correctly classified as the positive class divided by the total number of instances the classifier predicted to be positive, so the number of TP and FP predictions.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

The *F1 score* combines and balances the precision and recall measures into a single performance metric. The formula for the F1 score is as follows:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{17}$$

The F1 score is often used to compare the performance of two classifiers. For instance when one has higher recall and the other has higher precision, comparing the F1 score of the two classifiers predictions will show which classifier has better performance.

The final performance metric this thesis will use in experimentation is g-means [62], as it accurately represents performance in high class imbalance situations. G-means is calculated using the following formula:

$$g = \sqrt{Sensitivity * Specificity} \tag{18}$$

Where $Sensitivity = \frac{TP}{TP+FN}$, the accuracy on the positive instances. Similarly, $Specificity = \frac{TN}{TN+FP}$ is the accuracy on the negative instances. So g-means is an aggregated performance metric of the accuracy on both classes of instances.

2.6.2 Evaluation of Active Learning Methods

When applying machine learning classifiers without active learning selective acquisition, the performance metrics discussed in the previous section 2.6.1 are calculated after training the classifier on the complete training set. When applying active learning to a learning problem, the evolution of these performance metrics needs to be put in a graph in order to understand the improvement in performance when adding more labelled instances to the training set. This evolution of performance is visualised through showing a *learning curve.* In the curve, the x-axis depicts the amount of queries performed by the active learning algorithm and the y-axis depicts the chosen machine learning performance metric. The learning curve updates as the machine learning classifier is retrained and tested. As the main goal of active learning is to have a high performing machine learning classifier in as few labelled instances of possible, the learning curve provides insight into which active learning algorithms reach high performance in the fewest amount of queries.

Often it is useful to summarize the performance on the learning curve into a single value to easily compare active learning methods. The *area under the learning curve* or ALC has often been used as a summarised metric [63]. The ALC is a normalized score which represents how much total performance a classifier has accumulated as a result of adding queries to the labelled training set [64]. This value is calculated through taking the area under the receiver operating characteristic (ROC) [65] curve, commonly known as the AUC, at each point during training and plotting it as a function of the size of the labelled set. ALC is calculated using the following formula [66]:

$$ALC\ score = \frac{ALC - Arand}{Amax - Arand} \tag{19}$$

Where ALC is the area under the AUC learning curve, Arand is the area under the learning curve obtained by random prediction (0.5 AUC throughout learning) and Amax is the area under the best achievable learning curve (1 AUC throughout learning). The ROC depicts how the ratio between true and false positive prediction rate of a machine learning classifier evolves. The ALC is then the calculated area under this curve. A somewhat similar performance measure is the deficiency score [67]. The downside of using this metric is that it is dependent on the total number of instances in the labelled training dataset, so to compare two methods both need to be trained with the same amount of instances.

The ALC score and learning curve does not explain if the machine learning model generalizes well through training on the selected instances by an active learning method. Therefore, active learning research often makes use of k-fold cross-validation [68] to validate the machine learning model's performance on multiple validation sets. K-fold cross-validation separates the training sets into k proportional folds. At each of k iterations, $k - 1$ of these folds are grouped together to form the training set and the model's performance is validated on the left out fold. Each iteration, a different validation fold will be chosen.

To evaluate the difference in performance of two different active learning algorithms, it is important to look at the entire querying and training process instead of the performance of only one set of predictions. To test whether the performance of one active learning method over the entire learning process is significantly better than another, a statistical test like the Wilcoxon signed-rank test [69] can be applied to the summarizing ALC values of the two differently trained classifiers. The Wilcoxon signed-rank test evaluates whether there is a significant difference between two performance sets. The benefit of the Wilcoxon signed-rank test is that, unlike a standard t-Test, it does not require any parameters. However, the test does assume that the data follows a normal distribution.

2.6.3 Visualising Bias and Bias Reduction

Separate evaluation metrics and performance visualisations exist which depict the sampling bias introduced by active learning algorithms. These visualisations range from depicting an active learning query strategies behaviour, visualizing the difference in class distribution between the labelled dataset and the unlabelled population dataset to highlighting the labelled data distribution in the original distribution.

To understand how sampling bias occurs, it is important to study the behaviour of the active learning query strategy. In order to view this behaviour, ex ante analysis of what the query strategy deems the most informative instances is necessary. Here, the predetermined information heuristic is chosen as an evaluation metric. The information values of features can be visualised through a selection heatmap, depicting the input feature space with more informative features having a darker colour. This heatmap can be generated after a certain number of queries are conducted by the query strategy, to show how the choices of the query strategy evolves as more instances are added to the labelled dataset. Another option is to visualise a class selection frequency graph for label selections ex post, showing what labels a query strategy prefers at certain query iterations. These visualisations are visualised over multiple runs of a query strategy on the same data.

As studying sampling bias is at the forefront of this thesis, visualising how the bias evolves over queries will give clarification of the sampling bias in the labelled dataset. This thesis will focus on visualizing bias through class distribution disparity between the labelled dataset and the unlabelled population dataset. Therefore, the visualisations detailed in section 5.4 are created to provide insight into this form of sampling bias. These visualisations, alongside the learning curve and ALC scores discussed in the previous section, will provide insight into sampling bias and the performance of active learning debias techniques.

3 Related Work

While this thesis covers only one phenomenon associated with active learning, namely active learning sampling bias, there are other fields of research on the subject of active learning. The first section will be a reflection on and critique of various active learning and sample selection bias publications, and will provide more detail on the place of this research in active learning and sampling bias literature. The following two sections will discuss other fields of active learning research which are relevant to both this research and research done by the IDlab. One of these fields of research is the application of active learning in situations where labelling or misclassifying instances have different costs. The other field in which active learning is applied is the mitigation of extreme class imbalance in the underlying unlabelled distribution. Knowledge of both these fields of study can aid the implementation of active learning in deployed machine learning models. The final section discusses various issues one should be aware of when deploying active learning in a production setting.

One of the main topics of this research is active learning sampling bias and its effect on machine learning classifiers. There exists a myriad of research on bias' effects on machine learning, including human sampling bias. The survey conducted by Mehrabi et al. [25] provides an excellent overview of the kinds of bias and their effect on machine learning classifiers. The survey also summarises various existing techniques for mitigating bias and improving fairness in machine learning predictions. In contrast to bias reduction techniques in supervised model training and predictions, part of this thesis is concerned with the effects of addressing sampling bias through using specific bias mitigating active learning query strategies.

The experiments conducted in this thesis examine active learning query strategies in binary classification cases on multivariate data. Active learning query strategies and debiasing techniques have also been applied to image classification. For instance, Prabhu et al. conducted an empirical study on the effects of sampling bias on image data [22]. Another example is the supervised contrastive active learning (SCAL) algorithm introduced by Krishnan et al. [70]. Through combining active learning and contrastive learning in the loss function, SCAL selects diverse and informative instances in image classification.

3.1 Reflection on Sampling Bias and Active Learning Research

Most work on sampling bias through active learning querying is concerned with debiasing algorithms and modifications to existing query strategies. Many perspectives on debiasing sampling bias have already been discussed in section 2.5, along with some well-performing published examples per perspective. Publications on the reduction of active learning sampling bias state that the introduction of sampling bias might be introduce problems for the machine learning classifier [57][71]. However, these publications do not state what factors influence sampling bias and what effect sampling bias has on the performance of the supervised model trained on the newly labelled dataset. Therefore, the first half of this thesis will empirically study the effects of four facets in the pool-based active learning pipeline on sample selection bias. These facets are: the choice of active learning algorithm, machine learning classifier, the level of imbalance in the data pool and the class ratio of the initial training set.

In research on active learning query strategies, there is a general knowledge in active learning research that informative-based query strategies can introduce sampling bias. Sampling bias is often visualised through feature bias, by visualising the feature space and showing the selection of instances through querying. While this is one of two types of sampling bias [22], publications on active learning sampling bias often neglect to other form of sampling bias: class bias. Therefore, this thesis will study sampling bias through visualising class bias in various ways. Studying the effect of the four aforementioned facets in the active learning pipeline on sampling bias through looking at class bias will offer new insights in active learning sampling bias research.

The second half of this thesis is concerned with the reduction of sampling bias through the implementation of multiple active learning query strategies which are more representative-based. Multiple publications on the reduction of sampling bias through new active learning algorithms have been published. However, there has never been a large-scale performance comparison of these algorithms and a clear depiction of their querying behaviour through class selection and class bias visualisations. Usually, publications of a new bias mitigation algorithm contain a brief performance comparison on around 3 datasets to a couple of well-known query strategies. By comparing 3 active learning debiasing methods with random (passive) sampling, density-weighted sampling and uncertainty sampling on 15 datasets, this thesis will provide a general overview of performance differences and the relation between bias mitigation and the learning process.

3.2 Cost-Sensitive Active Learning

Active learning's main focus is on minimizing the cost of labelling while producing a high quality labelled dataset suited for training supervised classifiers. While there are costs associated with labelling, sometimes the costs of labelling instances may vary depending on the class of instance. Furthermore, there might be a varying misclassification cost of the supervised classifier where the entries of the confusion matrix in table 2 are all associated with different penalties. Researchers have modified the standard active learning schema to incorporate cost-sensitivity in instance querying.

Annotation cost-sensitive active learning algorithms have been proposed as a remedy to cases where labelling instances have differing costs. Tsou and Lin proposed a cost-sensitive tree algorithm [72], which simultaneously estimates the unknown labelling costs and the utility of querying instances. Inspired by hierarchical sampling [24], the algorithm iteratively constructs a tree where nodes represent queried instances.

Charles Elkan [73] proposed a foundational framework for cost-sensitive classification, making use of a modified version of the confusion matrix, referred to as the cost matrix. Given a cost value associated to each field of the cost matrix, new instances should be predicted by the classifier to have the class that leads to the lowest expected cost. This expectation is computed by using the conditional probability of each class given the example. For binary classification, the cost matrix is used to rewrite the optimal decision threshold. To make this optimal decision threshold correspond to the data of the training set, the number of negative samples in the training set should be reweighed or the dataset should be resampled.

Since Elkan's Foundations of Cost-Sensitive Learning paper, the idea of cost-sensitive classification has been adapted into active learning selective acquisition. Krishnamurthy et al. create an active learning algorithm for cost-sensitive multiclass classification, which they name Cost Overlapped Active Learning (COAL) [74]. COAL regresses each label's cost and queries instances which both have the lowest predicted cost and the most ambiguity about their label, the latter rule being akin to uncertainty sampling.

3.3 Class Imbalance and Active Learning

Throughout the years since active learning's inception, active learning algorithms have been applied to class imbalance cases to create more balanced labelled datasets. It is important to note that creating a balanced distribution in the labelled dataset while the original distribution is severely skewed, means that the labelled dataset exhibits sampling bias. However, in some cases this sampling bias may be useful, for instance when the minority class needs to be represented more. In these cases, active learning algorithms can be considered alongside resampling techniques to create a new, more balanced data distribution. This section highlights some interesting publications on applying active learning for rebalancing data distributions.

Ertekin et al. [75] demonstrate that active learning can be utilized in order to rebalance class imbalanced data. Their proposed solution is a support vector machine (SVM) based selection strategy which queries only a small pool of data at each query iteration instead of searching through the entire input feature space. Through comparison with other resampling techniques, the SVM based active learning algorithm is shown to produce well-balanced datasets and leads to classifiers with high performance.

Attenberg and Ertekin [61] provide further clarification on the class imbalance problem in active learning. They do so by providing an overview of class imbalance reducing active learning algorithms. The different algorithms discussed in their overview range from density-sensitive active learning, a modification to the query-by-committee algorithm with class-specific costs by Tomanek et al. [23] and Ertekin's VIRTUAL resampling algorithm [18]. The final sections address some of the difficulties encountered when applying active learning. Applying active learning to extreme class imbalance cases can lead to informative-based and even some more representative-based query strategies to severely underperform. Another important possible problem is the *cold start problem.* This term is used to describe the effects of poorly selected starting data for the supervised classifier, which leads to poor future instance selection.

From the descriptions of previous research on class imbalance and active learning, it is noticeable that most literature on these subjects uses some form of active learning to address class imbalance instead of researching the effect of imbalance on active learning methods. While some publications on the subject imply some influence of class imbalance on querying behaviour, there is hardly any research on the exact effect of imbalanced data on instance acquisition through active learning. Therefore, this thesis will provide new insights into the connection between imbalanced data and active learning, by experimenting and studying the effect of class imbalance on querying behaviour and on active learning sampling bias.

3.4 Active Learning in Deployed Models

Uncertainty in informative-based query strategies is seen as analogous to the distance to the decision boundary of the classifier, seen as the predictive uncertainty of the classifier. This is only one kind of uncertainty however, as is made clear when deploying active learning and machine learning in real-world situations. Deployed machine learning models can encounter two differing forms of uncertainty: *aleatoric and epistemic uncertainty.* Hüllermeier and Waegemen describe these two types of uncertainty and how they affect machine learning predictions [76]. Aleatoric or statistical uncertainty is the type of uncertainty resulting from randomness or noise in the training data, as in experiments the variability of an outcome is due to some degree of randomness. A clear example of aleatoric uncertainty is the flip of a coin, where the results of the coin flip are affected by some stochasticity which can never be reduced. Epistemic uncertainty is the "uncertainty caused by ignorance" [76]. In contrast to aleatoric uncertainty, epistemic uncertainty can be reduced by gaining knowledge of the problem. In machine learning, epistemic uncertainty is reduced by training the machine learning model with more instances.

In uncertainty sampling, predictive uncertainty has been exchanged for aleatoric and epistemic uncertainty by Nguyen et al. [77] to see whether these types of uncertainty are more useful. In order to query instances based on aleatoric and epistemic uncertainty, these types of uncertainty are modelled through the method proposed in [78]. Due to the random nature of aleatoric uncertainty, Nguyen et al. conclude that this type of uncertainty is unfit for determining the informativeness of instances. However, separating aleatoric and epistemic uncertainty and querying instances purely based on their epistemic information value is a better criterion, yielding similar and sometimes better performance results than traditional uncertainty sampling.

Quantifying the amount of aleatoric uncertainty has its implications and uses in supervised learning problems. For instance, the estimation of aleatoric uncertainty can aid in estimating and modelling epistemic uncertainty, as done in [79]. Another use of estimating aleatoric uncertainty is in querying instances. If instances in iterative querying have high aleatoric uncertainty, an oracle can label these instance and provide more in depth expertise on the nature of these instances. This can improve the expertise of the selected training data and improve the performance of classifiers. While this thesis' focus is not on aleatoric and epistemic uncertainty in active learning, it does visualise the learning process more clearly through graphs which show sampling bias and querying behaviour. Other research on active learning and sampling bias often only shows the performance of classifiers trained with a new bias reducing active learning algorithm. Alongside performance graphs, the bias graphs shown in this thesis will show more of the learning process and effectively the reduction of epistemic uncertainty.

4 Dataset description

In order to both study the effects of various factors on active learning sampling bias and to compare the performance of various active learning bias mitigation algorithms, running experiments on a single dataset would result in inconclusive evidence. To provide general and complete conclusions on active learning algorithm behaviour, experiments on a larger variety of datasets are necessary as it shows that the results shown are not purely data-dependent. Therefore, after discussion with the IDlab, the decision has been made to run experiments on 15 datasets. This section provides an analysis of the used data and the preprocessing steps taken in order to feed the data through the experiment pipeline.

4.1 Data Sources and Analysis

Fifteen binary classification datasets, with varying amounts of instances and features, have been obtained by using the OpenML platform API [80]. All instances in these tabular datasets have been annotated with their corresponding labels. The choice was made to acquire the 15 datasets using OpenML, as it gathers datasets from various acclaimed data repositories while making them easily accessible. One of the criteria when selecting the datasets was that each dataset had < 15000 instances. Another criteria was that a classifier fully trained on a created training set would achieve a high performance. Specifically, when training the classifiers (described in section 5.1) on a complete training dataset, the classifier's AUC on a test set would have to be a minimum of 0.8. This characteristic was kept in mind when deciding on which datasets to use, as a high fully trained classifier performance would entail a noticeable and gradually improving learning curve when applying active learning to the dataset. The last chosen dataset characteristic was that either around 50% or less of the instances belong to the positive class and the rest to the negative class. Using this and to avoid confusion, the next sections will refer to the positive class as the minority class and the negative class as the majority class.

The 15 datasets have all been created in order to solve various distinct problems. Examples of classification problems covered by these datasets are distinguishing between genuine and forged banknotes, common or faulty steel plates, edible or poisonous mushrooms, if the King+Rook versus King+Pawn endgame in chess wins or doesn't and whether a tree are diseased or not. For the sake of brevity, table 3 details some of the most important characteristics of the used datasets. For more specific information about a dataset such as how the data is collected, please have a look at the original papers in which the datasets are introduced. All papers are cited in the table. A dataset's class ratio was defined in equation 1. In further descriptions of experiments, class ratio will be used to refer to this proportion in the original data distribution.

Dataset Name	# Instances	# Features	Class Ratio	Source
monks-problems-3	554	7	0.52	UCI [81]
qsar-biodeg	1055	42	0.337	Mansouri et al. [82]
hill-valley	1212	101	0.5	Graham and Oppacher on UCI [81]
banknote-authentication	1372	5	0.445	Lohweg and Doerksen on UCI [81]
steel-plates-fault	1941	34	0.347	Semeion on UCI [83]
scene	2407	300	0.179	Mulan [84]
ozone-level-8hr	2534	73	0.063	Zhang et al. [85]
jasmine	2984	145	0.5	AutoML [86]
kr-vs-kp	3196	37	0.522	UCI [81]
Bioresponse	3751	1000	0.542	Kaggle [87]
spambase	4601	58	0.394	UCI [81]
wilt	4839	6	0.054	Johnson, B., Tateishi, R., Hoan [88]
churn	5000	21	0.141	Used in [89]
mushroom	8124	23	0.482	Lincoff et al. [90]
PhishingWebsites	11055	31	0.557	UCI [81]

Table 3: All datasets used in the active learning sampling bias experiments.

4.2 Data Preprocessing

Few preprocessing steps were taken in order to provide the active learning query pipeline with clean data. This is due to the fact that all the used datasets taken from the OpenML platform have been integrated into the platform to be easily reusable. All chosen datasets contain neither missing nor sparse data, so no dataset specific preprocessing steps were necessary before running the active learning experiments. The labels were encoded to 0 being the majority (negative) class and 1 being the minority (positive) class respectively.

For answering the first research question, experiments were conducted visualising the effect of class imbalance on active learning sampling bias and on performance. Therefore, the final preprocessing step was to take three subsets of the original dataset in order to manually introduce class imbalance in the dataset. These subsets are only used for the class imbalance experiment discussed in 5.3. The three subsets of balanced data $class\ ratio = 0.5$, minor class imbalance $class\ ratio = 0.25$, and high class imbalance $class\ ratio = 0.05$ were created using random undersampling of both the majority class and minority class. The choice of class to randomly undersample the minority or majority class depends on the choice of subset. For $class\ ratio = 0.5$, the majority class is undersampled until both classes are equally represented. For $class\ ratio = 0.25$ and $class\ ratio = 0.5$, the majority class is undersampled to reach the ratio of 0.25 or 0.05. If this is not possible, the minority class is undersampled to reach the class ratio required in the subsets. The three subsets were chosen because they differ greatly in class ratio, thus making it interesting to see how active learning algorithms are possibly affected by different levels of class imbalance in the data pool they are fed.

5 Methodology

This section elaborates on the design choices made for answering the three main research questions. These design choices were made in accordance with both prior academic research on sampling bias in active learning, and with prior research on active learning conducted by the IDlab. When looking at what factors influence active learning sampling bias, experiments were conducted through looking at different settings of 4 facets of the active learning pipeline. The chosen facets were: the choice of machine learning classifier, class ratio of the initial training set, active learning query strategy and the level of class imbalance in the data pool. Section 4.2 has provided detail on the chosen levels of class imbalance for each dataset. The following subsections provide more detail on the other three facets and the various chosen settings for that facet. Section 5.2 then details the three debiasing methods chosen for answering the second and third research questions.

5.1 The Base Classifier and Class Ratio of Initial Training Set

Three different machine learning classifiers were used and compared in this research: logistic regression, random forest and XGBoost. In order to compare their performance and influence on sampling bias, these three classifiers were compared while all other facets in the pipeline were identical. Logistic regression, being a global classifier (see 2.3), has been proven to be affected by human sampling bias [27]. This makes it interesting to see if the performance of a logistic regression classifier is affected when sampling bias occurs in the labelled dataset due to active learning instance selection. A random forest classifier was added, as including it showed how a commonly used ensemble decision tree approach is affected by sampling bias. The random forest classifier is often used by the IDlab in various machine learning research projects, so it is suitable to compare this classifier to logistic regression and XGBoost. Finally, XGBoost was added as a classifier choice because of its inclusion in the previous IDlab active learning research on classifying ship waste dumping [2].

All classifiers were initialized with a randomly selected labelled training set of 10 instances, where the same initial training set was used for all different settings of a facet in the pipeline. Using identical initial training sets ensured clear comparison between the settings of a certain facet. The initial training set was required by both active learning libraries used in this thesis: ModAL [91] and libact [92]. As in the previous IDlab research [2], a size of 10 instances was chosen to allow differences in the initial query iterations for the individual committee members of the QBC approach. The class ratio of these initially labelled instances, from now on referred to as the *initial set ratio*, is another factor that might influence the querying behaviour of active learning algorithms. Therefore, experiments have been conducted wherein the initial set ratio is changed from balanced 0.5, minor imbalance 0.25 and imbalanced 0.1. For all other experiments, a balanced initial set ratio of 0.5 is used as it resulted in the highest initial performance (see section 18).

5.2 Active Learning Methods

This subsection discusses all the different query strategies experimented with in this research. All query strategies have been described more thoroughly in sections 2.4 and 2.5.

For the first research question (see 1.2) on the influences on sampling bias, the following active learning algorithms were studied: *random sampling (or passive learning), uncertainty sampling, density-weighted sampling and QBC sampling.* See table 4 for a summary of these methods and their parameters. These algorithms were chosen as they are some of the most popular active learning algorithms to use in unknown situations. The algorithms also differ greatly in method, especially the more representative-based density-weighted sampling.

For the second and third research questions, three informative- and representative-based query strategies methods were studied: *hierarchical sampling, QUerying Informative and Representative Examples (QUIRE) and Active Learning By Learning (ALBL).* These algorithms were included as active learning sampling bias mitigation or debiasing methods, as they query more representative instances while still having some informativeness measure to ensure high performance. This makes them attractive query strategies to study for practical applications of IDlab studies, as they are more likely to be implemented than purely representative sampling due to their overall faster learning rates. These debiasing methods all make use of a logistic regression classifier for querying instead of all three classifiers discussed in section 5.1. This design choice was made in order to simplify experimentation with these algorithms, as including other classifiers would complicate interpreting results. Random sampling, uncertainty sampling and density-weighted sampling were also included in experiments as comparison. Random sampling and uncertainty sampling serve as a baseline comparison to the debiasing methods that focus more on querying representative instances. Density-weighted sampling, also being a hybrid query strategy, was included as comparison in terms of sampling bias reduction and performance. Table 5 contains a summary of these query strategies along with their chosen parameters. For all chosen query strategies in tables 4 and 5, the default or most common hyperparameters were chosen as in [2]. The reason for this was to produce results that show the use of active learning in new situations where there is little known about the available data.

5.2.1 Uncertainty Sampling

Being a widely popular and generally well-performing information-based query strategy, uncertainty sampling was included in all experiments. In this thesis, the least confidence strategy or $U(x) = argmax(1 - P(\hat{y}|x))$ was chosen as uncertainty measure for uncertainty sampling. This measure was the default and highest performing measure on all 15 datasets. For studying the factors that influence sampling bias, uncertainty sampling was an apt inclusion due to it's myopic nature. When comparing various active learning debiasing methods, uncertainty sampling is included as a baseline comparison in order to study whether removing sampling bias can yield a higher performance than a query strategy that generally introduces sampling bias.

5.2.2 Density-weighted Sampling

Like in [2], uncertainty sampling using density as weight factor was included in this research. The combination of uncertainty sampling with density weighing ensures that this query strategy, while overall being a representative-based query strategy, still has some informative-based elements as well. This ensures that in higher class imbalance situations, the minority class will still be sampled. Being a generally representative-based method and hopefully less prone to introduce sampling bias in the labelled dataset, this query strategy was included in experiments for answering all research questions.

5.2.3 QBC Sampling

The final query strategy included for looking at the factors that influence sampling bias is query-by-committee or QBC sampling. Like uncertainty sampling, this method belongs to the informative-based query strategies. However, this method uses a completely different informativeness measure from uncertainty sampling, namely a disagreement between multiple machine learning models. Another reason for its inclusion is that QBC can in many cases outperform uncertainty sampling in terms of learning rate. QBC learning has no standard configuration, therefore the choice of configuration is a hyperparameter when implementing this query strategy. This research will use one of the highest performing configurations found in [2]. As finding the ideal configuration for all 15 datasets is not the primary focus of this research, the choice of a proven high performing configuration for QBC sampling was made. This configuration uses both random forest and the XGBoost classifiers as learners, each instantiated with 4 committee members. Therefore, in total 8 committee members are used to decide on the most informative instances using a maximum disagreement sampling strategy.

Query Strategy	Parameters
Random Sampling	No parameters
Uncertainty Sampling	Least confidence strategy used as uncertainty measure: instance with lowest posterior probability chosen.
Density-weighted Sampling	Uncertainty sampling with density as weight to instances. Density defined as summary of cosine similarity between an instance and all other instances.
QBC Sampling	Set of 4 Random Forest and 4 XGBoost classifiers. Training sets for QBC learners randomized with replacement to form different hypotheses. Max disagreement sampling used as informativeness measure for committee members.

Table 4: Q1: Summary of query strategies used for studying the factors that influence sampling bias.

5.2.4 Hierarchical Sampling

Hierarchical sampling uses the standard clustering technique described in section 5. Two choices have been made when implementing hierarchical sampling. The first is the inclusion of least confidence uncertainty sampling as the subsample query strategy. This strategy is used instead of random sampling to select instances in the created pruning, which ensures a more informative approach when combined with the representative clustering done by hierarchical sampling. The other is the addition of so-called active selecting, in which the sample weight of a pruning is defined by its weighted error bound instead of the weight being the number of unseen leaves.

5.2.5 QUIRE

The second debiasing technique studied in this thesis is QUIRE. Due to QUIRE's min-max view of active learning, all instances in the pool must be evaluated at each query iteration. This significantly slows down computational performance when running this query strategy. For this reason, QUIRE was only applied to the 8 datasets with instances ≤ 3000. Specifically, QUIRE was applied to the following datasets from table 3: monks-problems-3, qsar-biodeg, hill-valley, banknote-authentication, steel-plates-fault, scene, ozone-level-8hr and jasmine. Running QUIRE on about half of the datasets will be enough to offer proper insights into its querying behaviour and performance. When implementing QUIRE, the most important hyperparameter is the choice of kernel. This is used for estimating the remaining class labels of the unlabelled instances in the pool. For this research, after experimenting with all kernels on the 8 datasets, a radial basis function kernel was chosen as it resulted in the highest overall average performance on the 8 datasets.

5.2.6 Active Learning By Learning

Active learning by learning or ALBL has been included as the multi-armed bandit approach to querying informative and representative instances. It makes use of the EXP4.P algorithm for selecting the different arms: the 5 different query strategies. The 5 strategies chosen as arms are random sampling, 2 uncertainty sampling with different uncertainty measures and 2 density-weighted sampling methods with varying amounts of clusters. The two different uncertainty sampling algorithms make use of least confidence and entropy as their uncertainty measures. The two density-weighted sampling algorithms use 5 and 10 clusters respectively. The distribution of representative- and informative-based query strategies was chosen to ensure a uniform starting point for the ALBL algorithm before running the EXP4.P algorithm to iteratively choose arms for querying.

Query Strategy	Parameters
Random Sampling	No parameters
Uncertainty Sampling	Least confidence strategy used as uncertainty measure: instance with lowest posterior probability chosen.
Density-weighted Sampling	Uncertainty sampling with density as weight to instances. Density defined as summary of cosine similarity between an instance and all other instances.
Hierarchical Sampling	Using uncertainty sampling with logistic regression classifier as subsample query strategy to sample nodes in selected pruning. Sample weight of a pruning is its weighted error bound.
QUIRE	Using radial basis function (rbf) kernel with kernel coefficient of 1.
ALBL	Using EXP4.P as multi-armed bandit algorithm to choose between random sampling, 2 uncertainty sampling (1 least confident and 1 entropy based) and 2 density-weighted uncertainty sampling (1 with 5 clusters and 1 with 10).

Table 5: Q2&3: Query strategies and sampling bias mitigation strategies.

5.3 Experiment Pipeline

Below is the experimental pipeline used for all experiments.

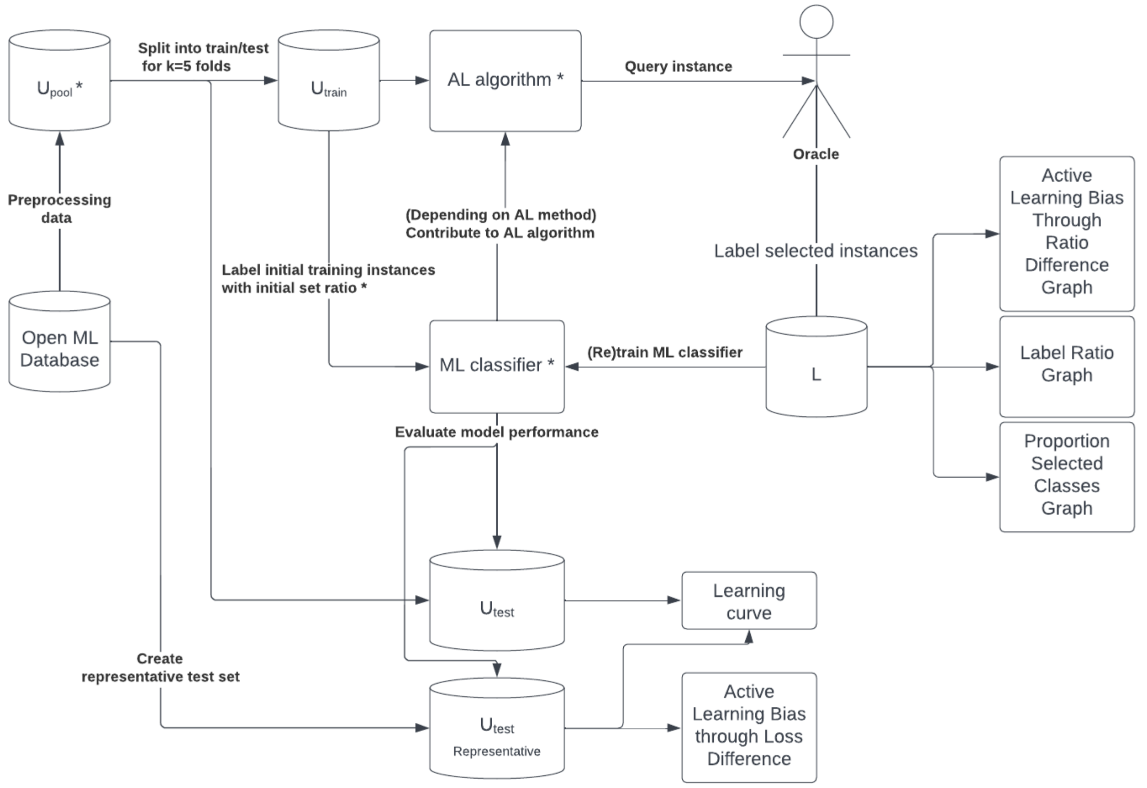

Upool *
Split into train/test for k=5 folds
Utrain
AL algorithm *
Query instance
Oracle
Preprocessing data
(Depending on AL method) Contribute to AL algorithm
Label initial training instances with initial set ratio *
Label selected instances
Active Learning Bias Through Ratio Difference Graph
Open ML Database
ML classifier *
(Re)train ML classifier
L
Label Ratio Graph
Proportion Selected Classes Graph
Evaluate model performance
Utest
Learning curve
Create representative test set
Utest Representative
Active Learning Bias through Loss Difference

Figure 6: Experimental pipeline layout for answering research questions. Sections marked with a $*$ are facets of the active learning pipeline experimented with to see the effect on active learning sampling bias.

Experiments were conducted using $k = 5$ fold cross-validation and ran for a total of 20 executions, with the exception of experiments using QBC and QUIRE sampling, which ran for 5 executions. At each fold, the unlabelled pool U_{pool} is split into $\frac{4}{5}$ training data U_{train} and $\frac{1}{5}$ test data U_{test}. Through active learning querying the labelled training dataset L is created. Using L, three out of four bias visualisations are created. The classifier is trained on L and is both validated on U_{test} and evaluated on a representative test set. U_{pool} is the same as the original database in all experiments with the exception of the class imbalance experiment. Furthermore, the oracle in experiment settings is automated as all labels for the corresponding instances are available. The parts of the pipeline above marked with an $*$ contain multiple settings which are changed in isolation in order to provide answers to the three research questions. This section summarizes the different facets and their settings, categorized by research question.

Research question 1: What factors in the active learning cycle influence active learning sampling bias in the labelled dataset? To answer this question, the settings of the following 4 facets marked with a $*$ in figure 6 are alternated in isolation while all other settings remain the same:

1. Class imbalance: class imbalance in U_{pool} was changed between the original dataset with original class ratio, a balanced 0.5, minor imbalance 0.25 and high imbalance with 0.05 class ratio.
2. AL method: options used were random sampling as baseline comparison, uncertainty sampling, density-weighted sampling and QBC sampling.
3. ML classifier: options were logistic regression, random forest and XGBoost.
4. Initial set ratio: the ratio of the initial 10 labelled instances was changed between balanced = 0.5 initial set ratio, minor imbalance 0.25 and imbalance 0.1.

Research questions 2&3: How do different active learning sampling bias mitigation techniques compare when looking at operational classification performance? Will the influence of active learning sampling bias mitigation techniques yield similar results on the same dataset with varying degrees of class imbalance? These questions are grouped together because they both require the same experiments settings. To answer these questions, the following facets in the experimental pipeline 6 were alternated in isolation while all other settings remained the same:

- AL method or AL debiasing method: random sampling and uncertainty sampling are included as comparison to passive learning and an informative-based query strategy. Density-weighted sampling is included as comparison to an informative- and representative-based strategy. The AL debiasing methods included were hierarchical sampling, QUIRE and ALBL.
- Class imbalance: class imbalance in the dataset fed to the active learning sampling was changed between the original dataset with original class ratio, a balanced 0.5 class ratio, minor imbalance 0.25 and high imbalance with 0.05 class ratio. Changing the class imbalance in the dataset fed to the active learning cycle was necessary for answering research question 3.

5.4 Evaluation and Validation

The experiments described in the previous section aim to measure and visualise the two results of applying active learning to classification problem. Firstly, the sampling bias in the labelled dataset is visualised through class bias. Secondly, the performance of the machine learning classifier is visualised through learning curve of various performance metrics. For the first research question, measuring and visualising sampling bias through various means will show how certain aspects of the active learning cycle might introduce sampling bias. In order to understand whether a reduced sampling bias in the labelled dataset will improve performance, the experiments second and third research questions will show how sampling bias and classifier performance relate . This section details the various performance measures and graphs chosen for quantifying classifier performance and visualizing sampling bias.

The main performance measures chosen in the learning curve were macro-averaged precision, recall, F1 and AUC (see section 2.6). These performance measures were chosen in accordance with prior IDlab experiments, where these metrics are more indicative of high performance than accuracy as performance metric. This is because most IDlab projects contain higher levels of class imbalance. All chosen performance metrics were macro-averaged as almost all datasets, the positive or target class is the minority class. Each macro score represents the average of the scores for each individual class, which ensures that in high class imbalance settings the majority class does not outweigh the minority class. Macro-averaging has been used in prior IDlab research like in [2], which enables clear comparison of results between publications. For all performance graphs, the lower and upper quartile of performance results were only shown after every 5 query iterations instead of at every query to increase readability.

The ALC value (see section 2.6) was chosen to evaluate and compare learning performance between different active learning algorithms in section 6.2. The ALC values of the AUC learning curve ensure that statistical significance tests can be made on performance differences between active learning algorithms. Since part of this thesis concerns itself with research into the general performance differences of active learning debiasing algorithms, the ALC calculation and comparison will be over all 15 datasets. The ALC can then be used in a Wilcoxon signed-rank test to see if the changes in active learning method result in a statistically significant performance difference.

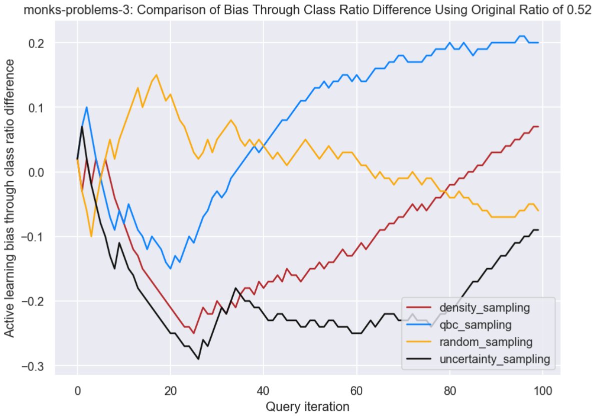


Figure 7: Example of class ratio graph on the monks-problem-3 dataset. Results here and in following example graphs were obtained using a logistic regression classifier and balanced initial set ratio.

Query strategy behaviour and sampling bias was visualised through 4 different graphs. Analyzing a variety of graphs will hopefully provide more insights into active learning sampling bias. While sampling bias can also be seen through sensitivity or feature selection analysis, this research will focus on viewing sampling bias through the lens of class bias and class selection behaviour through querying. The reason for this is the fact that this thesis experiments with a widespread of datasets and tries to offer general conclusions on querying behaviour through class selection and class ratio disparity analysis, whereas sensitivity and feature analysis is more dataset specific. First, a graph detailing the difference between the class ratio of the labelled dataset and the class ratio (see formula 1 of the original dataset was included. An example of this graph is shown in figure 7. Specifically, the difference in class ratio's is defined as:

$$class\ ratio(U_{pool}) - class\ ratio(L) \tag{20}$$

Where $class\ ratio(U_{pool})$ is the class ratio of the preprocessed pool data and $class\ ratio(L)$ is the class ratio of the labelled dataset, which changes as instances are queried and added. Tracking the class ratio difference will show if the label distribution of the labelled dataset differs from the distribution of the dataset the active learning algorithm is fed in the pipeline, one of the key indicators of sampling bias.

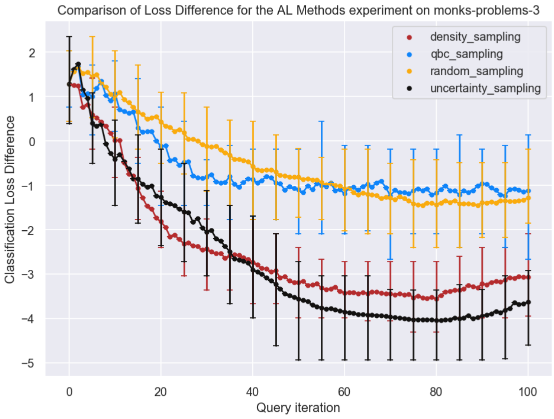


Figure 8: Example of a loss difference graph on the monks-problem-3 dataset using random sampling, uncertainty sampling, density-weighted sampling and QBC sampling.

Figure 8 shows an example graph of the second bias visualisation this research will utilize, through visualizing average loss difference between the fully trained classifier and a classifier trained on data gathered through an active learning query strategy. Inspired by the paper by Farquhar et al. [21], the graph visualises sampling bias by showing the difference between the empirical risk and the sub-sample empirical risk. The empirical risk is defined as the average loss of the classifier on the test set T over all trained instances n in the training dataset N. The sub-sample empirical risk is the average loss of the classifier on the same test set over all actively sampled and labelled points l in the labelled dataset L. For the upcoming experiments, log-loss was chosen as performance metric. To summarize, this graph will visualise bias using the following formula:

$$\begin{aligned} AvgLoss_{Full} - AvgLoss_{Labelled} &= \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\hat{y}^n, y^n) - \frac{1}{L}\sum_{l=1}^{L}\mathcal{L}(\hat{y}^l, y^l) \\ = \frac{1}{N}\sum_{n=1}^{N} -\frac{1}{T}\sum_{i=1}^{T}[y_i ln\ p_i + (1-y_i)ln(1-p_i)] &- \frac{1}{L}\sum_{l=1}^{L} -\frac{1}{T}\sum_{j=1}^{T}[y_j ln\ p_j + (1-y_j)ln(1-p_j)] \end{aligned} \tag{21}$$

Where p is the predicted probability, y is the actual value, N is the size of the full dataset and M is the size of the queried and labelled dataset through an active learning algorithm. $N > L$ for all queried training data and both the empirical risk and sub-sample empirical risk are calculated through loss calculations over y_i from the same representative test set T. For the sub-sample empirical risk over M, the loss on the test set depends on which instances have been labelled by the querying algorithm, for which there there is no adaptable stopping criterion. This is an example of inductive active learning [93], in which the goal of active learning is to classifier that generalizes well on an unknown test set. The visualisation of the loss difference is a form of visualising inductive risk. Kottke et al. [93] show a translation of an inductive active learning scenario to a transductive one, and introduce a function for stopping criteria in the transductive scenario that considers the misclassification- and annotation cost.

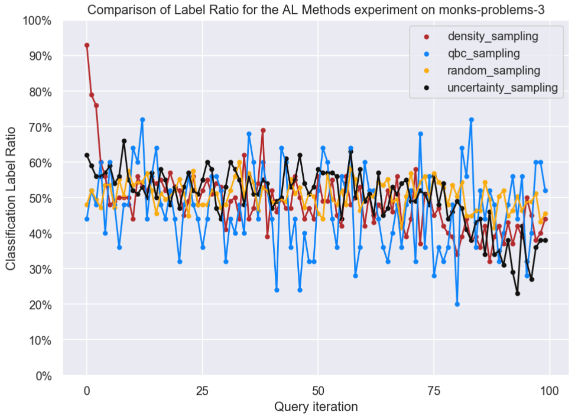


Figure 9: Example of selected label ratio graph on the monks-problem-3 dataset using random sampling, uncertainty sampling, density-weighted sampling and QBC sampling.

Figure 9 shows an example of the third sampling bias visualisation through showing the average selected label ratio over the 100 queries. This graph details the averaged ratio of labels at different points in the learning process, where a label ratio of 100% means that only positive (often the minority) class instances were selected and 0% means that only negative class instances were selected.

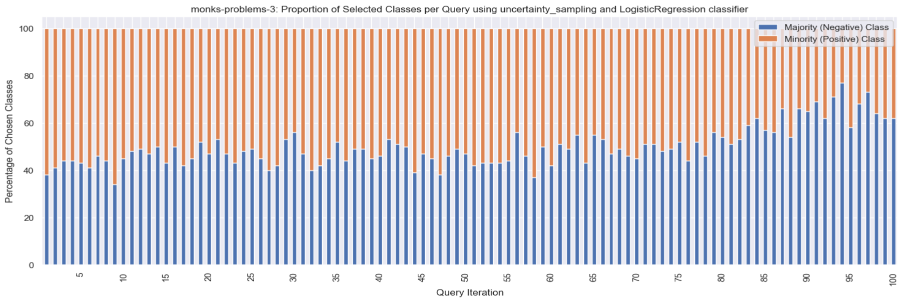


Figure 10: Selected label proportion graph on monks-problem-3 dataset using uncertainty sampling.

Finally, figure 10 shows the proportion of times either the minority or majority class was selected at each query point in the 100 iterations. Keep in mind that the selected labels at each query iteration are queried after initializing a query strategy with an initial training set of 10 labelled instances. This shows the class selection behaviour over the $k = 5$ folds and 20 executions. If one class is predominantly selected at a point in time over folds and executions, it shows the class preference at specific query iterations.

6 Experiments and Results

This section exhibits the results obtained from the various experiments conducted in this thesis. The results of the first research question are structured as follows. Each of the subsections of this task contains results for changing 1 of the 4 factors discussed in section 5.3, while keeping the other settings the same. For each of the 4 facets, first the average performance and bias results are shown across the 15 datasets. Afterwards, results on specific and noteworthy datasets are shown to give more context to certain findings. Additional results for this task are found in Appendix A. The second and third task are concerned with the active learning debiasing techniques and how they compare in terms of general performance and in higher class imbalance settings. Additional results for these tasks are given in Appendix B and D. Appendix C shows significance tests for answering the second research question.

6.1 Task 1: AL Factors and their Influence on Sampling Bias

6.1.1 Effect of Changing the AL Method

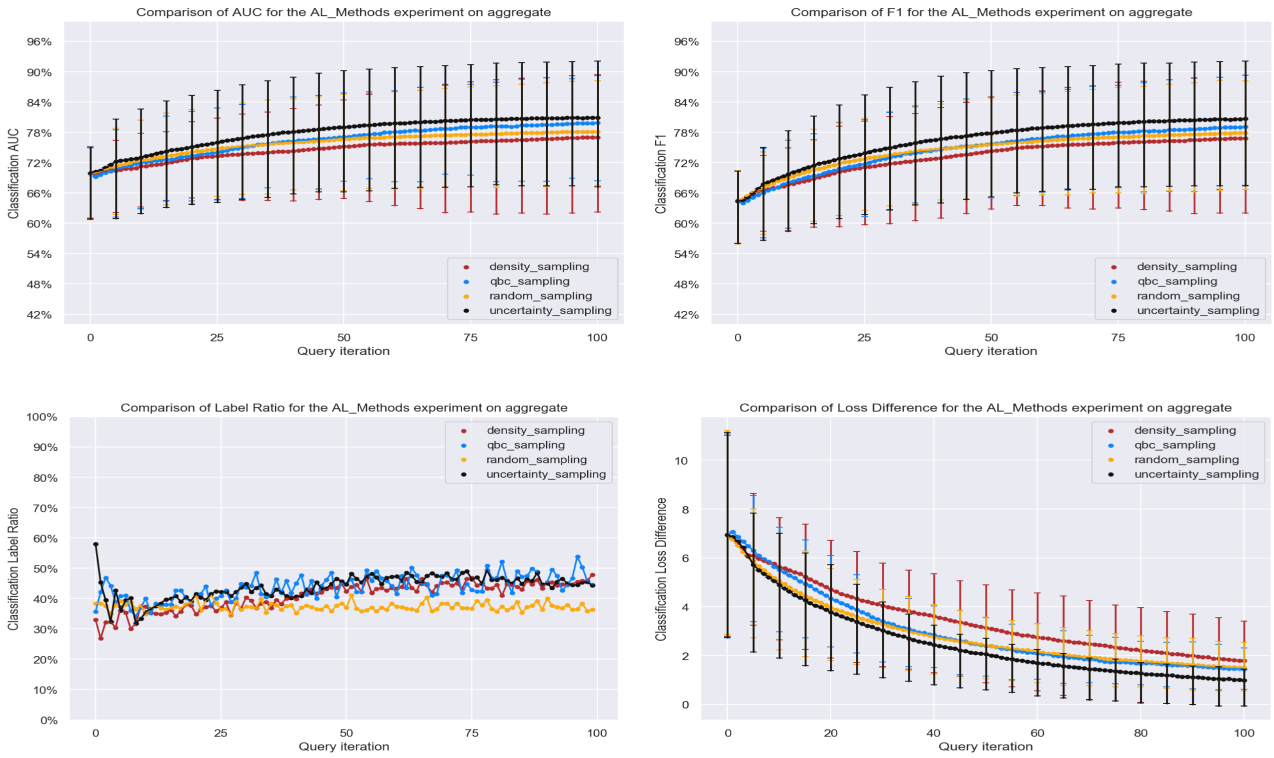


Figure 11: Average performance and bias through label ratio and loss difference for the AL method experiment over all 15 datasets. Experiments were run using logistic regression and 0.5 initial set ratio.

All results of task 1 highlighted in this section make use of the logistic regression classifier and a balanced initial set ratio, with the exception of the machine learning classifier and initial set ratio experiments. The choice for highlighting these results in this section is because both the logistic regression classifier and balanced initial set ratio yielded a high overall performance. Logistic regression yielded the highest performance for the first 50 queries and a high overall performance. See section 6.1.3 for the performance comparison between the chosen classifiers using uncertainty sampling. The balanced initial set ratio resulted in the highest and overall initial performance, see section 6.1.4.

Figure 11 shows little variance in results between varying the active learning algorithm. This experiment was also conducted with the XGBoost and random forest classifiers which yielded similar results. The results of using these classifiers are found in figures 24 and 25 in Appendix A.

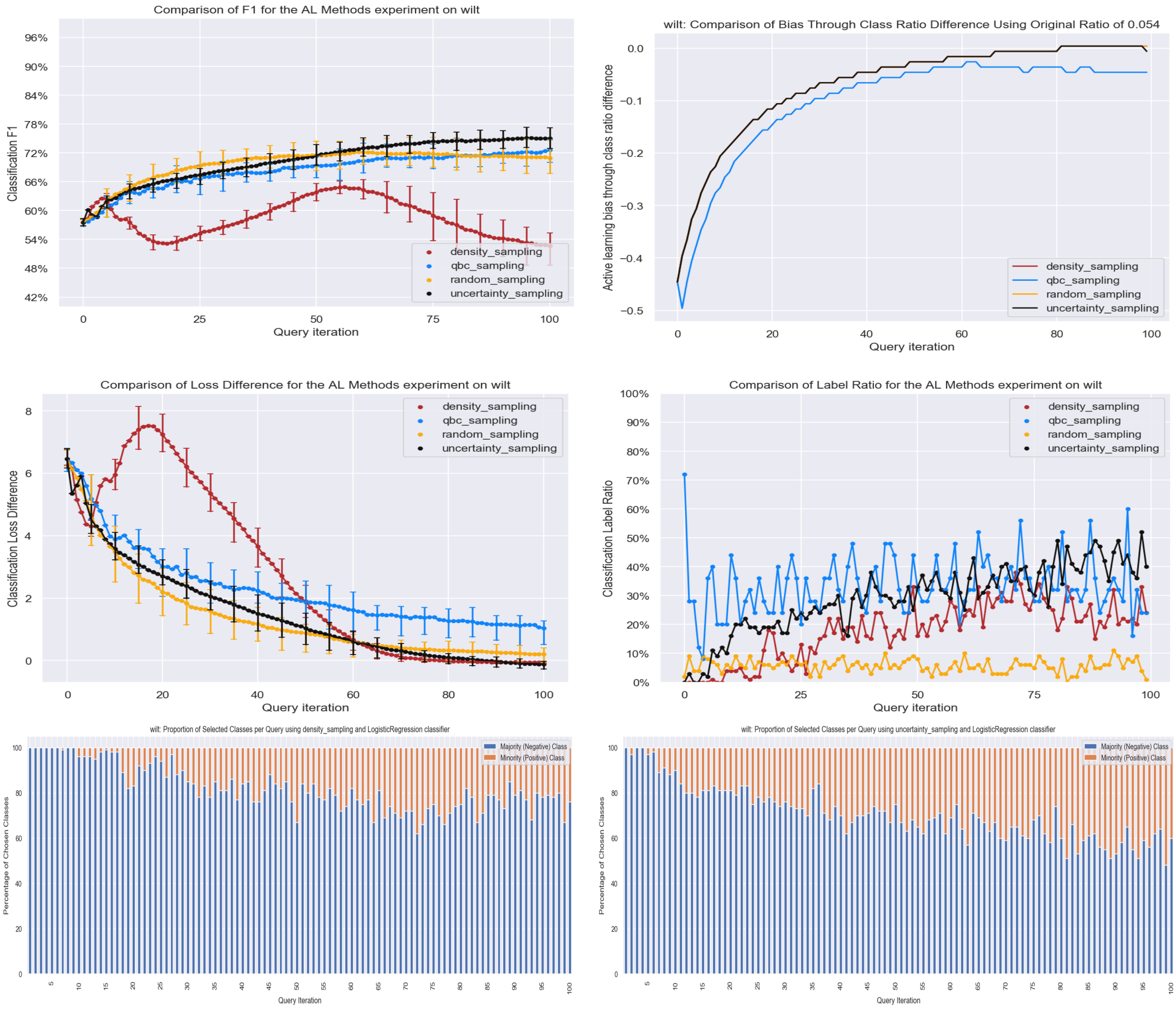


Figure 12: Performance on wilt dataset through F1 and bias visualisations. Wilt has an imbalanced class ratio of 0.054. Bottom row shows proportion of selected classes at each query iteration for density-weighted sampling on the left and uncertainty sampling on the right. These results were obtained using a logistic regression classifier and 0.5 initial set ratio.

Clearer insights in behaviour occur when looking at individual dataset results, in particular datasets with high class imbalance like wilt. The results on wilt using logistic regression are shown in figure 12. The results on the wilt dataset are comparable to the results of this experiment on other higher class imbalance datasets, like ozone-level-8hr, churn and scene. Here, informative-based uncertainty and QBC sampling query the the minority class (only 5% of the dataset) more often during the 100 query iterations than a representative-based density-weighted sampling method. The informative-based query strategies therefore introduce more sampling bias, as their querying behaviour is not in adherence with the class distribution of the original data pool. Especially during the first $15 - 20$ iterations, density-weighted sampling heavily samples the majority class, while uncertainty and QBC sampling sample the minority class more often. This behaviour is logical, as a representative-based method would sample the majority class more often to create a more representative dataset. Consequently, density-weighted creates a more representative dataset through sampling the majority class more often, but does not yield an improved learning rate. However, density-weighted sampling does not lead to an increased reduction in sampling bias through loss difference. Density-weighted sampling only starts to have a higher loss difference reduction than uncertainty sampling after 60 query iterations.

The informative-based methods deem the minority class instances to be informative more often, which leads to a better performance after training classifiers on their queried datasets. In fact in all higher class imbalance datasets, either uncertainty or QBC sampling yields the best performance. This suggests that using active learning methods that mitigate sampling bias in high class imbalance situations does not lead to a better performance, as sampling the minority class instances increases the overall quality of the labelled dataset. A difference between machine learning classifiers is seen on high imbalance datasets like wilt. Using random forest or XGBoost classifiers, the minority class is sampled heavily by uncertainty sampling during the first 20 iterations. See figure 13 for the label ratio graphs for wilt using XGBoost and random forest. As QBC sampling is comprised of a committee of ensemble classifiers sampling, it also exhibits this behaviour in figure 12. Section 16 provides further elaboration on these differences.

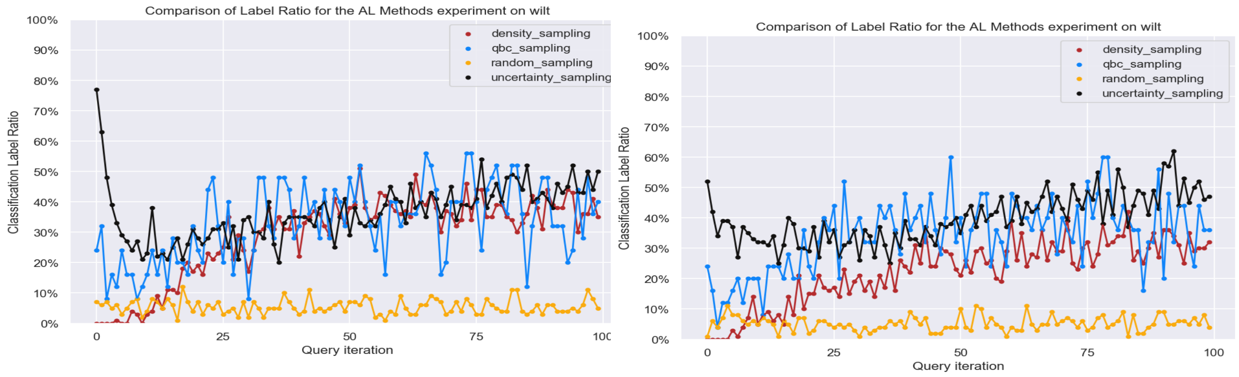


Figure 13: Wilt Selected Label Ratio graphs using XGBoost (left) and random forest (right) classifiers.

6.1.2 Effect of Various Degrees of Class Imbalance

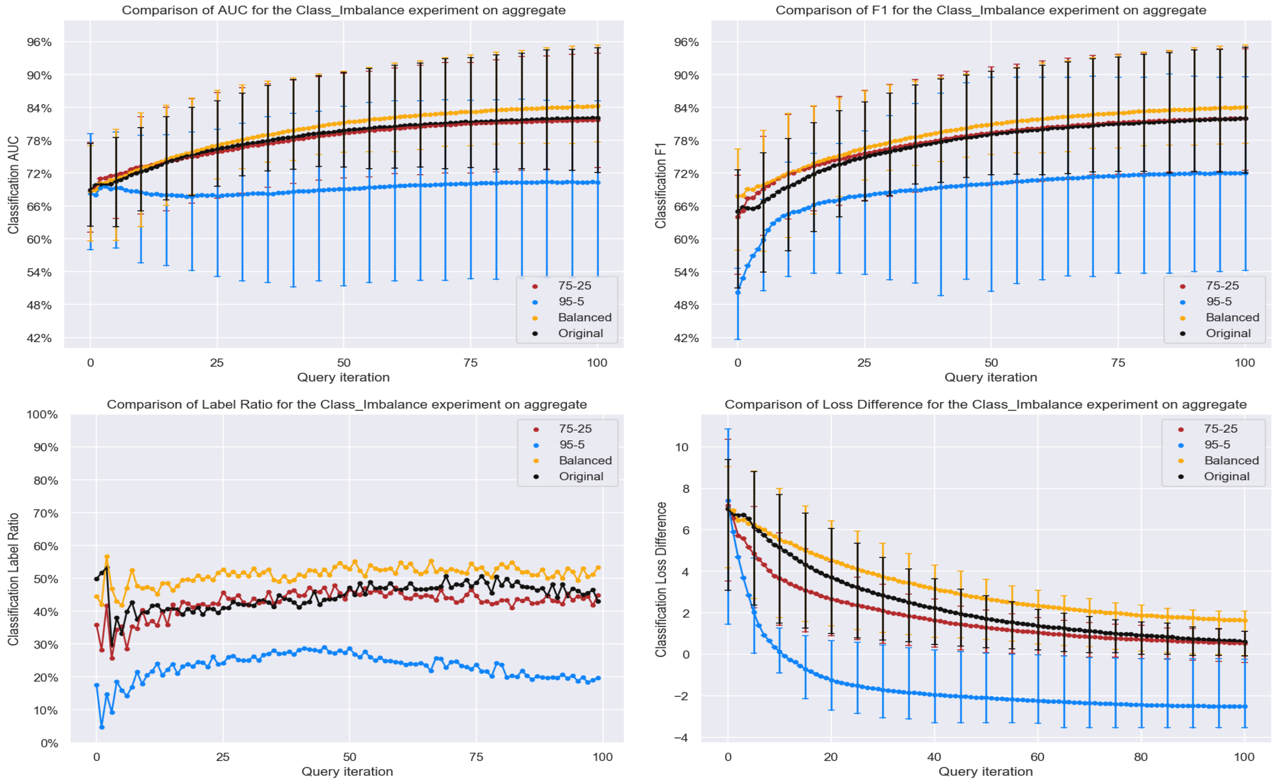


Figure 14: (Macro) AUC and F1 over query iterations over all 15 datasets. Bottom row is querying behaviour and sampling bias through Label Ratio and Loss Difference. These experiments were run using uncertainty sampling, a logistic regression classifier and a 0.5 initial set ratio.

The average results for the class imbalance experiment suggest larger differences in performance and sampling bias than the seen in the active learning method experiment. Especially using the high $95 - 5$ class imbalanced data results in generally low performance, barely increasing over time, and a skewed labelled dataset in which the minority class is represented for about 20%. However, this is an improvement when compared to the original class ratio of 0.05. Another interesting result of this experiment is the tendency for the informative-based algorithms to sample the minority class more often in high class imbalance situations when using an XGBoost or random forest classifier. This behaviour is similar to that seen in the active learning method experiment on the wilt dataset in figures 12 and 13. Seemingly, the minority class instances seem to have receive a higher information value than the majority class instances, even if the unlabelled pool contains high imbalance. Both XGBoost and random forest classifiers exhibited this behaviour, whereas logistic regression did not. More of this behaviour is visible in the ML classifier experiment in section 6.1.3. Presumably, this querying preference is due to a difference in using ensemble versus single classifiers an in their initial decision thresholds. Interestingly, querying using a balanced 0.5 class ratio yields a slightly better performance than using the original dataset. This might be due to the query strategy having equal choice between classes, which can counteract the query strategy oversampling one class and learning idiosyncrasies of the data in high imbalance situations.

Especially on datasets with lower amounts of instances, the effects of higher levels of class imbalance through undersampling negatively affects performance and querying behaviour as less instances are available for querying and training. The qsar-biodeg exemplifies these results as a smaller dataset. Therefore results on this dataset are shown in figure 15.

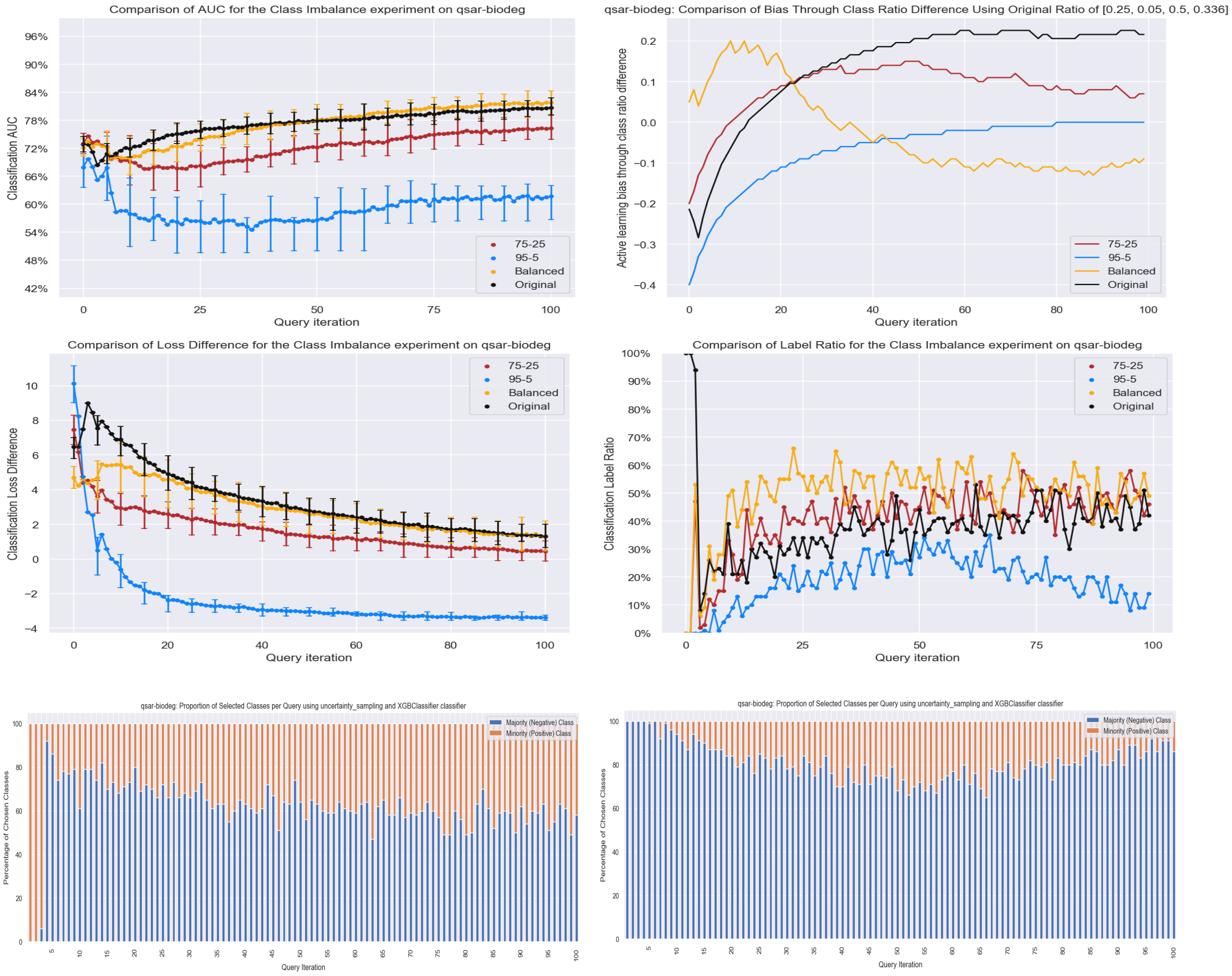


Figure 15: Performance on qsar-biodeg dataset (original class ratio 0.048) in the class imbalance experiment. Bottom row shows proportion of selected classes at each query iteration on original data on the left and 0.05 class ratio data on the right. These results were obtained using a logistic regression classifier, uncertainty sampling and 0.5 initial set ratio.

For qsar-biodeg, the AUC score for the $95 - 5$ imbalanced data is lower than other subsets and never reaches a similar performance. This performance difference is similar for all other datasets in this experiment. This experiment was also conducted with the XGBoost classifier and uncertainty sampling and density sampling, see figures 27 and 28 in Appendix A. Figure 29 in Appendix A shows a comparison between uncertainty and density-weighted sampling on the hill-valley dataset using logistic regression.

When looking at figure 15, and particularly at the class ratio difference and loss difference graphs, there seem to be two contradicting findings about the 0.05 class ratio set. The class ratio difference graph seems to suggest that high class imbalance leads to less sampling bias, as the difference approaches 0. However, this finding must not be viewed in isolation, as looking at the loss difference graphs shows that this subset yields the highest loss difference with a fully trained classifier. In fact all subsets seem to result in some bias through uncertainty sampling, as querying all subsets does not immediately yield a loss difference or population risk difference of 0. The $75-25$ subset results in the least sampling bias in terms of a combination of loss difference and class ratio difference. However, the original and balanced subsets are the best performing versions of the qsar-biodeg dataset, outperforming the $75-25$ and $95-5$ subset.

6.1.3 Effect of Changing the ML Classifier

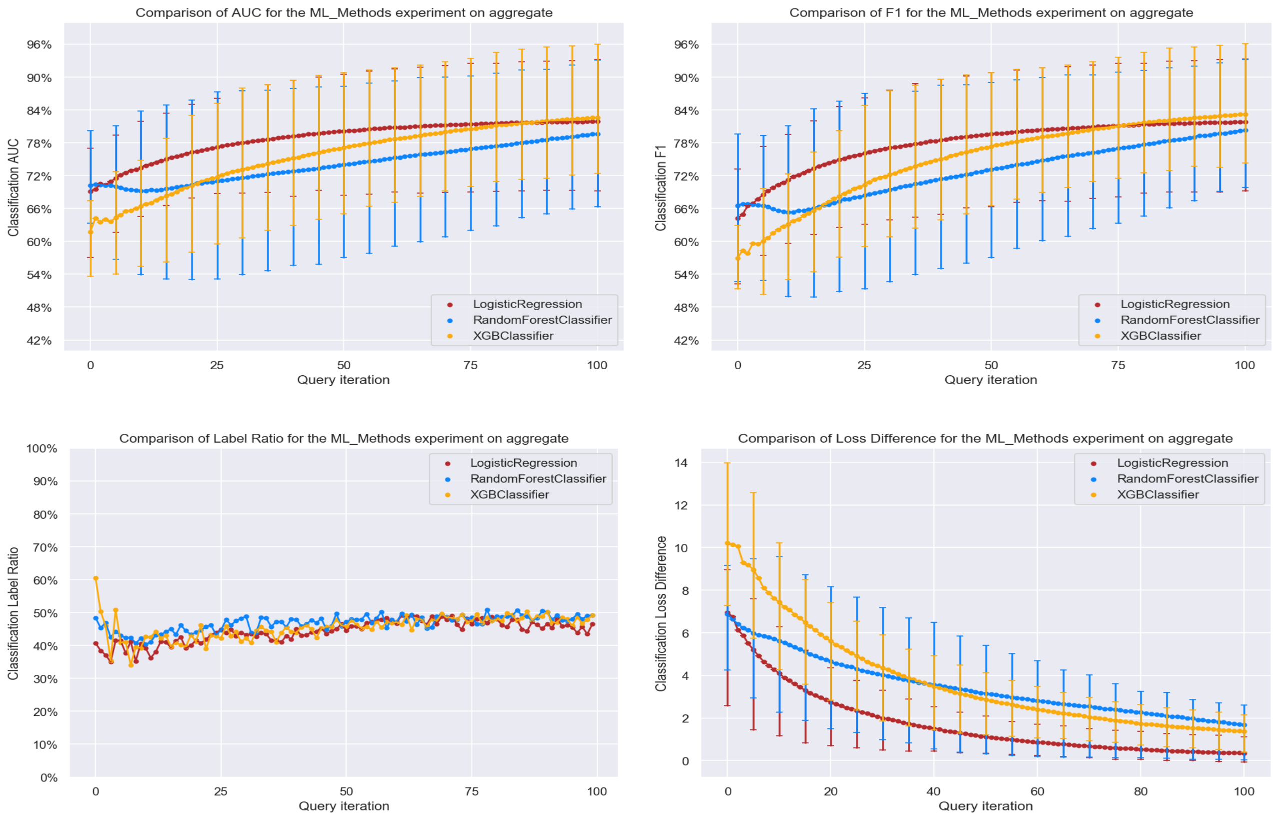


Figure 16: (Macro) F1, Recall, and Precision over query iterations over all 15 datasets. Bottom row is querying behaviour and sampling bias through Label Ratio and Loss Difference. Experiments on ML classifier conducted using uncertainty sampling and 0.5 initial set ratio.

Changing the ML classifier did not make a noteworthy difference in average sampling bias and classifier performance after 100 queries, as shown in figure 16. However, there are distinct differences between the classifiers when looking at individual dataset results like in figure 17. The differences in initial performance in figure 16 and other classifier experiments are due to the choice of classifier. This experiment was also conducted using density-weighted sampling, see figures 30 and 31 in Appendix A for the average results using this active learning algorithm and a comparison between uncertainty sampling and density-weighted sampling on the Bioresponse dataset. The same patterns emerged when changing the query strategy, where using uncertainty sampling yields slighted better overall results. Logistic regression has a higher learning rate for the first 60 query iterations, after which XGBoost reached a slightly higher performance. When looking at many individual dataset results, using logistic regression also resulted in a faster learning rate but slightly worse performance after 50 − 60 iterations when compared to using XGBoost. The exception to this was the hill-valley dataset, where logistic regression consistently outperformed the other classifiers.

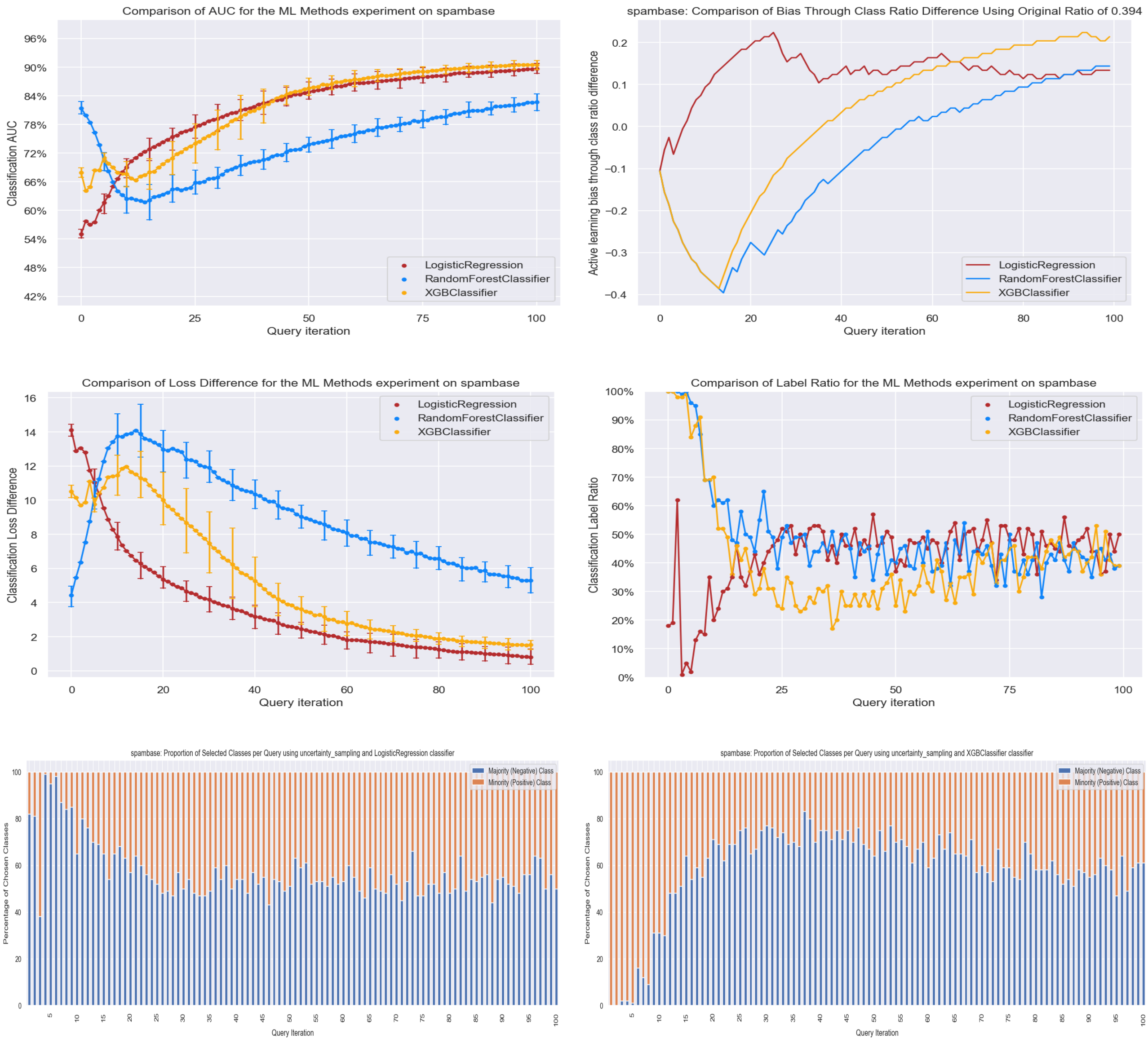


Figure 17: Performance on spambase dataset (class ratio of 0.394) in the ML classifier experiment. Bottom row shows proportion of selected classes using the logistic regression classifier on the left and the XGBoost classifier on the right. These results were obtained using the original spambase dataset, uncertainty sampling and 0.5 initial set ratio.

Results of the ML classifier experiment become more clear when looking at individual dataset results. Spambase was chosen in figure 17 to highlight, as the results exhibit some interesting differences in choosing an ensemble method like a random forest classifier or a probabilistic model like logistic regression. The bias visualisations for the spambase dataset show similar patterns to the results on the kr-vs-kp, qsar-biodeg and wilt datasets. Uncertainty sampling using ensemble methods for these datasets oversamples the minority class more often during the first 20 iterations. Using logistic regression for these datasets oversamples the majority class more heavily during the first 20 iterations. Afterwards, using logistic regression results in an almost equal proportion of sampling the minority and majority classes. This classifier also shows no dip in performance during the first 20 queries, whereas using the ensemble classifiers does yield this decrease in performance. For the churn and steel-plates-fault datasets, the majority class is oversampled by the ensemble methods. It seems the ensemble methods have a different starting decision boundary than the logistic regression classifier, leading them to oversample instances of one class more often during the first iterations. When analyzing the spambase results, this oversampling of minority instances leads to a dip in performance, especially for the random forest classifier.

6.1.4 Effect of Initial Set Ratio

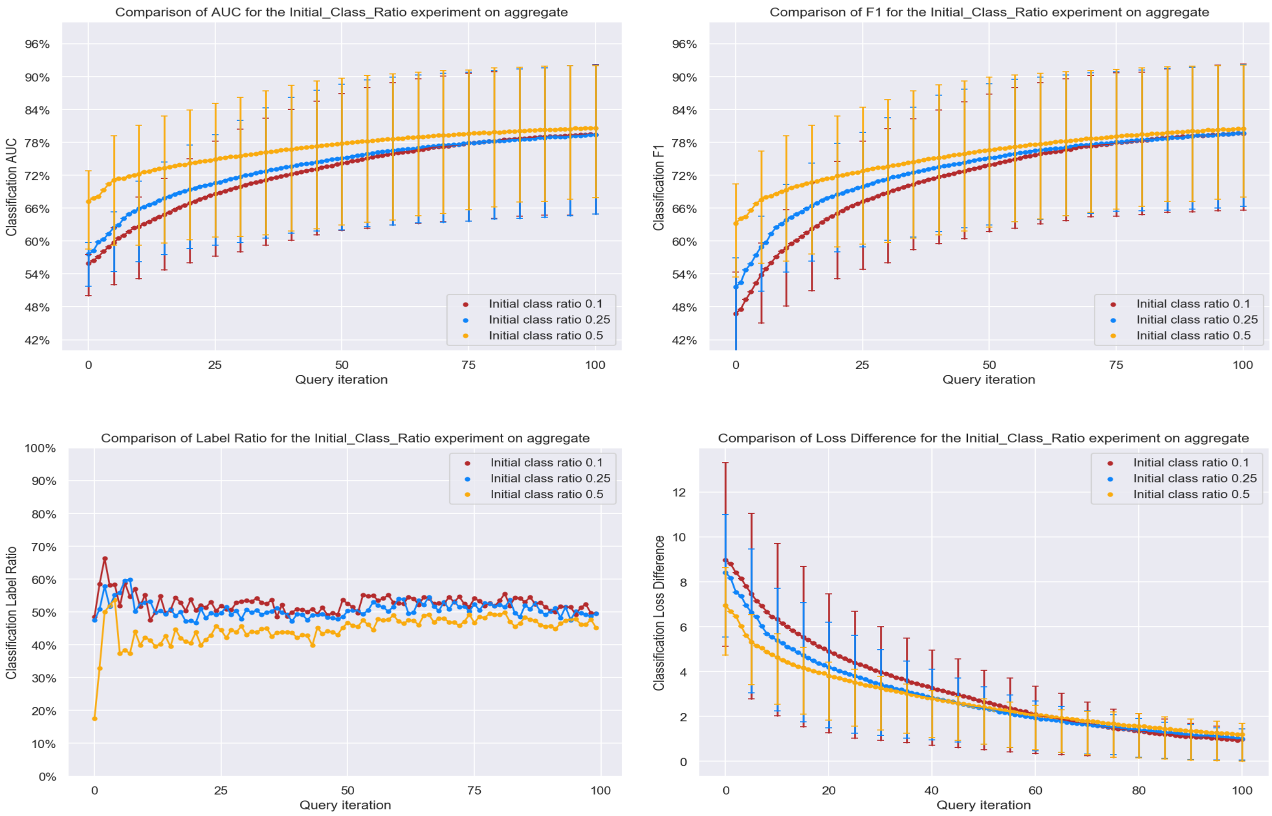


Figure 18: (Macro) AUC, F1, label ratio and loss difference over query iterations over all 15 datasets for the initial set ratio experiment. This experiment was conducted using a logistic regression classifier and uncertainty sampling.

Changing the initial set ratio, the class ratio of the original 10 instances shows some differences in performance and early querying behaviour. Figure 18 shows the average results of this experiment with the logistic regression classifier and uncertainty sampling. Using a balanced initial set ratio of 0.5 results in the highest starting performance and a slightly higher overall performance throughout the 100 queries. Other initialization sets detrimentally affect starting performance. The performance in the learning curve using these sets needs to catch up to the balanced set. Interestingly, query strategies using these imbalanced initial training sets seem to query the minority class slightly more often during the first 50 queries, as can be seen from the label ratio graph. It seems that using imbalanced initial training sets causes the query strategy to make up for this imbalance by querying the minority class more often, no matter the machine learning classifier or query strategy they use. See figure 32 of Appendix A for the results of the initial set ratio experiment using XGBoost as classifier and uncertainty sampling as query strategy. See figure 33 for the results of this experiment using density-weighted sampling and the logistic regression classifier. After the 100 query iterations, there is little difference in performance between using the balanced and 0.25 initial set ratio. However, as the goal for applying active learning is to get a high performance in as few queries as possible, using a balanced initial set ratio is the preferred option. This is why in the other experiments of this section and the experiments conducted using the AL debiasing algorithms, the balanced initial set ratio is used. The difference in initial performance between the balanced and other initial set ratios persists in results of all 15 datasets.

The results on the scene dataset are shown as an example in figure 19. For the scene dataset, the difference in both performance and sampling behaviour between the balanced and other initial set ratios persists more clearly throughout the 100 queries. Using the balanced initial set ratio for this dataset also creates the most representative labelled dataset, as seen from the class ratio and loss difference graphs. Querying using the other initial set ratios makes the class ratio of the labelled set overshoot the bias difference with the original class ratio by sampling the minority class more often. Using the balanced initial set ratio samples both classes evenly, as seen in the bottom right graphs of figure 19. This lowers the difference of the class ratio of the labelled set and the original class ratio and for the scene dataset also brings about a higher performance.

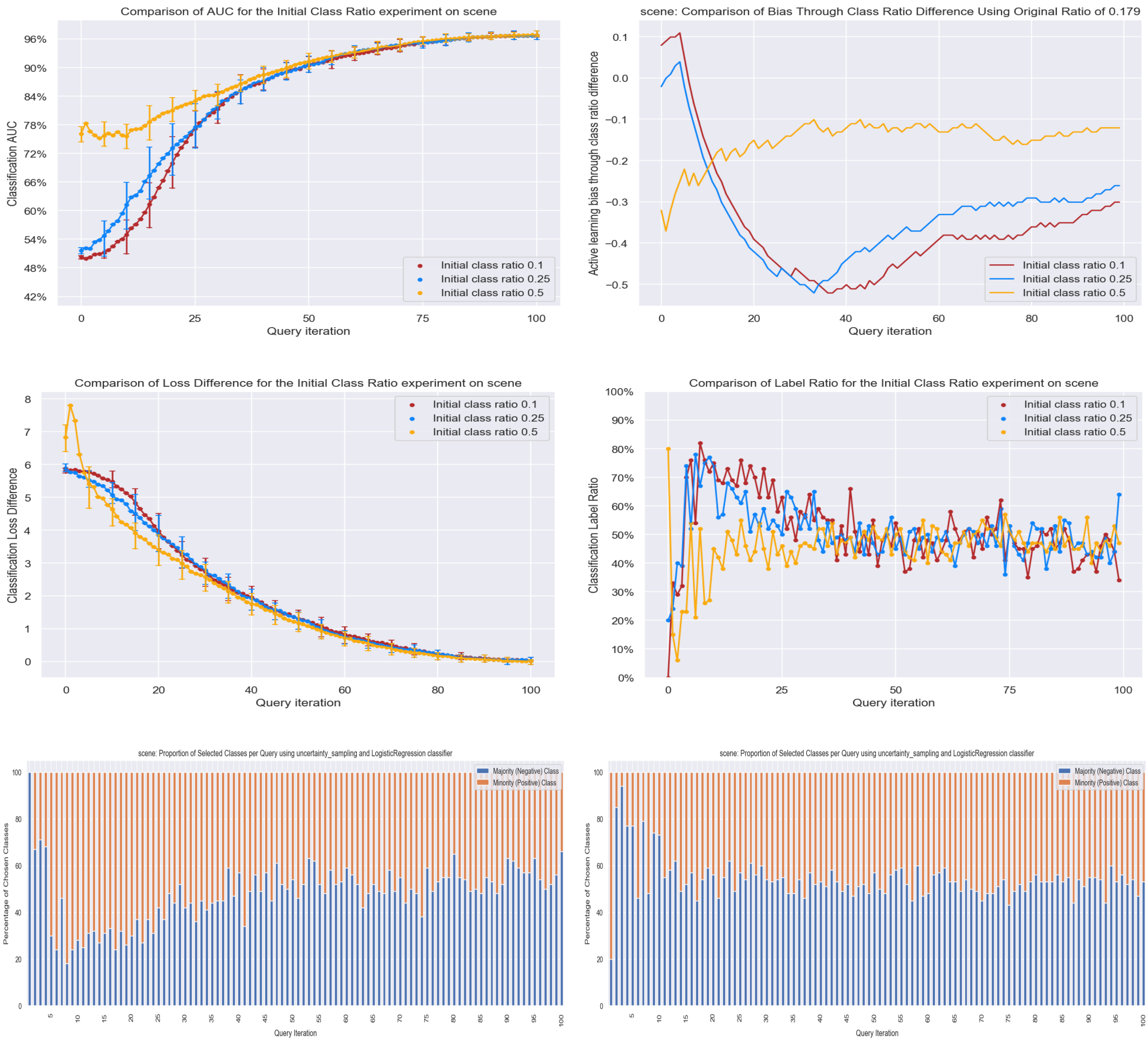


Figure 19: Performance on the scene dataset (class ratio of 0.178) in the init set ratio experiment. Bottom row shows proportion of selected classes using 0.1 init set ratio on the left and 0.5 on the right. These results were obtained using uncertainty sampling and logistic regression.

6.2 Task 2: General Performance Comparison AL Debiasing Algorithms

This section shows the comparison between the different chosen active learning debiasing algorithms on the 15 datasets distributed by OpenML [80]. Figure 20 shows a average performance comparison.

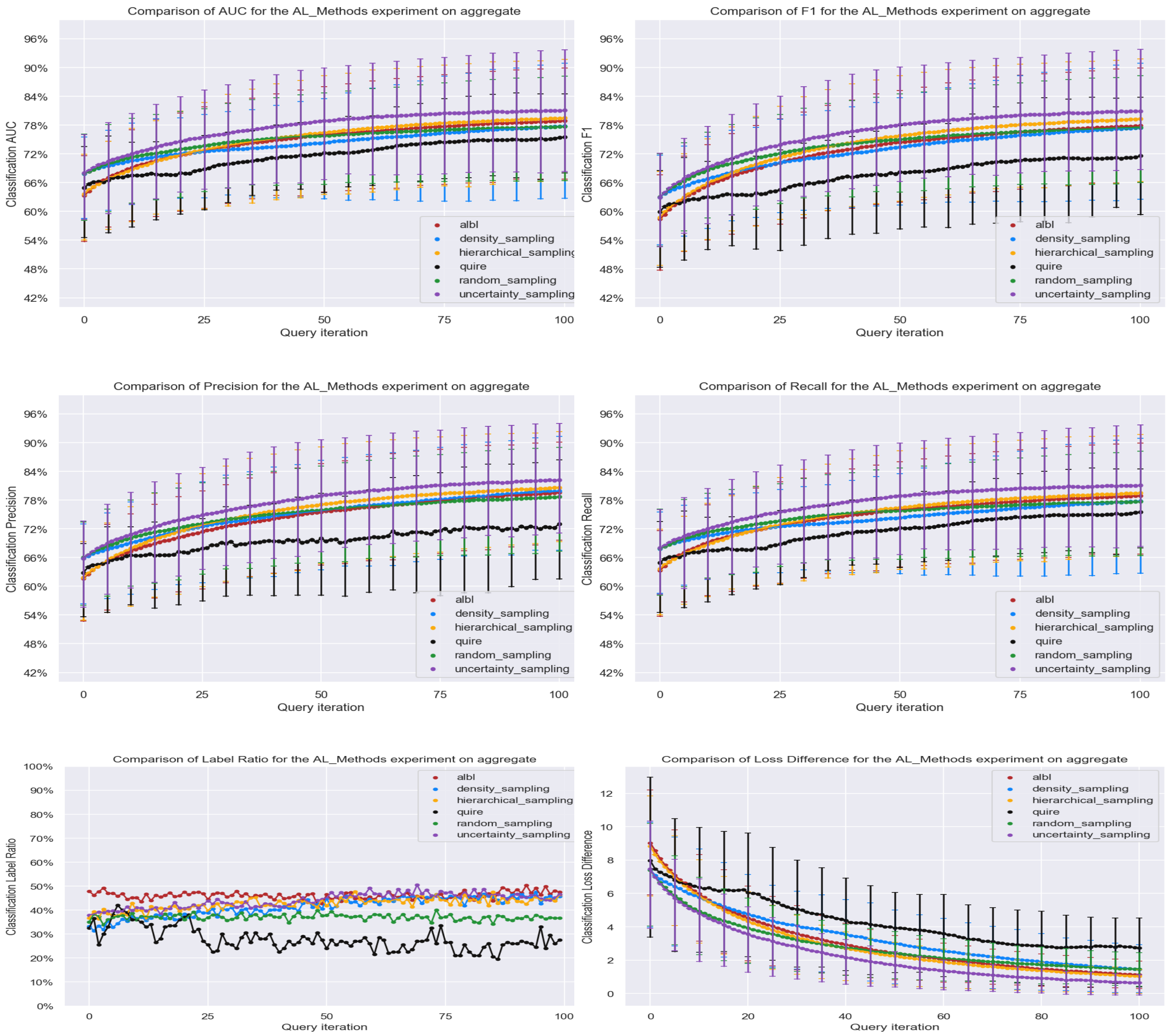


Figure 20: Average performance on 15 datasets of ALBL, QUIRE and hierarchical sampling compared to random sampling, uncertainty sampling and density-weighted sampling. Experiment was conducted using 0.5 initial set ratio and a logistic regression classifier.

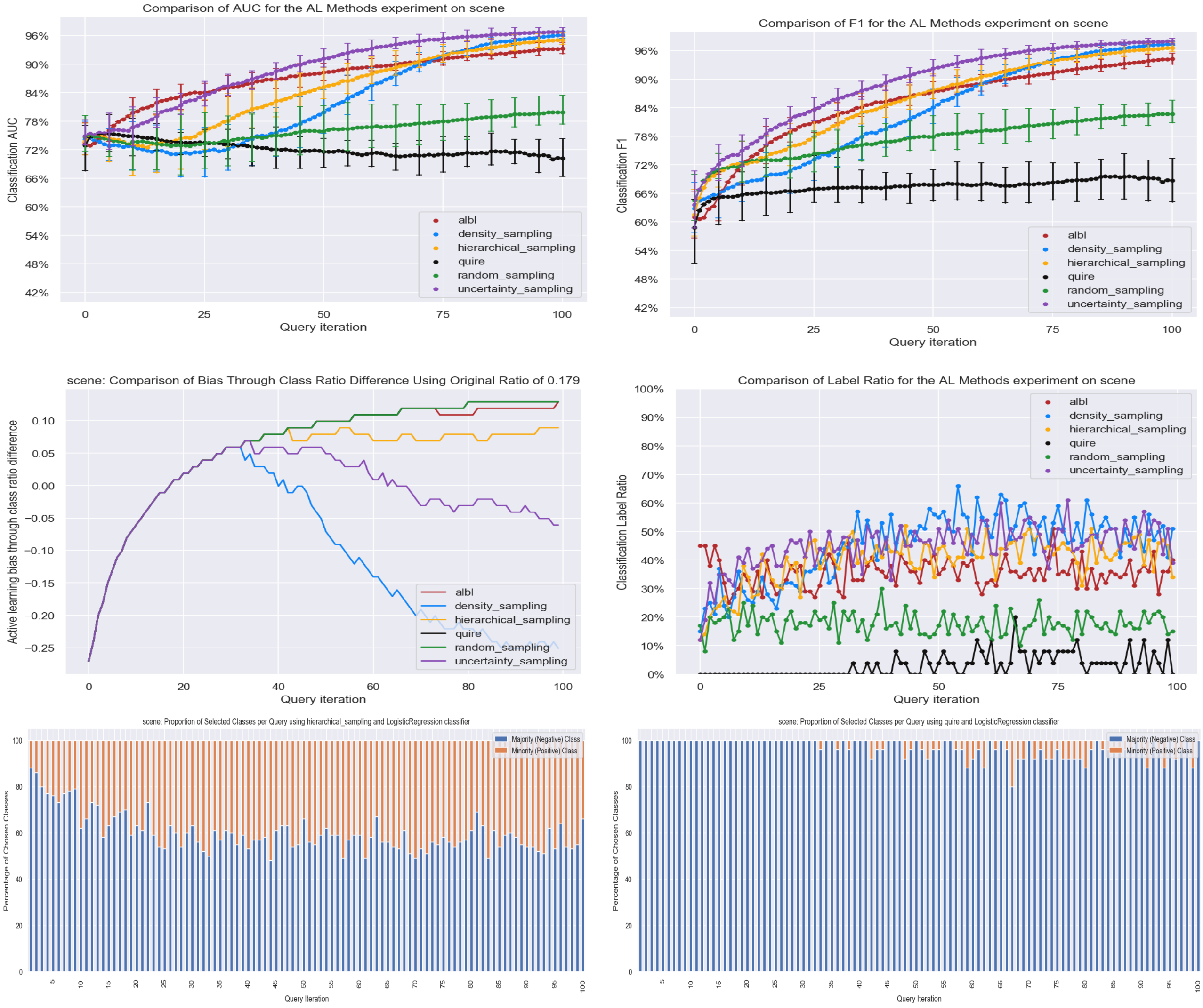
Figure 20 shows the average results of the different debiasing strategies when compared to random, uncertainty and density-weighted sampling. Table 6 shows the performance differences in terms of ALC score on the 15 datasets (8 for QUIRE). These results were generated using logistic regression as the classifier for all query strategies. The differences in average performance are similar when running this experiment using a random forest or XGBoost classifiers for the non-debiasing algorithms. Appendix B shows the results of this experiment using an XGBoost classifier for uncertainty sampling and density-weighted sampling. Logistic regression is used as classifier for the debiasing methods. On average, the debiasing strategies do not outperform uncertainty sampling. Hierarchical sampling achieves the second-best performance, having a macro AUC of around 0.02 lower than uncertainty sampling from 25 queries onwards. QUIRE performs the worst out of all the AL debiasing algorithms, resulting in a worse average performance than random (passive) sampling. ALBL performs slightly better than QUIRE and performs best on the wilt dataset, but still results on average in a slightly lower AUC than hierarchical sampling. The performance of ALBL is due to the chosen configuration (see 5.2), which combines informative- and representative-based querying. For QUIRE, all kernels were tested on the 8 QUIRE specific datasets in which the radial basis function outperformed the others. Even so, QUIRE generally performs worse than the other debiasing methods in terms of ALC score. It also was the slowest query strategy in terms of computational speed. On the jasmine dataset, the QUIRE algorithm took 3 days and 21 hours to complete the querying process over 5 folds and 5 executions. This experiment took the longest of all experiments run in this thesis. More hyperparameter tuning on each individual dataset is necessary to improve QUIRE's performance.

Dataset	Random	Uncertainty	Density-Weighted	Hierarchical	ALBL	QUIRE
monks-problems-3	0.543	0.652	0.664	0.609	0.555	0.506
qsar-biodeg	0.564	0.573	0.544	0.566	0.566	0.484
hill-valley	0.552	0.577	0.641	0.412	0.410	0.524
banknote-authentication	0.947	0.977	0.967	0.963	0.974	0.899
steel-plates-fault	0.128	0.124	0.168	0.126	0.069	0.013
scene	0.523	0.783	0.643	0.689	0.743	0.434
ozone-level-8hr	0.306	0.280	0.289	0.223	0.230	0.355
jasmine	0.435	0.503	0.457	0.443	0.447	0.247
kr-vs-kp	0.627	0.743	0.607	0.744	0.658	NA
Bioresponse	0.225	0.229	0.176	0.197	0.184	NA
spambase	0.647	0.686	0.480	0.699	0.659	NA
wilt	0.506	0.569	0.252	0.475	0.588	NA
churn	0.139	0.079	0.094	0.042	0.079	NA
mushroom	0.679	0.764	0.660	0.689	0.651	NA
PhishingWebsites	0.724	0.772	0.630	0.702	0.659	NA

Table 6: Performance comparison of chosen debiasing methods and random, uncertainty and density-weighted sampling through ALC score for all datasets. All query strategies use logistic regression as classifier and a balanced initial set ratio. Yellow cells show the highest ALC score for a dataset.

Uncertainty sampling yields the highest ALC score on most datasets, followed by density-weighted sampling and hierarchical sampling. Appendix C shows the results of applying the Wilcoxon signed-rank significance test on these ALC scores for the first 8 datasets of table 6. QUIRE was not applied to the other 7 datasets. This choice is explained in section 5.2. Uncertainty sampling generally has a statistically significant performance difference as compared to the other query strategies on these datasets. However, as detailed in section 6.1, uncertainty sampling biases the training dataset. If this is unwanted, hierarchical sampling and density-weighted sampling are the best performing options. However, if sampling bias does not cause major problems in terms of classification performance and fairness, the results indicate that uncertainty sampling is the best option.

To understand the debiasing methods in terms of the bias they attempt to mitigate, individual dataset results need to be studied. The results on the scene dataset are shown in figure 21.


Comparison of AUC for the AL Methods experiment on scene

Comparison of F1 for the AL Methods experiment on scene

scene: Comparison of Bias Through Class Ratio Difference Using Original Ratio of 0.179

Comparison of Label Ratio for the AL Methods experiment on scene

scene: Proportion of Selected Classes per Query using hierarchical_sampling and LogisticRegression classifier

scene: Proportion of Selected Classes per Query using quire and LogisticRegression classifier

Figure 21: Performance and bias visualisation graphs on the scene dataset using the 3 debiasing methods. Density-weighted sampling, uncertainty sampling and random sampling are also shown as baseline comparisons. The scene dataset contains a class ratio of 0.18, a medium to high class imbalance. Bottom row shows proportion of selected classes at each query iteration for hierarchical sampling on the left and QUIRE sampling on the right. These results were obtained using a logistic regression classifier and 0.5 initial set ratio.

When applying the query strategies to the scene dataset, QUIRE in particular values representativeness over information value when querying instances in higher class imbalance datasets. This can be seen in the label ratio and proportion of classes selected by QUIRE. The consequence of QUIRE almost exclusively sampling the majority class is a negligible improvement of the F1 score. The AUC score when using QUIRE actually worsens over time, whereas it improves when using other query strategies. The other AL debiasing methods perform better by sampling the minority class more often. In terms of measuring sampling bias through class ratio difference, the labelled datasets acquired by using debiasing methods approach the 0.18 class ratio, but never reach a class ratio disparity of 0. The closest class ratio difference of 0.05 was achieved by using hierarchical sampling. Therefore, hierarchical sampling is the best performance debiasing method in terms of both performance and mitigation of sampling bias on the scene dataset.

6.3 Task 3: Comparison of AL debiasing algorithms on Class Imbalanced Data

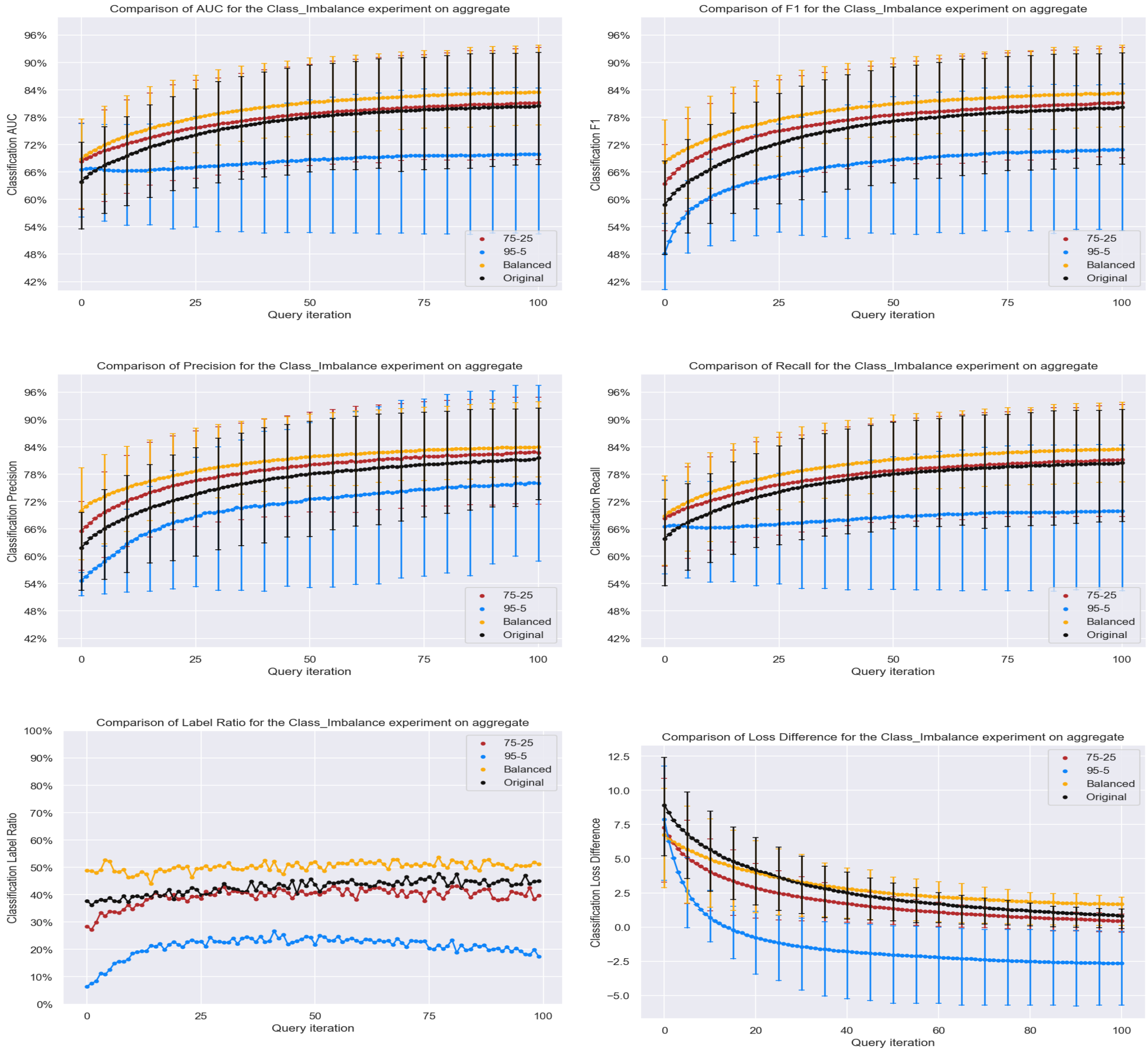


Figure 22: (Macro) AUC, F1, label ratio and loss difference over query iterations over all 15 datasets when applying hierarchical sampling and logistic regression to imbalanced data.

Figure 22 shows the average results for the class imbalance experiment using hierarchical sampling. For the sake of brevity, this section highlights the results for hierarchical sampling only. Although QUIRE and ALBL yielded similar patterns in their results. Figures 44 and 45 of Appendix D show these results. Looking at the average performance results of the different subsets when using hierarchical sampling, the differences in performance are similar to when using uncertainty sampling, see figure 14. In fact, both the AUC and F1 scores are lower when using hierarchical sampling (or any debiasing method) than when using uncertainty sampling for the $95-5$ class imbalance ratio. Therefore, it may be concluded that using debiasing techniques in the active learning cycle does not yield similar results on the same datasets with varying degrees of class imbalance. Results on all individual datasets have been studied in order to verify this conclusion. For all datasets, applying debiasing techniques to subsets with higher class imbalance did not result in similar performance to the application of those techniques on the original dataset.

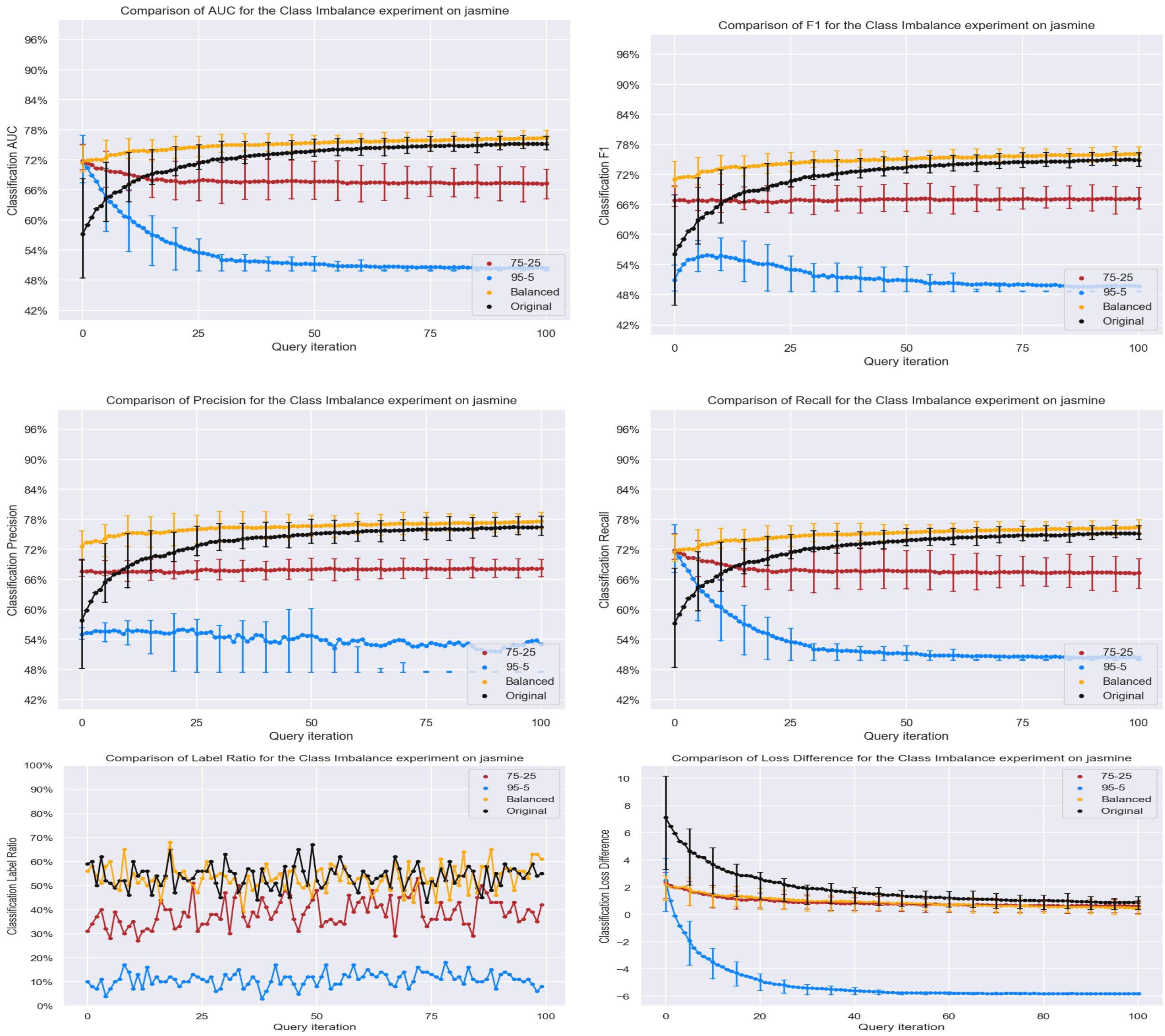


Figure 23: (Macro) AUC, F1, precision and recall as well as the label ratio and loss difference over 100 query iterations for the jasmine dataset (class ratio 0.5) when applying hierarchical sampling and logistic regression to imbalanced data.

Figure 23 shows the results of applying hierarchical sampling on the jasmine dataset. Using the balanced subset of the jasmine dataset, equal or sometimes better performance than using the original data pool was achieved. This is similar to many of the other 14 datasets. The performance worsens when using the $75 - 25$ and $95 - 5$ subsets. When looking at the label ratio graph, a difference in querying behaviour between debiasing algorithms and informative-based algorithms is visible. When using debiasing methods, the label ratio is more skewed towards the majority class than when using uncertainty sampling on the $75 - 25$ and $95 - 5$ subsets. This behaviour is logical when viewing the debiasing methods as informative- and representative-based query strategies. Figure 23 shows that hierarchical sampling queries more representative instances on imbalanced data. This algorithm will therefore query more of the majority class instances than an informative-based algorithm, like uncertainty sampling. It can be concluded that the use of a debiasing method results in a more representative dataset and less sampling bias in terms of class bias. However, when looking at the learning curves, the performance of the classifier trained on this dataset worsens over each query iteration. These results, as well as results on other datasets for the class imbalance experiment, support previous findings i.e. introducing some sampling bias in the labelled dataset can be beneficial when dealing with high class imbalance situations. By querying the minority class more often using informative-based query strategies in high class imbalance settings, the resulting dataset is more balanced, leading to improved performance. However, when looking at the difference in class ratio, a more balanced training dataset contains sampling bias as compared to the original imbalanced data pool. Note however that the examples given in section 2.3.4 show that sampling bias can lead to unfair and potentially harmful incorrect predictions. This difference in querying behaviour is also clearly visible for the mushroom dataset: figure 46 in Appendix D shows the difference in querying behaviour between using uncertainty sampling and using hierarchical sampling.

7 Conclusions

This thesis researched two subjects related to active learning and active learning sampling bias:

1. *The influence of multiple factors in the active learning pipeline on active learning sampling bias.* To study sampling bias in the active learning pipeline, this thesis conducted empirical research by using 15 datasets. Section 6.1 is concerned with the effect of changing four integral parts of the active learning pipeline on sampling bias and performance. Specifically, the settings of the active learning query strategy, the level of class imbalance in the data pool, the machine learning classifiers and the class ratio of the initial training set were modified.

2. *The effect of AL debiasing algorithms on sampling bias reduction and classification performance.* The latter part of this research studied three informative- and representative-based query strate-

gies which all aim to reduce active learning sampling bias. These three methods, dubbed active learning debiasing methods, were hierarchical sampling, QUerying Informative and Representative Examples (QUIRE) and Active Learning By Learning (ALBL). The performance comparison of these debiasing methods was both a general performance comparison on the same datasets and a performance comparison on imbalanced versions thereof. The effect on operational performance by using these debiasing methods was compared to using more conventional query strategies like uncertainty sampling and density-weighted sampling.

The following section will go through each research question and will answer them by summarising key findings in the results of the various experiments conducted in this thesis. Section 7.1 will then provide a more detailed discussion of these findings and the limitations of this research. Section 7.2 will suggest possible future research topics based on insights provided by this thesis.

Research question 1:

- *What factors in the active learning cycle influence active learning sampling bias in the labelled dataset?*

To answer this question fully, the findings of the four experiments with the chosen variables in the active learning cycle will be summarised:

Effect of changing the AL method

Changing the active learning algorithm did not result in large differences in average performance and sampling bias. However, clearer differences in performance and querying behaviour were visible when looking at individual dataset results. On datasets with higher levels of class imbalance, differences appeared between the various query strategies. These strategies were, on the one hand, the informative-based uncertainty sampling and QBC sampling, and on the other hand, the more representative-based density-weighted sampling. The informative-based query strategies queried instances of the minority class more often, especially during the first 20 queries. Density-weighted sampling queried more representative instances throughout the learning process, but this resulted in a slower learning rate. However, all active learning methods did outperform random sampling on most datasets. The experiments on datasets with higher levels of class imbalance showed that choosing uncertainty or QBC sampling led to the highest performance. So when sampling bias in the labelled dataset matters less, uncertainty or QBC sampling (both strategies that can potentially introduce sampling bias) are recommended. In cases when sampling bias needs to be avoided, implementing density-weighted sampling is the safest option. The results of this experiment showed that after around $40-50$ queries, density-weighted sampling yielded a higher performance than random sampling when using ensemble classifiers. The performance of density-weighted sampling was marginally lower but still comparable to the informative-based methods. The question whether sampling bias is truly harmful is dependent on the application case. If it is harmful, a more representative form of active learning querying is required.

Effect of various degrees of class imbalance

Changing the class imbalance made the largest difference in querying behaviour and had the largest effect on sampling bias in the resulting labelled dataset. Using a higher level of class imbalance in the same dataset also led to the a worse performance. Interestingly, using a balanced class ratio yielded a better performance than using the original dataset class ratio. This might be due to query strategies having an even choice between both classes, resulting in a faster learning rate of the correct decision boundary. In particular the 0.05 class ratio subset resulted in the worst performance and most sampling bias through loss difference. Feeding the 0.25 class ratio subset to the pipeline resulted in the most unbiased selection in terms of loss difference. Using uncertainty sampling on the imbalanced subsets resulted in the minority class being sampled more often: around 20% throughout learning. When using density-weighted sampling on the subsets, the overall label selection probabilities were more in line with the class ratio in the data pool. This is because in high class imbalance situations, the majority class instances will be in more dense areas of the feature space, so density-weighted sampling will query these instances more often. The oversampling of the minority class for uncertainty sampling led to sampling bias, but a better performing classifier when trained on the dataset. This experiment showed that using active learning to remove some imbalance from a dataset but introducing sampling bias can sometimes be beneficial to classifier performance.

Effect of changing the ML classifier

Using logistic regression, XGBoost and random forest classifiers all had an influence on sampling bias and querying behaviour. The main difference in querying behaviour between using these classifiers was visible during the first $20-25$ queries. The ensemble methods queried the minority class much more often during these first query iterations when the dataset contained higher levels of class imbalance (≤ 0.2). When the datasets were more balanced (≥ 0.3), the ensemble methods sometimes oversampled the majority class during the first 25 query iterations. This oversampling of one class sometimes led to a decrease in performance, especially for the random forest classifier. It also led to a higher class distribution disparity between labelled dataset and data pool during the first query iterations. The oversampling of instances of a single class at the start of the learning process is due to a difference in single classification and ensemble classification. The initial decision boundary of the single classifier and the initial decision trees of the ensemble methods differed greatly, which led to different ideas of the most informative instance when using an informative-based algorithm. In this case, during the first query iterations query scores using the ensemble classifiers gave the highest information value to instances of a single class. After querying these instances more often, the ensemble methods assigned a similar information value to instances of both classes. For the query strategies using logistic regression, which is a single classifier, instances of both classes were closest to the decision boundary during the first 20 query iterations.

Effect of changing the initial set ratio

In terms of sampling bias there was one noticeable difference in querying behaviour during the first 40 query iterations. The active learning algorithms initialised with initial set ratios 0.1 and 0.25 queried the minority class slightly more often than the algorithms using the balanced initial set ratio. This suggests that the active learning algorithm will try to address this imbalance by querying the minority class more often, due to a lack of minority class instances in the initial set of 10 instances. There was also a noticeable difference in initial performance. Using a balanced initial set ratio yielded a much higher initial performance and a slightly better overall performance. As query strategies using the balanced initial training set queried the minority class less aggressively, using this set also yielded a more representative training set in general. Consequently, the labelled dataset using this balanced initial training set contained less sampling bias through class disparity and loss difference.

Research question 2:

- *How do different active learning sampling bias mitigation techniques compare when looking at operational classification performance?*

All sampling bias mitigation techniques in this thesis performed worse than uncertainty sampling. Of the three debiasing methods, hierarchical sampling performed the best and was the fastest both in terms of learning rate and computational speed. Hierarchical sampling performed only marginally worse than uncertainty sampling, followed by ALBL and finally QUIRE. The problems with ALBL and QUIRE in terms of learning rate might be due to the hyperparameters chosen for these methods. This research made the choice to use 3 different uncertainty, 2 density-weighted uncertainty and 1 random sampling method as query strategies for ALBL. The use of other query strategies or different settings for the chosen query strategies could lead to a better learning rate. However, more hyperparameter tuning is necessary to find the best performing setups for both ALBL and QUIRE. For QUIRE, the radial basis function kernel was used with a gamma of 1. For every dataset, the kernel, gamma value and other hyperparameters need to be set accordingly to result in the highest QUIRE performance. As this was a general study, the overall highest performing kernel was chosen for QUIRE. However, this choice resulted in insufficient performance when compared to the other debiasing methods. QUIRE would often only sample the majority class in imbalanced datasets, leading to high representativeness in the labelled dataset but a barely improving performance. Hierarchical sampling and ALBL prioritised the information-value of instances more than QUIRE, which led to better performance throughout the learning process. In active learning cases, it is important to first consider whether sampling bias through active learning is harmful or not. When it is harmful, the results of the experiments conducted with the debiasing strategies (see table 6) showed that the use of hierarchical or density-weighted sampling is preferable. However, when highest performance is the most important, uncertainty sampling is the logical choice. Uncertainty sampling had the highest learning rate and ALC score on most of the 15 datasets, and showed a statistically significant improvement in ALC score when compared to the debiasing methods on the first 8 datasets (see Appendix C).

Research question 3:

- *Will the influence of active learning sampling bias mitigation techniques yield similar results on the same dataset with varying degrees of class imbalance?*

Using the three debiasing methods did not result in similar performance on varying levels of class imbalance on the same dataset. The differences in performance were similar to using uncertainty sampling, QBC sampling or density-weighted sampling. Class imbalance of class ratio 0.05 still yielded the lowest performance and a balanced subset resulted in the highest performance when applying the debiasing methods to the datasets. The performance on the extreme class imbalance subset was worse when applying the debiasing methods than when applying uncertainty sampling. When fed the high class imbalance subset, the debiasing methods sampled the majority class more often than uncertainty sampling. This led to a lower learning rate for the debiasing methods. The learning rate was similar to that exhibited by applying density-weighted sampling to the class imbalanced subsets. The similarity of the performance and sampling bias results between the chosen debiasing methods and density-weighted sampling is logical, because density-weighted sampling also aims to mitigate sampling bias. However, similar to the results of research question 2 in section 6.2, the results in section 6.3 showed that the use of uncertainty sampling also led to a higher performance on imbalanced data than the debiasing methods. To understand the true effectiveness of more representative-based active learning algorithms on highly imbalanced data, more experimentation with different active learning algorithms and different hyperparameter settings is necessary.

7.1 Further Discussion and Limitations

This section will delve deeper in discussion of the results of this thesis' experiments and will go over some of the limitations of this research.

During computation of the experiments, except for the results detailed in section 6.1, the choice was made to generate a different initial training set at each fold in the 5-fold cross-validation. This was subsequently adjusted for the results of task 1 in section 6.1: the initial training set is now randomly determined once for each dataset, with size 10 and initial set ratio 0.5, except for the initial set ratio experiment. For the initial set ratio experiment, the initial training set is kept similar by generating a training set of size 10 and initial set ratio 0.5 for each dataset. Then for lower initial set ratios, minority class instances in the same initial training set are replaced with majority class instances. The choice of using identical initial training sets will ensure that all variations of experiments, e.g. the choice of AL method, will use the same initial training set, allowing the random choice of initial training set not to interfere with the results of the experiment. However, time did not allow to make this adjustment for all other experiments. Thankfully, the overall differences between the various settings of the experiments were the same without this adjustment made, so making the adjustment did not completely change the results. In the future, all experiments, with the exception of the initial set ratio experiment, will be rerun with this adjustment made.

Next to studying the effects of choices in the active learning pipeline on sampling bias through class bias, the results of the experiments conducted in this thesis can also provide guidance for implementation choices in future active learning research. The results of this thesis can be used to guide certain implementation decisions when choosing between the machine learning classifier, the active learning method, the level of balance in the dataset and the level of balance in the initial training set. Next to showing what effect these facets have on sampling bias through class bias, this thesis has also shown the effect on classification performance on a variety of datasets. Therefore, this research has shown what settings are preferable in certain situations. Of course, when making these choices for a new active learning pipeline on a new dataset, it is useful to experiment with a variety of settings before choosing a final structure. However, the results of this thesis can inform the initial choices of settings, algorithms and whether to use informative-based or more representative-based algorithms.

This thesis studies sampling bias in active learning strictly through the disparity of class distribution between the labelled dataset and original data pool. However, this is not the only form of sampling bias, as differences in feature distribution between the labelled data and the original can also result in sampling bias. As this thesis was a study on 15 different datasets, all with different feature spaces, the choice was made to study sampling bias through difference in class distribution and loss, as these factors could be studied through more general and less dataset dependent experiments. However, visualisations of the feature space and instance queries within this feature space can yield key insights in sampling bias through active learning. The choice to omit this kind of visualisation was necessary for this thesis and its form of experimentation, but limits the conclusion in its results. This is because feature selection through querying plays an important part in sampling bias in training datasets for classification.

To study the effect of various factors on active learning sampling bias, the choice was made to study four facets of the active learning pipeline and to experiment with specific parameters. The combination of multiple settings per experiment resulted in a large set of experiments for the first research question, in which a large number of differences in settings became apparent. However, it is a possible limitation in this research that only four facets were studied and furthermore only specific settings within those facets were covered. A certain added setting to a part in the pipeline or experiments on an entirely different section of the pipeline could produce new insight or challenge the findings of the results described in this thesis.

The choices of hyperparameters for the QUIRE and ALBL in this study have been made by testing the best overall performing settings on all 15 datasets for ALBL and 8 datasets for QUIRE. For QBC sampling, one of the best performing combinations in [2] was used. Since the focus of this thesis was sampling bias and operational performance, this way of finding well-performing versions of these algorithms was sufficient. However, if these methods are to be applied to a specific dataset another evaluation of hyperparameter settings will be necessary.

7.2 Future Work

The conclusions drawn from the results of this research can be used as advice for future IDlab research on the application of active learning to ILT inspection cases. When applying different query strategies to the 15 datasets, using uncertainty sampling resulted in the highest overall performance but this method did sample the minority class more often which resulted in a biased dataset. When applying active learning to a new IDlab study, careful consideration should be taken in order to determine whether sampling bias is welcome or harmful. In situations with high class imbalance and a limited inspection budget (≤ 40 instances) like the zeezwaaien study [2], the focus is on obtaining the highest performance. In both the zeezwaaien study and in the active learning method experiment in section 6.1.1, it was proven that sampling bias in the labelled dataset through oversampling the minority class can boost classification performance by creating more balanced training data. In these cases, results from this thesis supports the use of either QBC or uncertainty sampling using a balanced initial training set and either logistic regression or a random forest classifier. However, when sampling bias is detrimental to performance and raises issues of fairness, using hierarchical sampling or density-weighted sampling as more representative-based methods is recommended. These query strategies reduced sampling bias in the labelled dataset while still yielding a high operational performance, yet still a slightly lower performance than uncertainty sampling.

As detailed in the previous section, one of the limitations of this thesis is that sampling bias through active learning querying was studied exclusively through studying class bias. While this was done to allow for a larger empirical study with many different datasets, it will also be useful to look at sampling bias through active learning in terms of feature bias. Visualizing feature bias via graphs and heatmaps of the feature space and a query strategies selections within that space might produce new findings in the behavioural differences between informative- and representative-based query strategies.

The performance results of the machine learning classifier experiment showed a higher performance for the logistic regression classifier for the first $60 - 70$ queries. After 70 queries, the XGBoost classifier yielded a higher performance. In future research, it would be interesting to experiment with a combination of a logistic regression and XGBoost classifier to see if this would improve performance. A cut-off point could be used as a hyperparameter, which could be the amount of queries necessary with one classifier before switching classification to the other classifier.

For the performance comparison of more representative-based active learning algorithms, the choice was made to compare 3 informative- and representative-based algorithms on 15 datasets in a pool-based active learning scenario. While in this experiment uncertainty sampling outperformed the debiasing algorithms, further research can provide new and contrasting evidence of the performance of active learning algorithms that seek to create more representative labelled datasets. The inclusion of purely representative-based query strategies in the comparison is worth researching, as these methods can offer interesting insights into the differences in querying behaviour between the types of query strategy. For IDlab related research for risk-based inspections, it would be interesting to compare multiple different types of query strategies in a stream-based setting. This setting would be more applicable to the detection of non-compliance while there is still time to act, like in the en route classification of zeezwaaien in [2].

When studying the effect of class imbalance on active learning sampling bias, the learning rate of the classifiers worsened greatly on higher levels of class imbalance. The low performance when feeding the high class imbalance subsets to the query strategies might be due to the lack of any class imbalance mitigation method beforehand. In many cases when training classifiers on highly imbalanced data, the developer chooses a strategy for addressing class imbalance. These strategies include oversampling the minority class either by copying minority class instances or synthetically creating new ones, or by rebalancing or reweighing class importance. For the class imbalance experiments in this thesis, a naive approach was taken to applying active learning algorithms to highly imbalanced data as the focus of this thesis was on the behaviour of the active learning algorithms. For future research, studying a combination of a rebalancing strategy and active learning could offer new insights into this situation.

References

[1] Punit Kumar and Atul Gupta. "Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey". In: *Journal of Computer Science and Technology* 35 (2020), pp. 913–945.

[2] Samuel Meyer. *Investigating the Use of Active Learning for Classification of Ship Waste Dumping in the North Sea.* 2021.

[3] Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. "Internationaal Verdrag Ter Voorkoming van Verontreiniging Door Schepen, 1973, Zoals Gewijzigd Door Het Protocol van 1978 Daarbij , Londen, 02-11-1973". In: *(visited on 28/02/2022)* (1973). URL: `https://wetten.overheid.nl/BWBV0003241/2021-01-01#Verdrag_2_VerdragtekstII_3`.

[4] Shpat Cheliku. "APPLYING AI-BASED ANOMALY DETECTION TECHNIQUES TO IDENTIFY WASTE DISCHARGERS AT THE NORTH SEA". In: (2020).

[5] Thomas M. Mitchell. *Machine Learning.* 1st ed. USA: McGraw-Hill, Inc., 1997. ISBN: 0070428077.

[6] Konstantina Kourou et al. "Machine learning applications in cancer prognosis and prediction". In: *Computational and structural biotechnology journal* 13 (2015), pp. 8–17.

[7] M Ikonomakis, Sotiris Kotsiantis, and V Tampakas. "Text classification using machine learning techniques." In: *WSEAS transactions on computers* 4.8 (2005), pp. 966–974.

[8] Michael Crawford et al. "Survey of review spam detection using machine learning techniques". In: *Journal of Big Data* 2.1 (2015), pp. 1–24.

[9] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.* Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[10] Zhi-Hua Zhou, Ke-Jia Chen, and Yuan Jiang. "Exploiting unlabeled data in content-based image retrieval". In: *European Conference on Machine Learning.* Springer. 2004, pp. 525–536.

[11] Ido Dagan and Sean P. Engelson. "Committee-Based Sampling for Training Probabilistic Classifiers". In: *Proceedings of the Twelfth International Conference on International Conference on Machine Learning.* ICML'95. Tahoe City, California, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 150–157. ISBN: 1558603778.

[12] Eric Ringger et al. "Active Learning for Part-of-Speech Tagging: Accelerating Corpus Annotation". In: *Proceedings of the Linguistic Annotation Workshop.* Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 101–108. URL: `https://aclanthology.org/W07-1516`.

[13] Markus Becker and Miles Osborne. "A Two-Stage Method for Active Learning of Statistical Grammars." In: *IJCAI.* Vol. 5. Citeseer. 2005, pp. 991–996.

[14] Katrin Tomanek and Fredrik Olsson. "A Web Survey on the Use of Active Learning to Support Annotation of Text Data". In: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 45–48. URL: `https://aclanthology.org/W09-1906`.

[15] Burr Settles. "From theories to queries: Active learning in practice". In: *Active learning and experimental design workshop in conjunction with AISTATS 2010.* JMLR Workshop and Conference Proceedings. 2011, pp. 1–18.

[16] Nathalie Japkowicz and Shaju Stephen. "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5 (2002), pp. 429–449.

[17] Javad Zolfaghari Bengar et al. "Class-Balanced Active Learning for Image Classification". In: *CoRR* abs/2110.04543 (2021). arXiv: `2110.04543`. URL: `https://arxiv.org/abs/2110.04543`.

[18] Seyda Ertekin, Jian Huang, and C Lee Giles. "Adaptive Resampling with Active Learning". In: *Under Review* (2009).

[19] Sanjoy Dasgupta. "Two faces of active learning". In: *Theoretical computer science* 412.19 (2011), pp. 1767–1781.

[20] Jingbo Zhu et al. "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. 2008, pp. 1137–1144.

[21] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. "On statistical bias in active learning: How and when to fix it". In: *arXiv preprint arXiv:2101.11665* (2021).

[22] Ameya Prabhu, Charles Dognin, and Maneesh Singh. "Sampling bias in deep active classification: An empirical study". In: *arXiv preprint arXiv:1909.09389* (2019).

[23] Katrin Tomanek and Udo Hahn. "Reducing class imbalance during active learning for named entity annotation". In: *Proceedings of the fifth international conference on Knowledge capture*. 2009, pp. 105–112.

[24] Sanjoy Dasgupta and Daniel Hsu. "Hierarchical sampling for active learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 208–215.

[25] Ninareh Mehrabi et al. "A Survey on Bias and Fairness in Machine Learning". In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607.

[26] Bianca Zadrozny. "Learning and Evaluating Classifiers under Sample Selection Bias". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 114. ISBN: 1581138385. DOI: 10.1145/1015330.1015425. URL: https://doi.org/10.1145/1015330.1015425.

[27] Thomas Oommen, Laurie G Baise, and Richard M Vogel. "Sampling bias and class imbalance in maximum-likelihood logistic regression". In: *Mathematical Geosciences* 43.1 (2011), pp. 99–120.

[28] A. Albert and J. A. Anderson. "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models". In: *Biometrika* 71.1 (1984), pp. 1–10. ISSN: 00063444. URL: http://www.jstor.org/stable/2336390.

[29] Anqi Liu and Brian Ziebart. "Robust Classification Under Sample Selection Bias". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/d67d8ab4f4c10bf22aa353e27879133c-Paper.pdf.

[30] Corinna Cortes et al. "Sample selection bias correction theory". In: *International conference on algorithmic learning theory*. Springer. 2008, pp. 38–53.

[31] Punit Kumar and Atul Gupta. "Active learning query strategies for classification, regression, and clustering: a survey". In: *Journal of Computer Science and Technology* 35.4 (2020), pp. 913–945.

[32] David D Lewis and William A Gale. "A sequential algorithm for training text classifiers". In: *SIGIR'94*. Springer. 1994, pp. 3–12.

[33] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. "Query by committee". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 287–294.

[34] Ray Liere and Prasad Tadepalli. "Active learning with committees for text categorization". In: *AAAI/IAAI*. Citeseer. 1997, pp. 591–596.

[35] Ido Dagan and Sean P Engelson. "Committee-based sampling for training probabilistic classifiers". In: *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 150–157.

[36] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[37] Andrew McCallum and Kamal Nigam. "Employing EM and Pool-Based Active Learning for Text Classification". In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 350–358. ISBN: 1558605568.

[38] Naoki Abe and Hiroshi Mamitsuka. "Query Learning Strategies Using Boosting and Bagging." In: Jan. 1998, pp. 1–9.

[39] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[40] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[41] Burr Settles and Mark Craven. "An analysis of active learning strategies for sequence labeling tasks". In: *proceedings of the 2008 conference on empirical methods in natural language processing*. 2008, pp. 1070–1079.

[42] Yi Wu et al. "Sampling Strategies for Active Learning in Personal Photo Retrieval". In: *2006 IEEE International Conference on Multimedia and Expo*. 2006, pp. 529–532. DOI: `10.1109/ICME.2006.262442`.

[43] Dino Ienco et al. "Clustering based active learning for evolving data streams". In: *International Conference on Discovery Science*. Springer. 2013, pp. 79–93.

[44] Min Wang et al. "Active learning through density clustering". In: *Expert Systems with Applications* 85 (2017), pp. 305–317. ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2017.05.046`. URL: `https://www.sciencedirect.com/science/article/pii/S095741741730369X`.

[45] Nicholas Roy and Andrew McCallum. *Toward optimal active learning through sampling estimation of error reduction. Int. Conf. on Machine Learning*. 2001.

[46] Georg Krempl, Daniel Kottke, and Myra Spiliopoulou. "Probabilistic active learning: Towards combining versatility, optimality and efficiency". In: *International Conference on Discovery Science*. Springer. 2014, pp. 168–179.

[47] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[48] Georg Krempl, Daniel Kottke, and Vincent Lemaire. "Optimised probabilistic active learning (OPAL)". In: *Machine Learning* 100.2 (2015), pp. 449–476.

[49] Daniel Kottke et al. "Toward optimal probabilistic active learning using a Bayesian approach". In: *Machine Learning* 110.6 (2021), pp. 1199–1231.

[50] Burr Settles, Mark Craven, and Soumya Ray. "Multiple-instance active learning". In: *Advances in neural information processing systems* 20 (2007).

[51] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[52] Thomas Osugi, Deng Kim, and Stephen Scott. "Balancing exploration and exploitation: A new algorithm for active machine learning". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE. 2005, 8–pp.

[53] Yoram Baram, Ran El-Yaniv, and Kobi Luz. "Online Choice of Active Learning Algorithms". In: *Journal of Machine Learning Research* 5 (Dec. 2004), pp. 255–291.

[54] Wei-Ning Hsu and Hsuan-Tien Lin. "Active learning by learning". In: *Twenty-Ninth AAAI conference on artificial intelligence*. 2015.

[55] Djallel Bouneffouf et al. "Contextual bandit for active learning: Active thompson sampling". In: *International Conference on Neural Information Processing*. Springer. 2014, pp. 405–412.

[56] Shipra Agrawal and Navin Goyal. "Thompson sampling for contextual bandits with linear payoffs". In: *International conference on machine learning*. PMLR. 2013, pp. 127–135.

[57] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. "Active learning by querying informative and representative examples". In: *Advances in neural information processing systems* 23 (2010).

[58] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." In: *Journal of machine learning research* 7.11 (2006).

[59] Bo Du et al. "Exploring representativeness and informativeness for active learning". In: *IEEE transactions on cybernetics* 47.1 (2015), pp. 14–26.

[60] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. "Importance weighted active learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 49–56.

[61] Josh Attenberg and Şeyda Ertekin. "Class imbalance and active learning". In: *Imbalanced Learning: Foundations, Algorithms, and Applications* (2013), pp. 101–149.

[62] M. Kubat. “Addressing the Curse of Imbalanced Training Sets: One-Sided Selection”. In: *Fourteenth International Conference on Machine Learning* (June 2000).

[63] Gavin C Cawley. “Baseline methods for active learning”. In: *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*. JMLR Workshop and Conference Proceedings. 2011, pp. 47–57.

[64] Oscar Reyes, Abdulrahman Altalhi, and Sebastian Ventura. “Statistical Comparisons of Active Learning Strategies over Multiple Datasets”. In: *Knowledge-Based Systems* 145 (Jan. 2018). DOI: 10.1016/j.knosys.2018.01.033.

[65] Tom Fawcett. “Introduction to ROC analysis”. In: *Pattern Recognition Letters* 27 (June 2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[66] Yukun Chen, Subramani Mani, and Hua Xu. “Applying active learning to assertion classification of concepts in clinical text”. In: *Journal of Biomedical Informatics* 45.2 (2012), pp. 265–272. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2011.11.003. URL: https://www.sciencedirect.com/science/article/pii/S1532046411001912.

[67] Erelcan Yanık and Tevfik Metin Sezgin. “Active learning for sketch recognition”. In: *Computers & Graphics* 52 (2015), pp. 93–105. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2015.07.023. URL: https://www.sciencedirect.com/science/article/pii/S0097849315001260.

[68] Mervyn Stone. “Cross-validatory choice and assessment of statistical predictions”. In: *Journal of the royal statistical society: Series B (Methodological)* 36.2 (1974), pp. 111–133.

[69] Frank Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987. URL: http://www.jstor.org/stable/3001968.

[70] Ranganath Krishnan et al. “Improving Robustness and Efficiency in Active Learning with Contrastive Loss”. In: *CoRR* abs/2109.06873 (2021). arXiv: 2109.06873. URL: https://arxiv.org/abs/2109.06873.

[71] Zhao Xu et al. “Representative sampling for text classification using support vector machines”. In: *European conference on information retrieval*. Springer. 2003, pp. 393–407.

[72] Yu-Lin Tsou and Hsuan-Tien Lin. “Annotation cost-sensitive active learning by tree sampling”. In: *Machine Learning* 108.5 (2019), pp. 785–807.

[73] Huailong Dong, Bowen Zhu, and Jing Zhang. “A Cost-Sensitive Active Learning for Imbalance Data with Uncertainty and Diversity Combination”. In: *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*. ICMLC 2020. Shenzhen, China: Association for Computing Machinery, 2020, pp. 218–224. ISBN: 9781450376426. DOI: 10.1145/3383972.3384002. URL: https://doi.org/10.1145/3383972.3384002.

[74] Akshay Krishnamurthy et al. “Active learning for cost-sensitive classification”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1915–1924.

[75] Seyda Ertekin et al. “Learning on the border: active learning in imbalanced data classification”. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007, pp. 127–136.

[76] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110.3 (2021), pp. 457–506.

[77] Vu-Linh Nguyen, Sébastien Destercke, and Eyke Hüllermeier. “Epistemic Uncertainty Sampling”. In: *CoRR* abs/1909.00218 (2019). arXiv: 1909.00218. URL: http://arxiv.org/abs/1909.00218.

[78] Robin Senge et al. “Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty”. In: *Information Sciences* 255 (2014), pp. 16–29.

[79] Moksh Jain et al. “Deup: Direct epistemic uncertainty prediction”. In: *arXiv preprint arXiv:2102.08501* (2021).

[80] Joaquin Vanschoren et al. “OpenML: networked science in machine learning”. In: *SIGKDD Explorations* 15.2 (2013), pp. 49–60. DOI: 10.1145/2641190.2641198. URL: http://doi.acm.org/10.1145/2641190.264119.

[81] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[82] Kamel Mansouri et al. "Quantitative structure-activity relationship models for ready biodegradability of chemicals". In: *Journal of chemical information and modeling* 53.4 (Apr. 2013), pp. 867–878. ISSN: 1549-9596. DOI: 10.1021/ci4000213. URL: https://doi.org/10.1021/ci4000213.

[83] M Buscema, S Terzi, and W Tastle. *Steel Plates Faults Data Set*. 2010. URL: www.semeion.it.

[84] Grigorios Tsoumakas et al. "Mulan: A Java Library for Multi-Label Learning". In: *Journal of Machine Learning Research* 12 (2011), pp. 2411–2414.

[85] Kun Zhang et al. "Forecasting Skewed Biased Stochastic Ozone Days: Analyses and Solutions". In: vol. 14. Dec. 2006, pp. 753–764. DOI: 10.1007/s10115-007-0095-1.

[86] Pieter Gijsbers et al. "An Open Source AutoML Benchmark". In: *CoRR* abs/1907.00909 (2019). arXiv: 1907.00909. URL: http://arxiv.org/abs/1907.00909.

[87] Boehringer Ingelheim. *Predicting a Biological Response*. 2013. URL: https://www.kaggle.com/competitions/predicting-a-biological-response/overview.

[88] Brian Alan Johnson, Ryutaro Tateishi, and Nguyen Thanh Hoan. "A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees". In: *International Journal of Remote Sensing* 34.20 (2013), pp. 6969–6982. DOI: 10.1080/01431161.2013.810825. eprint: https://doi.org/10.1080/01431161.2013.810825. URL: https://doi.org/10.1080/01431161.2013.810825.

[89] Daniel T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. 2nd. Wiley Publishing, 2014. ISBN: 0470908742.

[90] G.H. Lincoff et al. *The Audubon Society Field Guide to North American Mushrooms*. A Chanticleer Press edition. Knopf, 1981. ISBN: 9780394519920. URL: https://books.google.nl/books?id=tsTbzYxTGcsC.

[91] Tivadar Danka and Peter Horvath. "modAL: A modular active learning framework for Python". In: (). available on arXiv at https://arxiv.org/abs/1805.00979. URL: https://github.com/modAL-python/modAL.

[92] Yao-Yuan Yang et al. *libact: Pool-based Active Learning in Python*. Tech. rep. available as arXiv preprint https://arxiv.org/abs/1710.00379. National Taiwan University, Oct. 2017. URL: https://github.com/ntucllab/libact.

[93] Daniel Kottke et al. "A Stopping Criterion for Transductive Active Learning". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD)*. (accepted). Springer, 2022.

Appendices

Appendix A: Additional Performance Results for Task 1

The following results were obtained using different settings for the various experiments in task 1. The graphs in this section show the results when varying certain settings for the 4 main experiments. Similar to the structure of section 6.1, for the different settings some overall performance graphs are shown as well as some results on certain individual datasets.

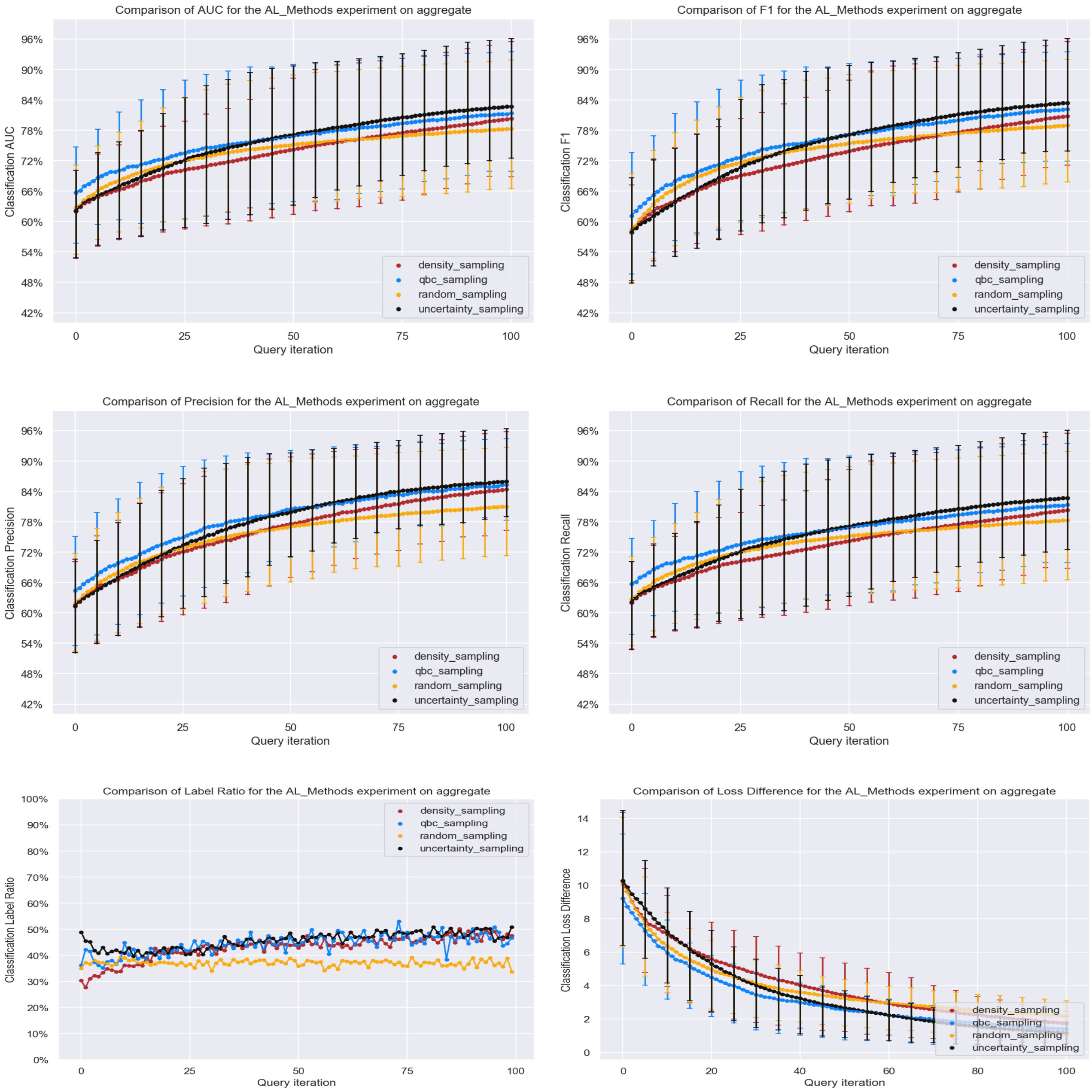


Figure 24: Average performance of AL experiment for Task 1 with 0.5 initial set ratio and using the XGBoost classifier.

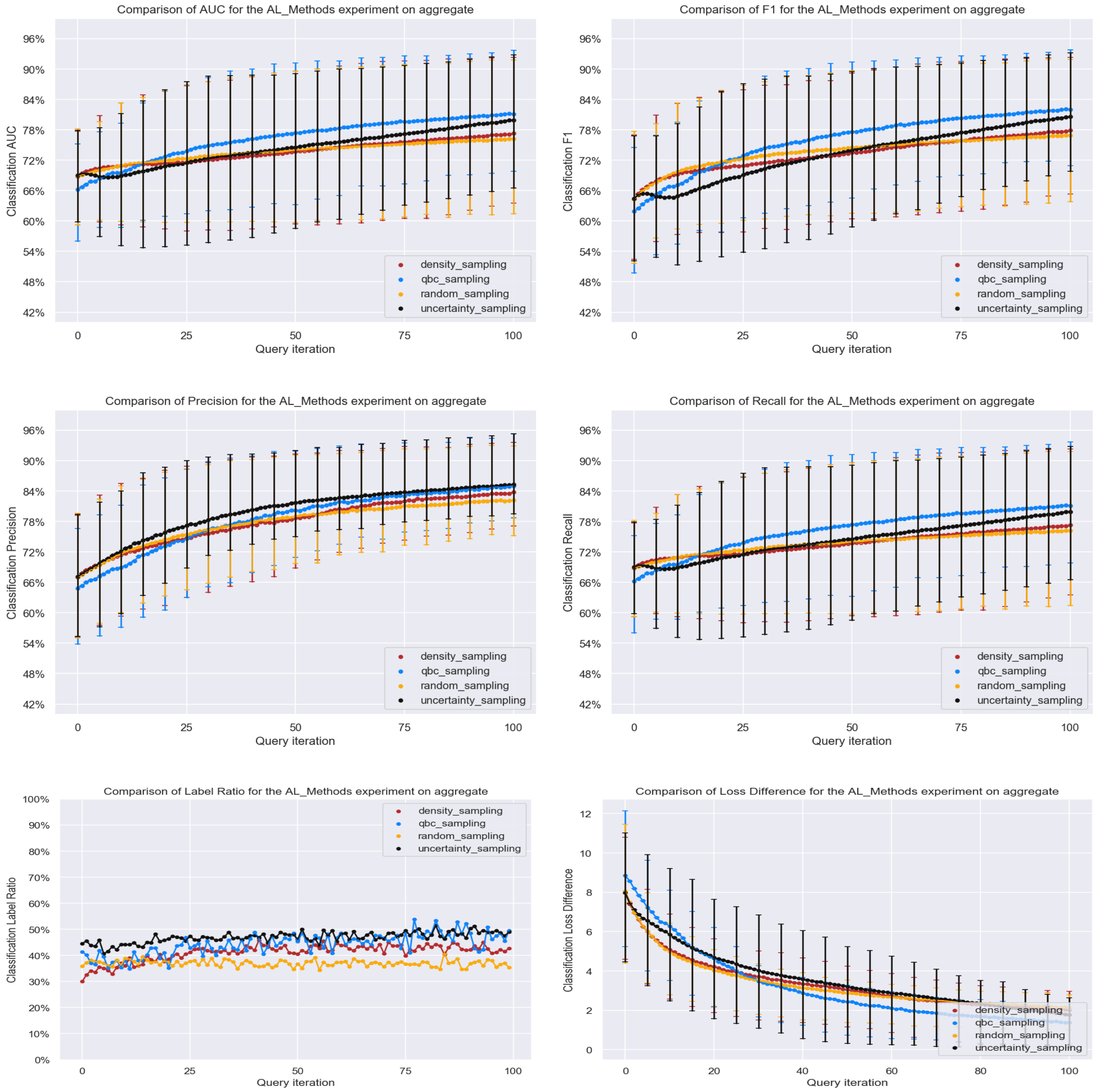


Figure 25: Average performance of the AL experiment for Task 1 with 0.5 initial set ratio and using the random forest classifier.

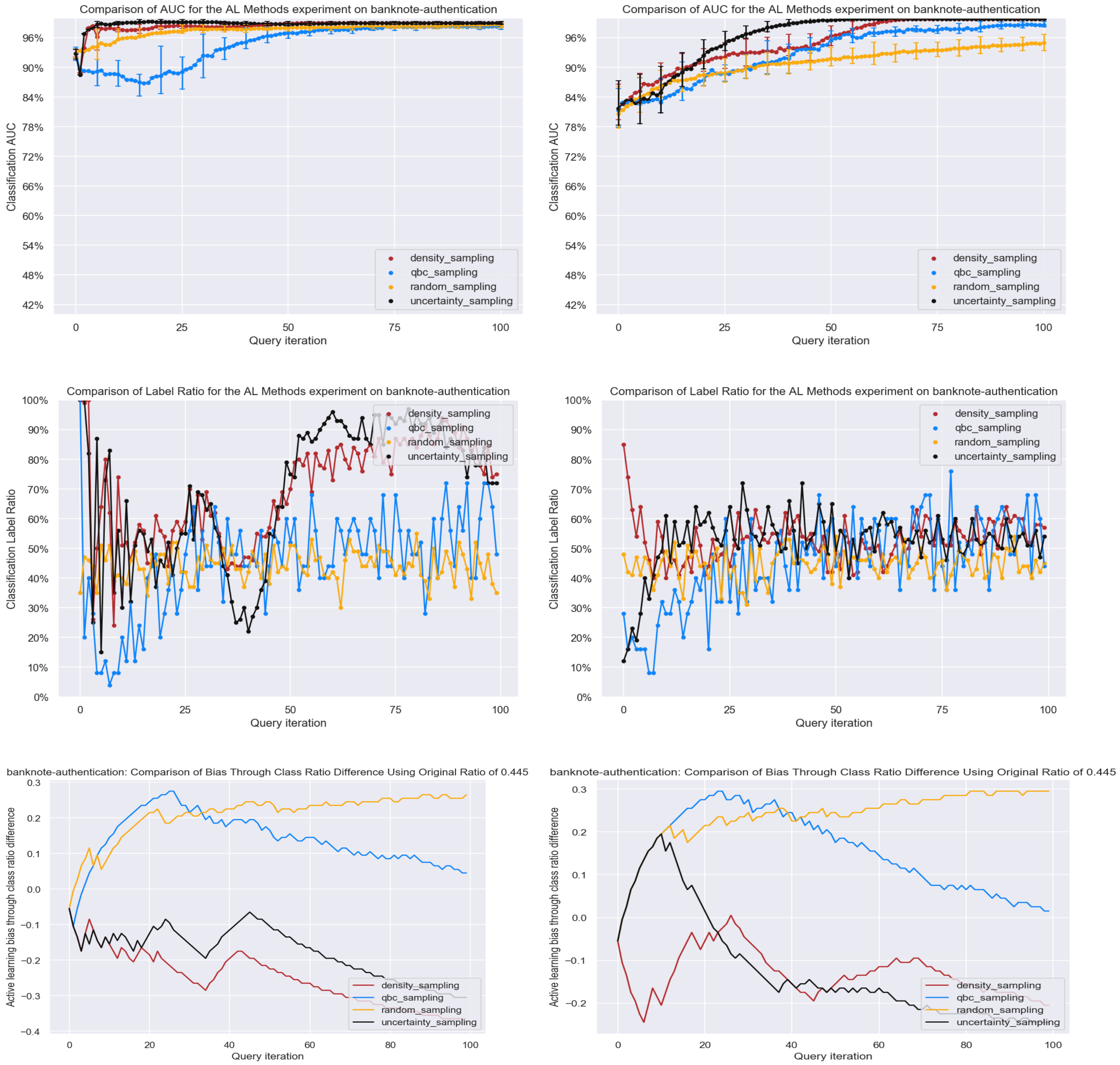


Figure 26: Comparison of performance and querying behaviour using logistic regression and random forest classifiers on the banknote-authentication dataset (class ratio 0.445). Both use a 0.5 initial set ratio. Logistic regression results are in the left column and random forest results in the right.

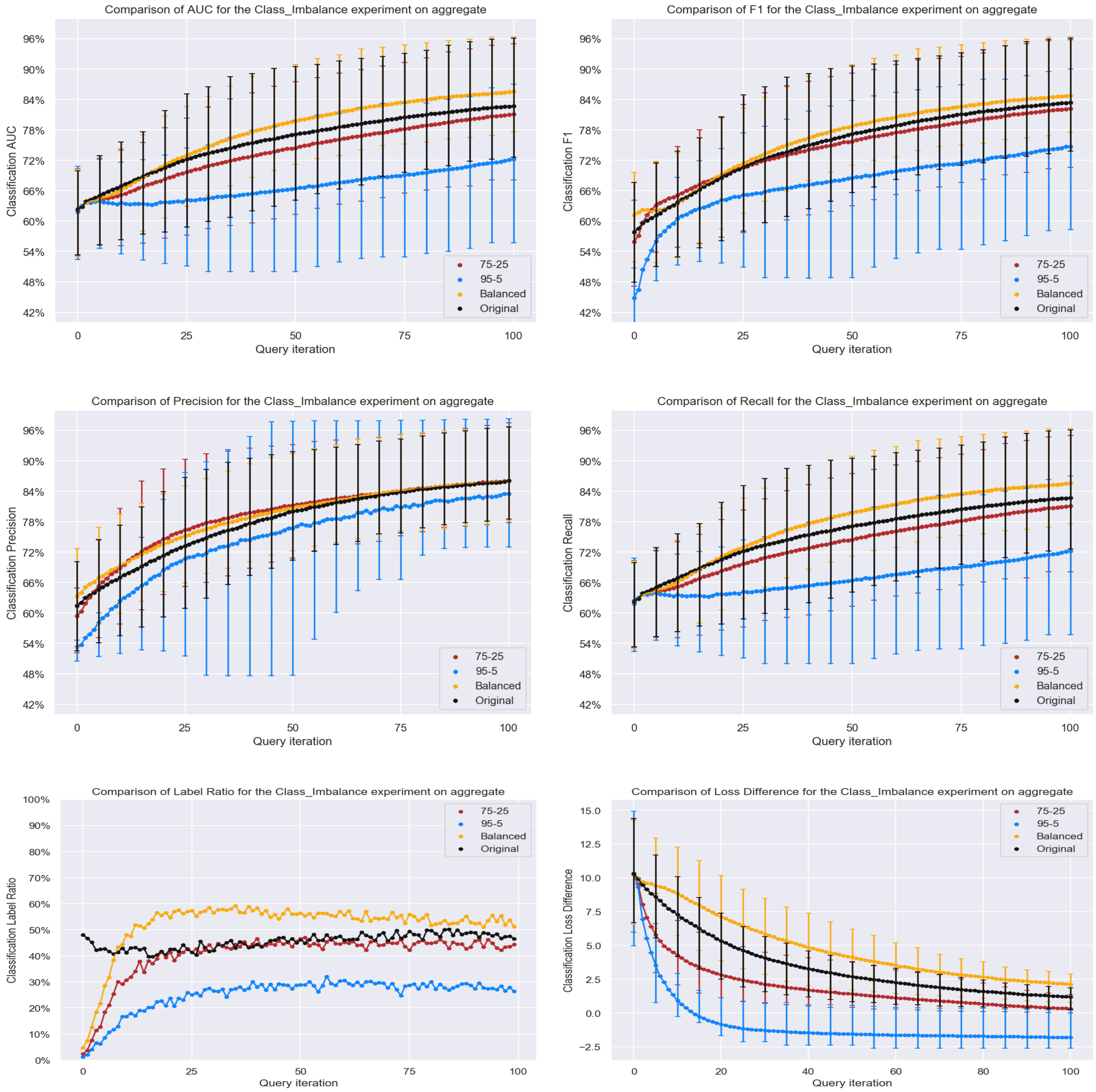


Figure 27: Average performance of CI experiment for Task 1 with uncertainty sampling, 0.5 initial set ratio and using the XGBoost classifier.

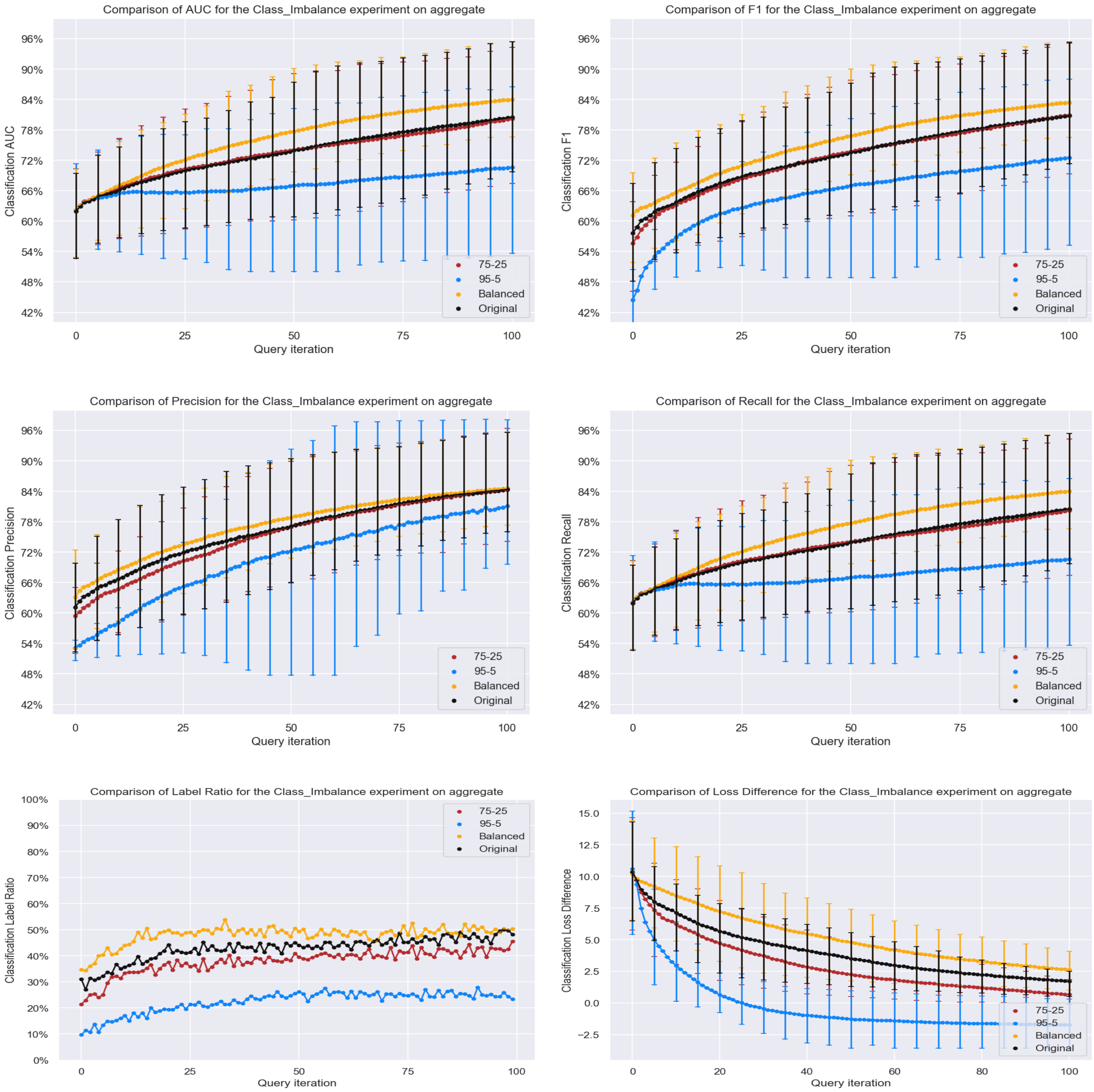


Figure 28: Average performance of CI experiment for Task 1 with density-weighted sampling, 0.5 initial set ratio and using the XGBoost classifier.

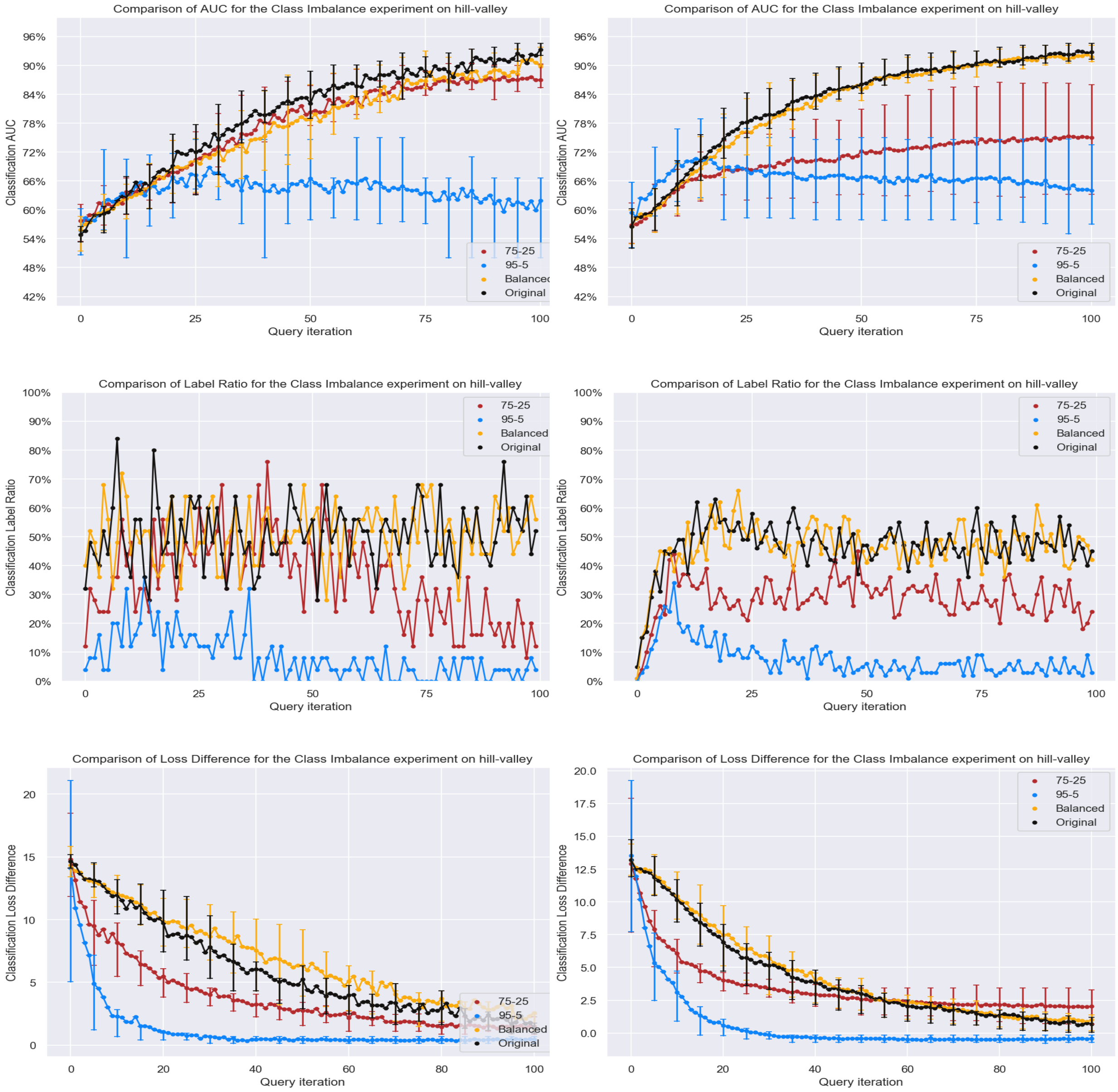
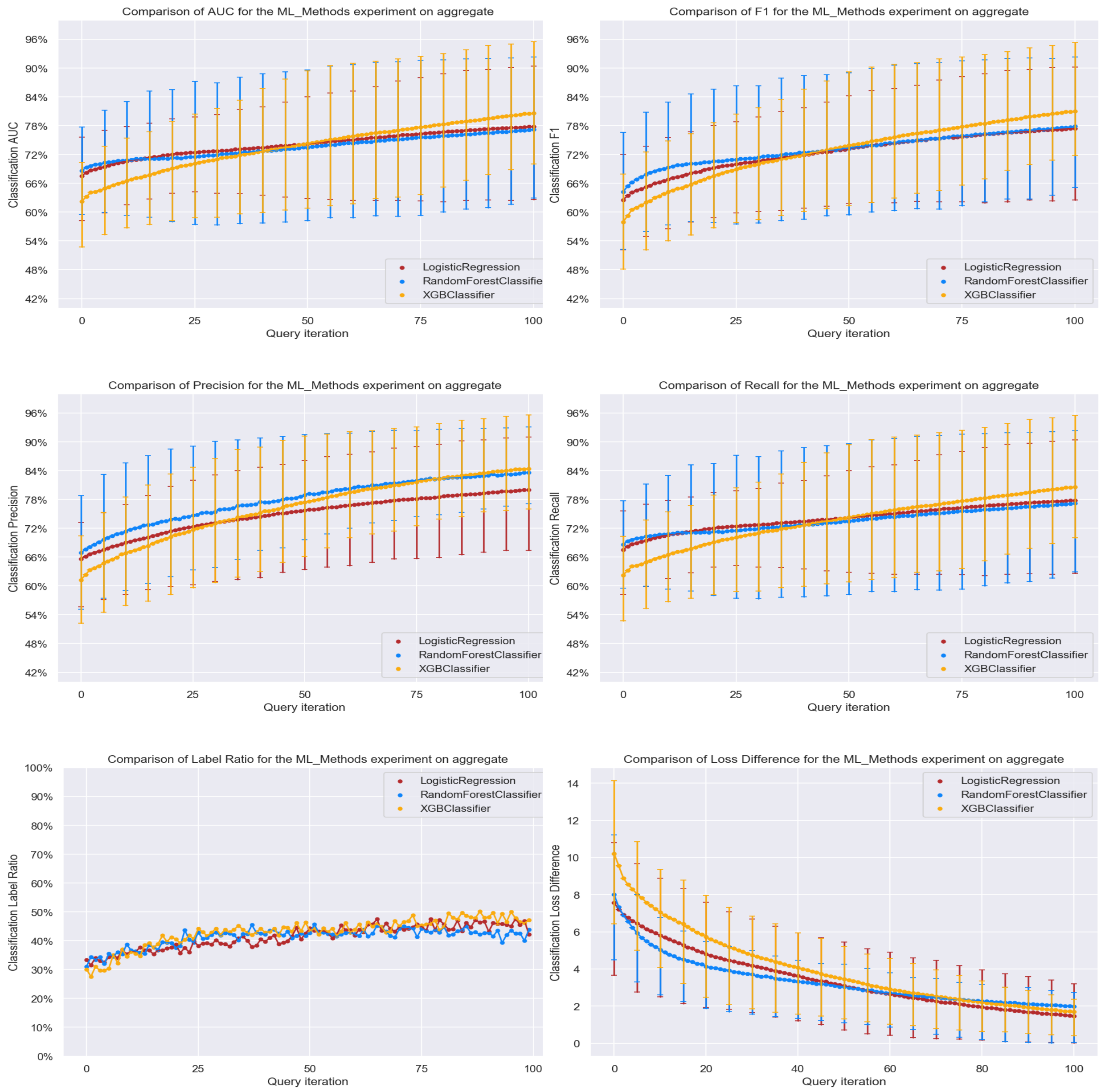


Figure 29: Comparison of performance and querying behaviour for the class imbalance experiment using uncertainty and density-weighted sampling classifiers on the hill-valley dataset (class ratio 0.5). Both use a 0.5 initial set ratio and a logistic regression classifier. Results using uncertainty sampling are in the left column and density-weighted sampling results in the right.


Comparison of AUC for the ML_Methods experiment on aggregate

Comparison of F1 for the ML_Methods experiment on aggregate

Comparison of Precision for the ML_Methods experiment on aggregate

Comparison of Recall for the ML_Methods experiment on aggregate

Comparison of Label Ratio for the ML_Methods experiment on aggregate

Comparison of Loss Difference for the ML_Methods experiment on aggregate

Figure 30: Average performance of machine learning classifier experiment for Task 1 using density-weighted sampling and a balanced initial set ratio.

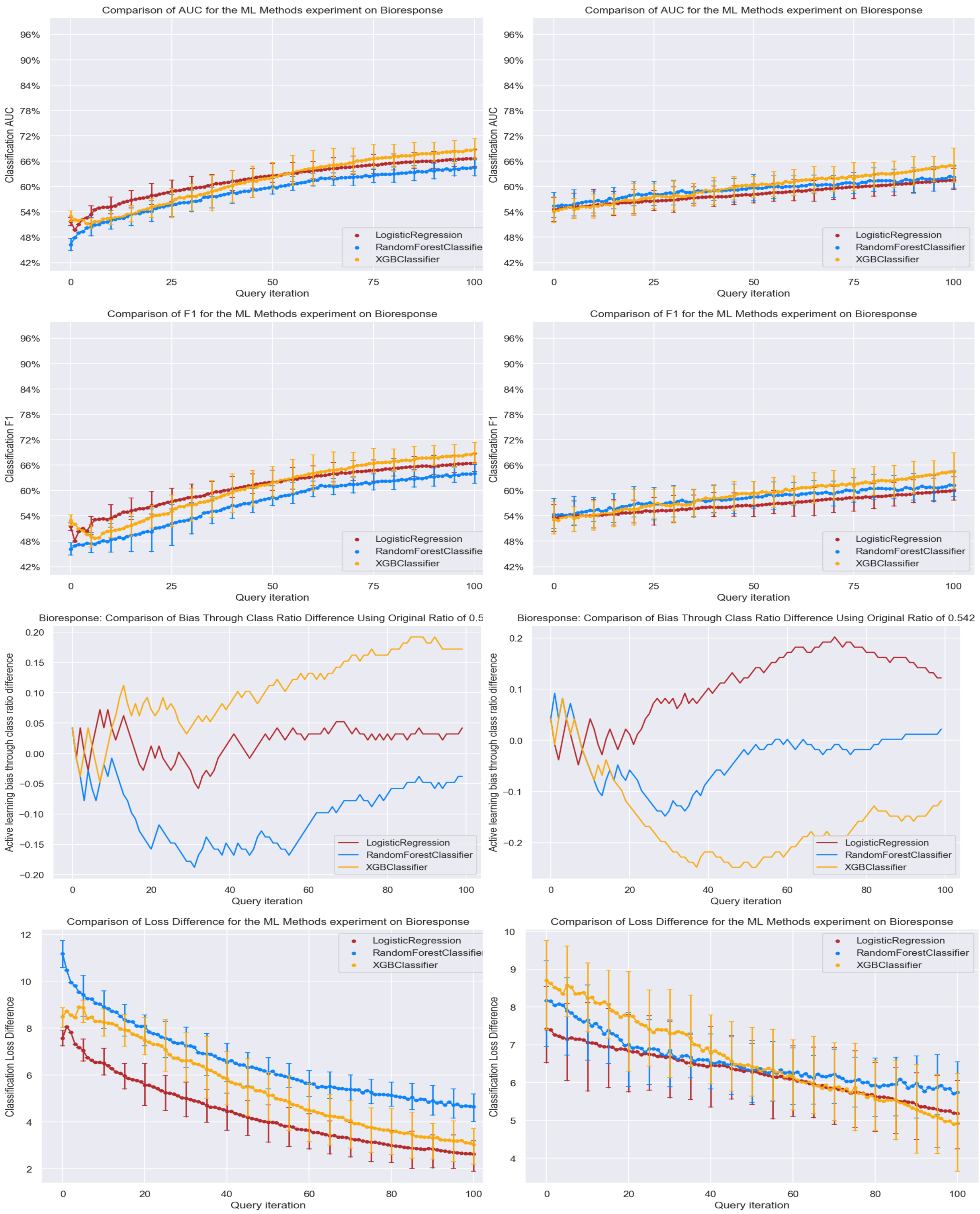


Figure 31: Performance and sampling bias comparison between using uncertainty sampling (left column) and density-weighted sampling (right column) on the Bioresponse dataset (class ratio 0.542). Results generated using a balanced initial set ratio of 0.5.

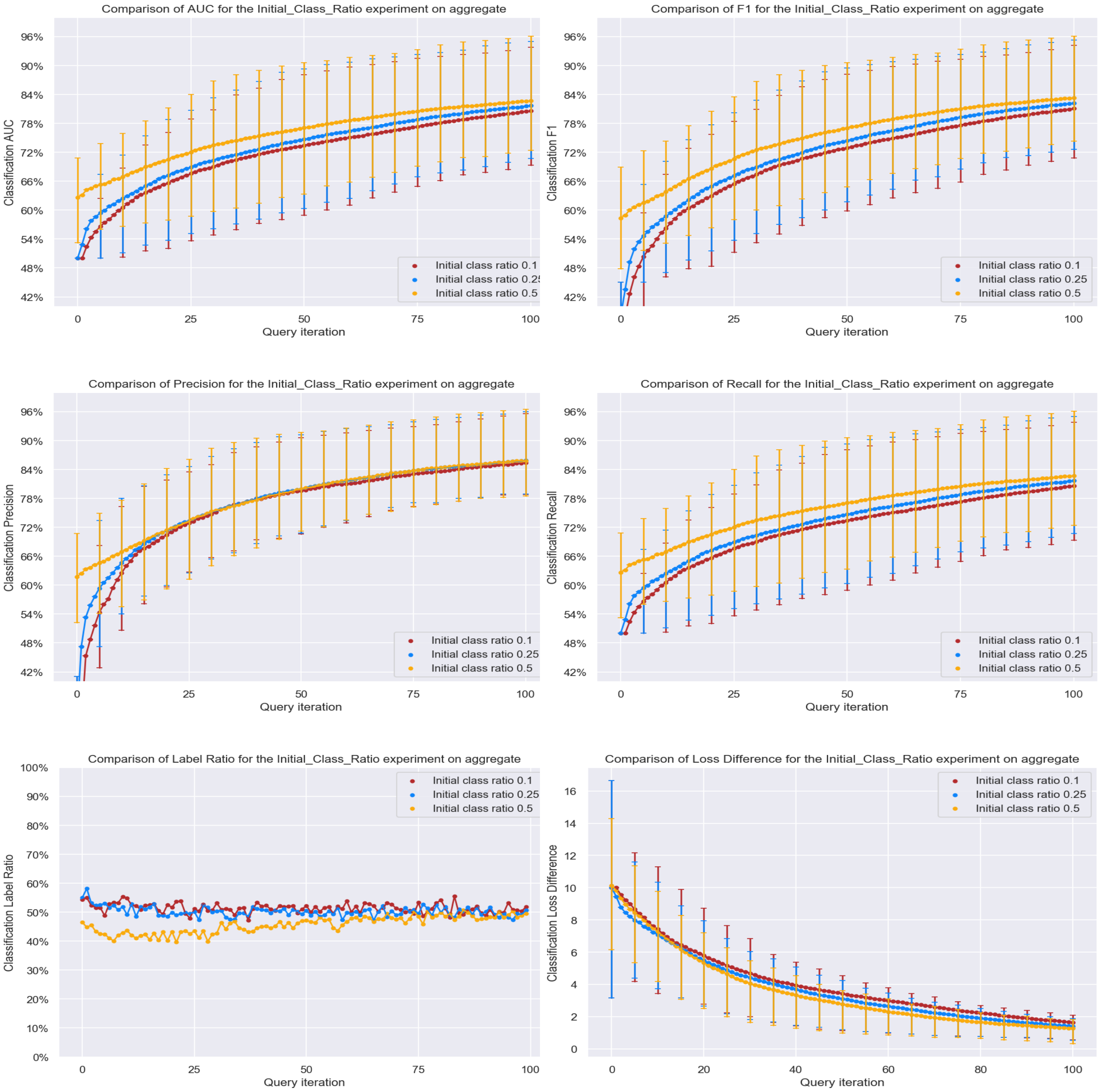


Figure 32: Average performance of initial set ratio experiment for Task 1 with uncertainty sampling and the XGBoost classifier.

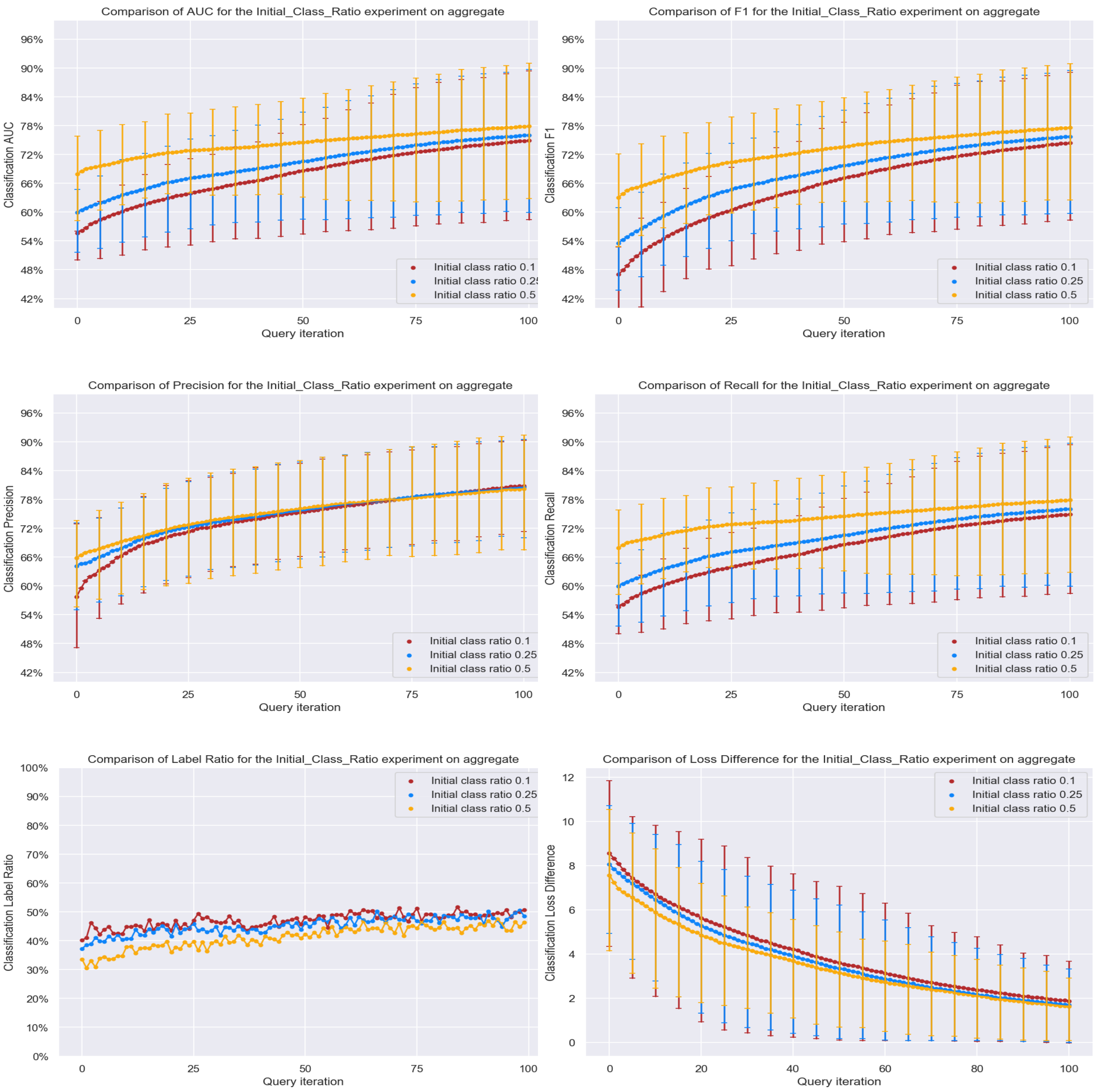


Figure 33: Average performance of initial set ratio experiment for Task 1 with density-weighted sampling and using the logistic regression classifier.

Appendix B: Additional Dataset Results for Task 2

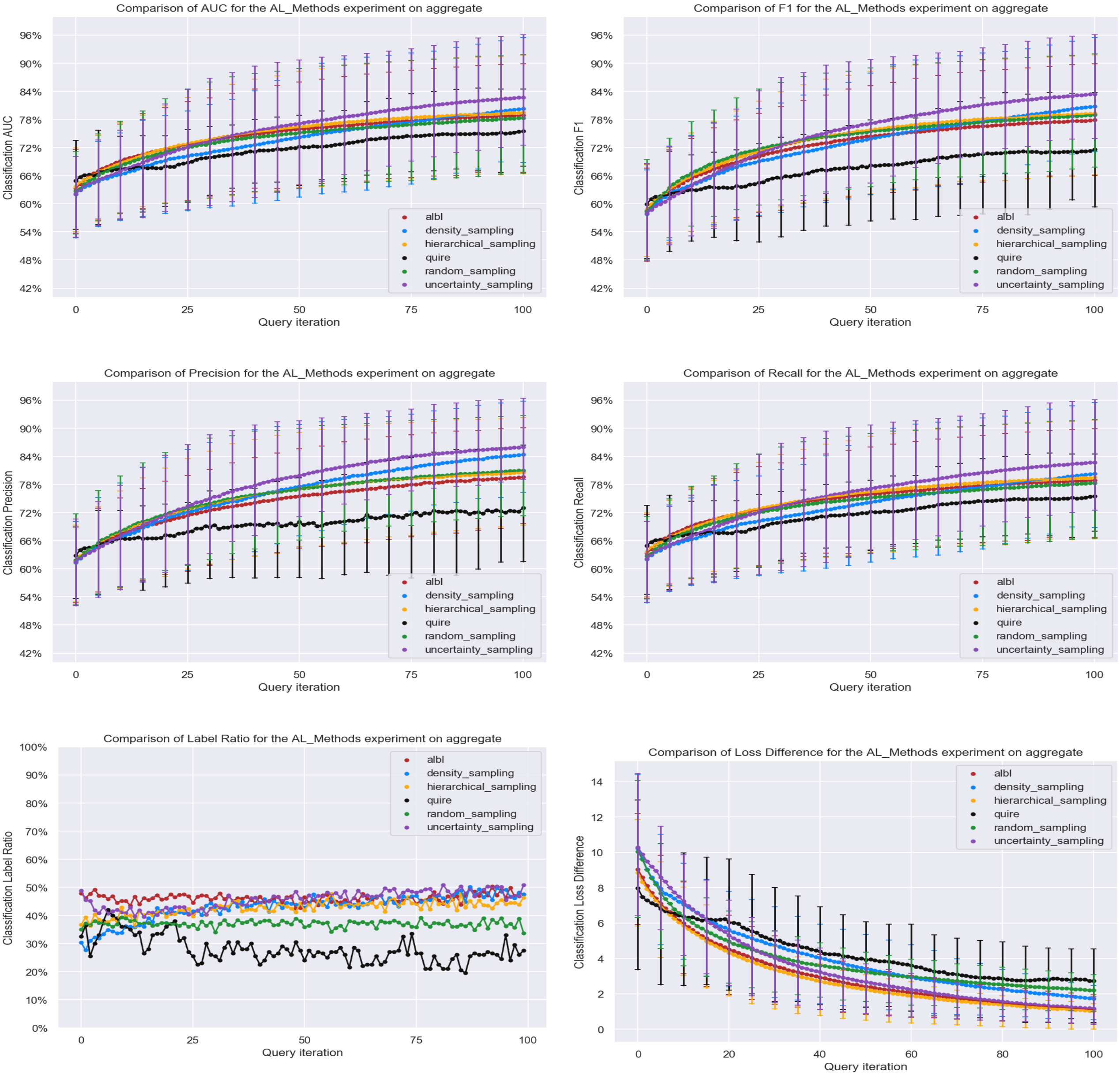


Figure 34: (Macro) AUC, F1, precision, recall, label ratio and loss difference over query iterations over all 15 datasets when comparing AL debiasing methods using a logistic regression classifier to random sampling, density-weighted sampling and uncertainty sampling with an XGBoost classifier.

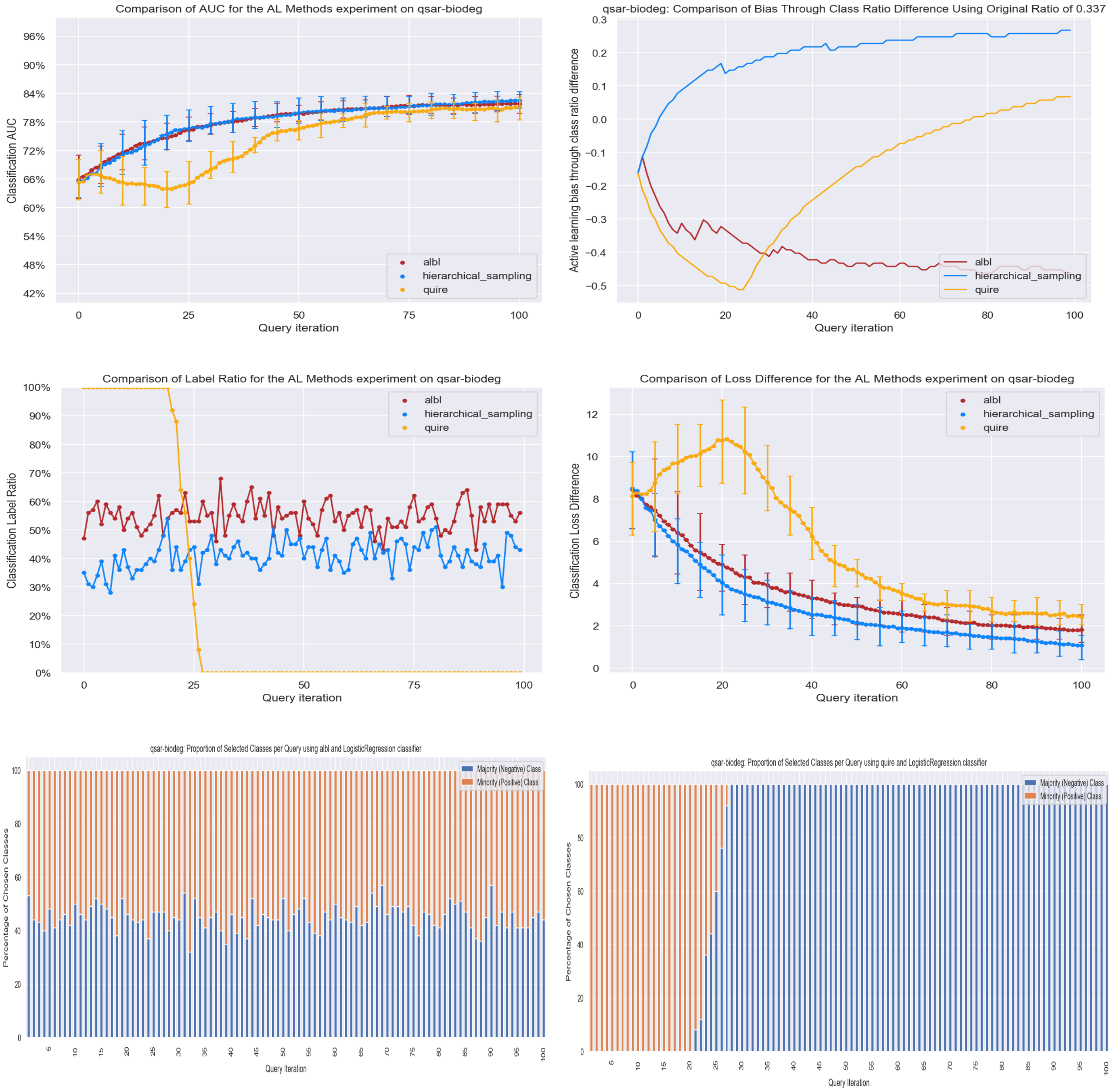
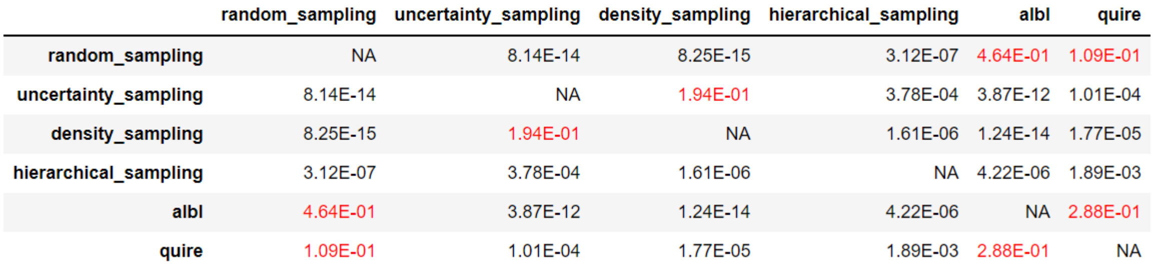


Figure 35: Comparison of (Macro) AUC, class ratio difference, label ratio, loss difference and proportion of selected classes for the AL debiasing methods using a logistic regression classifier on the qsar-biodeg dataset with class ratio 0.337. Bottom row shows proportion of selected classes per query iteration for ALBL on the left and QUIRE on the right.

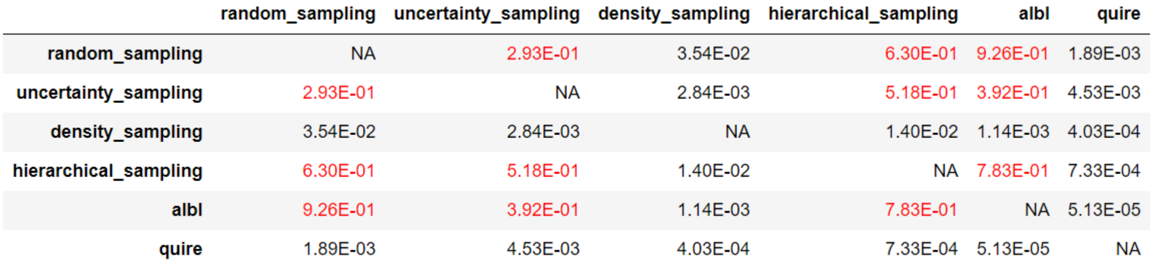
Appendix C: Results Wilcoxon Signed-Rank Significance Test

The following tables show the results of applying Wilcoxon signed-rank test (described in section 5.4) to see if the performance differences between the active learning debiasing methods and random, uncertainty and density-weighted sampling were statistically significant. If the value is in red, it means that the p-value was above 0.05, so the difference between performances was not statistically significant. The Wilcoxon signed-rank test was only applied on the first 8 datasets from table 6, as QUIRE has only been applied to these datasets.



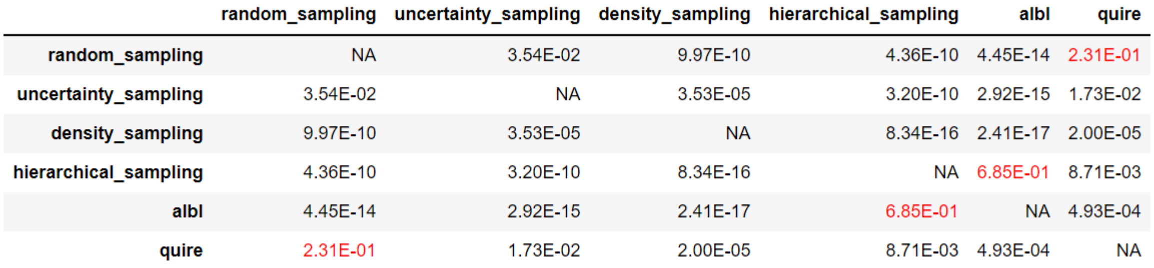
	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	8.14E-14	8.25E-15	3.12E-07	4.64E-01	1.09E-01
uncertainty_sampling	8.14E-14	NA	1.94E-01	3.78E-04	3.87E-12	1.01E-04
density_sampling	8.25E-15	1.94E-01	NA	1.61E-06	1.24E-14	1.77E-05
hierarchical_sampling	3.12E-07	3.78E-04	1.61E-06	NA	4.22E-06	1.89E-03
albl	4.64E-01	3.87E-12	1.24E-14	4.22E-06	NA	2.88E-01
quire	1.09E-01	1.01E-04	1.77E-05	1.89E-03	2.88E-01	NA

Figure 36: Wilcoxon-signed rank significance test results on the monks-problems-3 dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



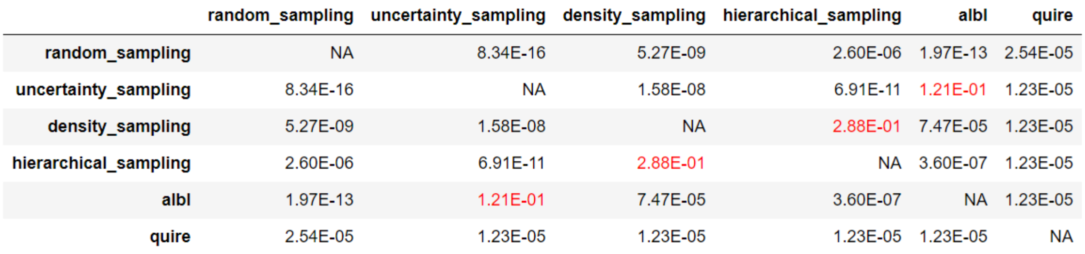
	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	2.93E-01	3.54E-02	6.30E-01	9.26E-01	1.89E-03
uncertainty_sampling	2.93E-01	NA	2.84E-03	5.18E-01	3.92E-01	4.53E-03
density_sampling	3.54E-02	2.84E-03	NA	1.40E-02	1.14E-03	4.03E-04
hierarchical_sampling	6.30E-01	5.18E-01	1.40E-02	NA	7.83E-01	7.33E-04
albl	9.26E-01	3.92E-01	1.14E-03	7.83E-01	NA	5.13E-05
quire	1.89E-03	4.53E-03	4.03E-04	7.33E-04	5.13E-05	NA

Figure 37: Wilcoxon-signed rank significance test results on the qsar-biodeg dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



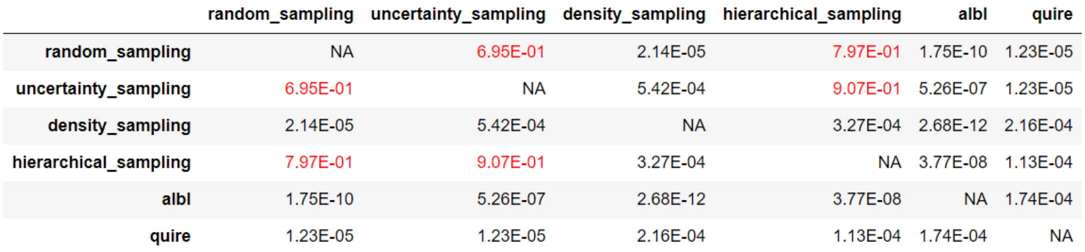
	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	3.54E-02	9.97E-10	4.36E-10	4.45E-14	2.31E-01
uncertainty_sampling	3.54E-02	NA	3.53E-05	3.20E-10	2.92E-15	1.73E-02
density_sampling	9.97E-10	3.53E-05	NA	8.34E-16	2.41E-17	2.00E-05
hierarchical_sampling	4.36E-10	3.20E-10	8.34E-16	NA	6.85E-01	8.71E-03
albl	4.45E-14	2.92E-15	2.41E-17	6.85E-01	NA	4.93E-04
quire	2.31E-01	1.73E-02	2.00E-05	8.71E-03	4.93E-04	NA

Figure 38: Wilcoxon-signed rank significance test results on the hill-valley dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



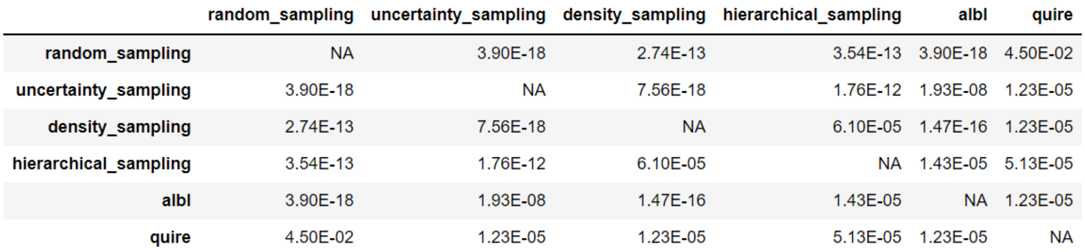
	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	8.34E-16	5.27E-09	2.60E-06	1.97E-13	2.54E-05
uncertainty_sampling	8.34E-16	NA	1.58E-08	6.91E-11	1.21E-01	1.23E-05
density_sampling	5.27E-09	1.58E-08	NA	2.88E-01	7.47E-05	1.23E-05
hierarchical_sampling	2.60E-06	6.91E-11	2.88E-01	NA	3.60E-07	1.23E-05
albl	1.97E-13	1.21E-01	7.47E-05	3.60E-07	NA	1.23E-05
quire	2.54E-05	1.23E-05	1.23E-05	1.23E-05	1.23E-05	NA

Figure 39: Wilcoxon-signed rank significance test results on the banknote-authentication dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



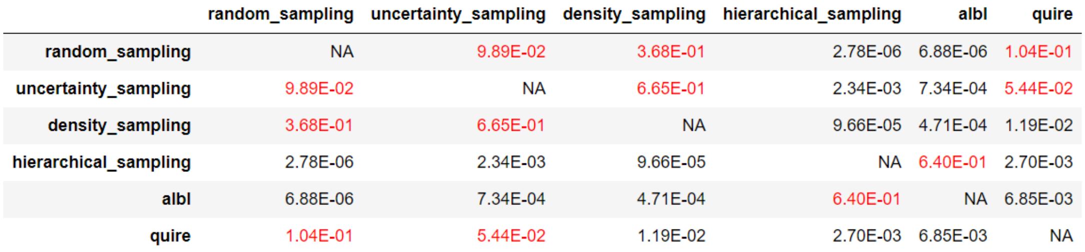
	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	6.95E-01	2.14E-05	7.97E-01	1.75E-10	1.23E-05
uncertainty_sampling	6.95E-01	NA	5.42E-04	9.07E-01	5.26E-07	1.23E-05
density_sampling	2.14E-05	5.42E-04	NA	3.27E-04	2.68E-12	2.16E-04
hierarchical_sampling	7.97E-01	9.07E-01	3.27E-04	NA	3.77E-08	1.13E-04
albl	1.75E-10	5.26E-07	2.68E-12	3.77E-08	NA	1.74E-04
quire	1.23E-05	1.23E-05	2.16E-04	1.13E-04	1.74E-04	NA

Figure 40: Wilcoxon-signed rank significance test results on the steel-plates-fault dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	3.90E-18	2.74E-13	3.54E-13	3.90E-18	4.50E-02
uncertainty_sampling	3.90E-18	NA	7.56E-18	1.76E-12	1.93E-08	1.23E-05
density_sampling	2.74E-13	7.56E-18	NA	6.10E-05	1.47E-16	1.23E-05
hierarchical_sampling	3.54E-13	1.76E-12	6.10E-05	NA	1.43E-05	5.13E-05
albl	3.90E-18	1.93E-08	1.47E-16	1.43E-05	NA	1.23E-05
quire	4.50E-02	1.23E-05	1.23E-05	5.13E-05	1.23E-05	NA

Figure 41: Wilcoxon-signed rank significance test results on the scene dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.



	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	9.89E-02	3.68E-01	2.78E-06	6.88E-06	1.04E-01
uncertainty_sampling	9.89E-02	NA	6.65E-01	2.34E-03	7.34E-04	5.44E-02
density_sampling	3.68E-01	6.65E-01	NA	9.66E-05	4.71E-04	1.19E-02
hierarchical_sampling	2.78E-06	2.34E-03	9.66E-05	NA	6.40E-01	2.70E-03
albl	6.88E-06	7.34E-04	4.71E-04	6.40E-01	NA	6.85E-03
quire	1.04E-01	5.44E-02	1.19E-02	2.70E-03	6.85E-03	NA

Figure 42: Wilcoxon-signed rank significance test results on the ozone-level-8hr dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.

	random_sampling	uncertainty_sampling	density_sampling	hierarchical_sampling	albl	quire
random_sampling	NA	1.48E-13	2.60E-03	1.61E-01	9.47E-02	4.07E-05
uncertainty_sampling	1.48E-13	NA	9.94E-09	2.17E-08	6.75E-10	1.23E-05
density_sampling	2.60E-03	9.94E-09	NA	6.67E-01	1.14E-01	1.77E-05
hierarchical_sampling	1.61E-01	2.17E-08	6.67E-01	NA	9.86E-01	1.40E-04
albl	9.47E-02	6.75E-10	1.14E-01	9.86E-01	NA	3.22E-05
quire	4.07E-05	1.23E-05	1.77E-05	1.40E-04	3.22E-05	NA

Figure 43: Wilcoxon-signed rank significance test results on the jasmine dataset for the performance differences of the three debiasing methods when compared to random, uncertainty and density-weighted sampling.

Appendix D: Results QUIRE and ALBL on Imbalanced Data

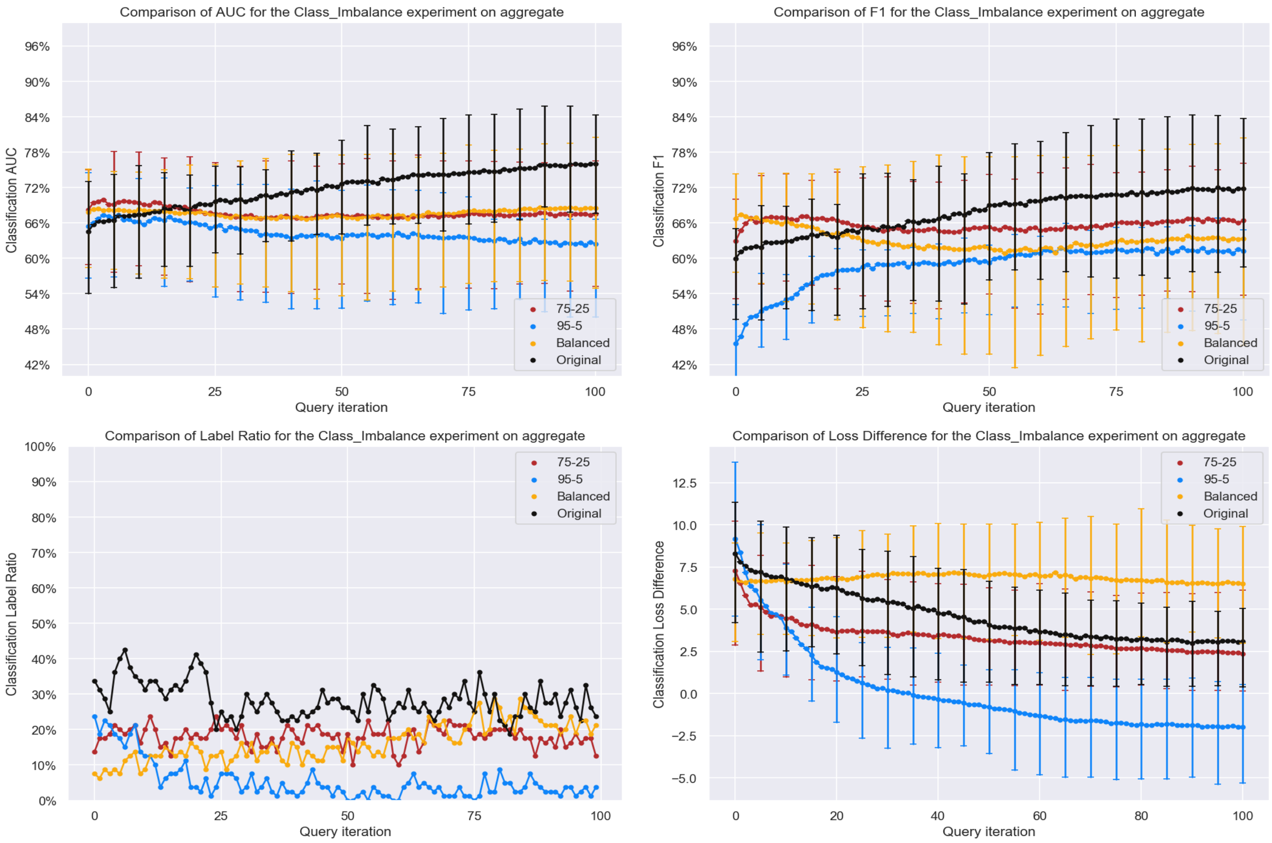
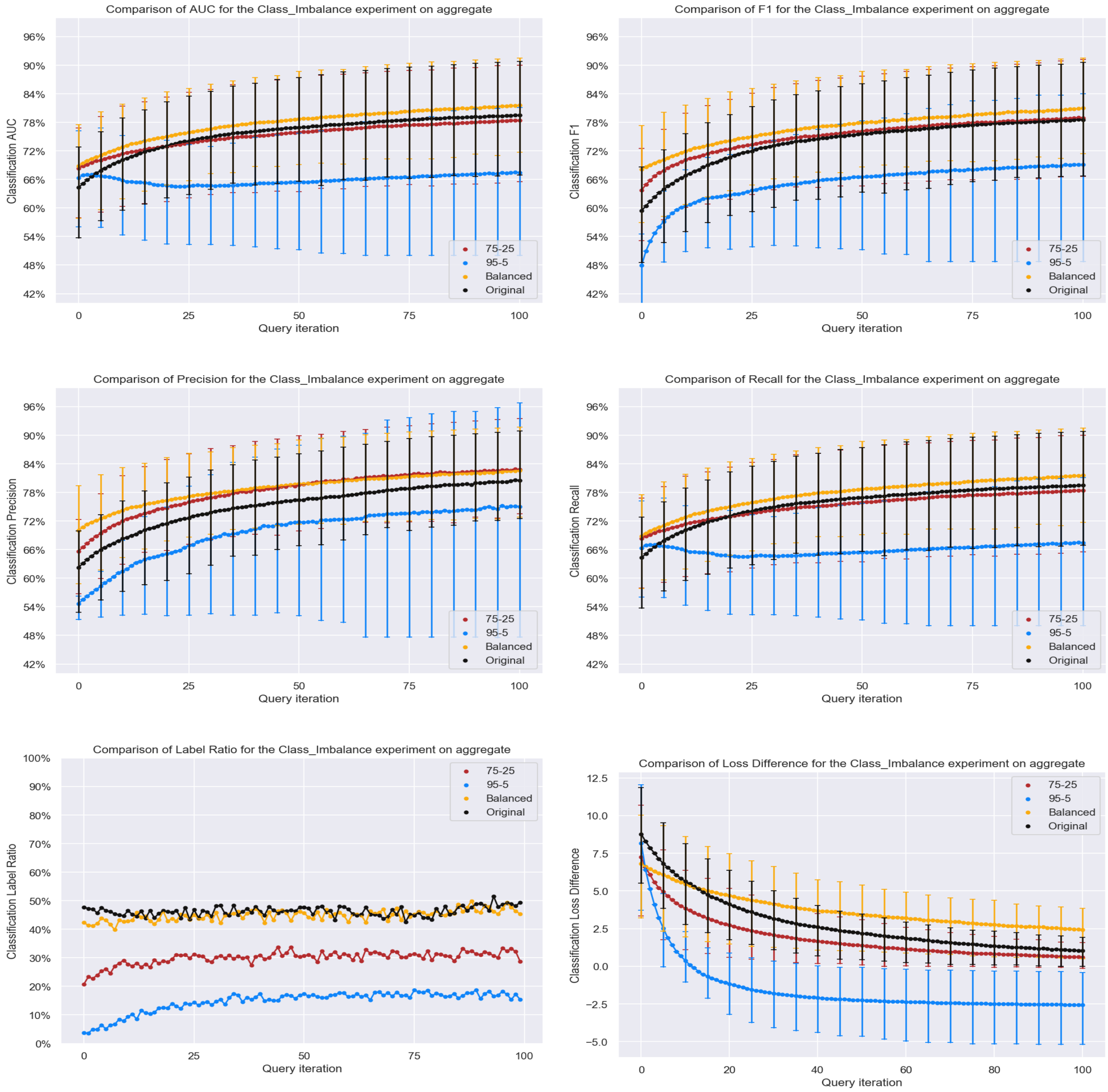


Figure 44: (Macro) AUC, F1, label ratio and loss difference over query iterations over all 15 datasets when applying QUIRE and logistic regression to imbalanced data.


Comparison of AUC for the Class_Imbalance experiment on aggregate

Comparison of F1 for the Class_Imbalance experiment on aggregate

Comparison of Precision for the Class_Imbalance experiment on aggregate

Comparison of Recall for the Class_Imbalance experiment on aggregate

Comparison of Label Ratio for the Class_Imbalance experiment on aggregate

Comparison of Loss Difference for the Class_Imbalance experiment on aggregate

Figure 45: (Macro) AUC and F1, as well as bias visualisation through label ratio and loss difference over 100 query iterations over all 15 datasets when applying ALBL and logistic regression to imbalanced data.

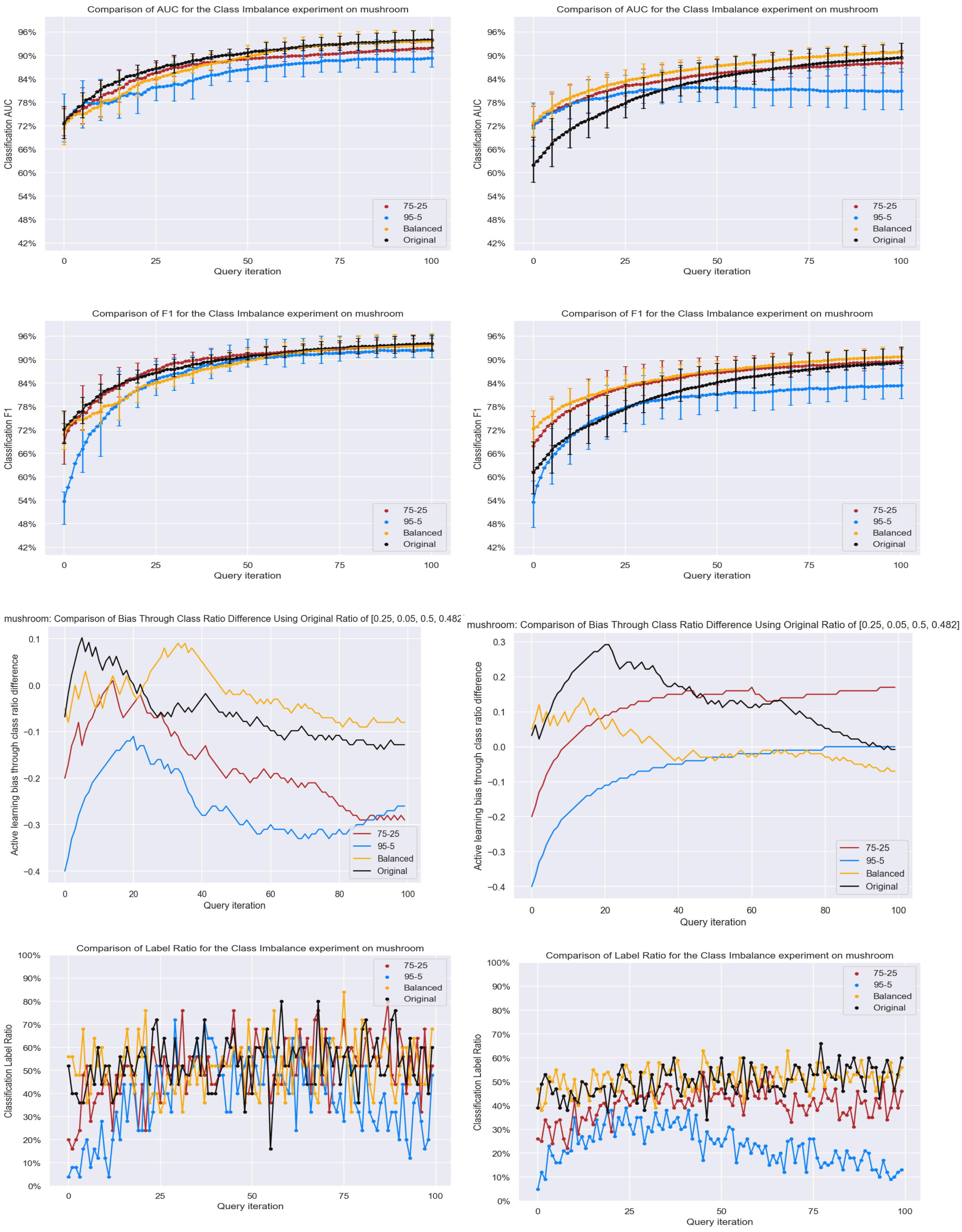


Figure 46: Comparison of (Macro) AUC and F1 and bias through label ratio and class ratio difference between uncertainty sampling (left) and hierarchical sampling (right) on the mushroom dataset with varying levels of class imbalance. The mushroom dataset has an original class ratio of 0.482. Both query strategies use a logistic regression classifier and balanced initial set ratio.