Using Discretization and Resampling for Privacy Preserving Data Analysis: An experimental evaluation

Sven Pors
4014618
s.j.pors@students.uu.nl

Master of Business Informatics
Department of Information and Computing Sciences
Faculty of Science
Utrecht University

First supervisor:
dr. ing. Georg Krempl
Algorithmic Data Analysis
Faculty of Science
Utrecht University
G.M.Krempl@uu.nl

Second supervisor:
prof. dr. Arno Siebes
Algorithmic Data Analysis
Faculty of Science
Utrecht University
A.P.J.M.Siebes@uu.nl

Daily supervisor:
Joop Snijder
Head of Artificial Intelligence Research
Info Support
Joop.Snijder@infosupport.com

Master Thesis

November 22, 2018

Abstract

Data analysis allows for the extraction of useful patterns or information from data. However, most data that is stored and processed contains personal information of individuals. The analysis of this data is therefore increasingly restricted by laws and regulations, and pressured by the public opinion. This calls for an approach that allows for performing data analysis, while protecting the privacy of individuals that are in the data. Such an approach would make storing, processing, exchanging and publishing data more feasible, and less restricted by regulations.

This thesis report contributes to the field of Privacy Preserving Data Mining, by addressing the research question: *How can data be accurately summarized by as few instances as possible to support data analysis, while preserving the privacy of individuals?* It does so by introducing a novel approach towards data anonymization, that can be used to provide privacy guarantees, while mostly preserving the utility of the continuous data. The existing concept of Density Estimation Trees (DETs) is used for the multidimensional discretization of continuous attributes. This research proposes to achieve the privacy model k-anonymity by using k as a minimum leaf constraint, and a stopping rule during the creation of DETs. This discretization of continuous instances therefore yields a number of equivalence classes, where each equivalence class is defined by one of the DET's leaf nodes, and contains at least k instances.

The proposed approach is validated through an experimental evaluation, by evaluating it using fifteen real-world, synthetic or mixed data sets, containing continuous attributes. The preservation of data utility is measured by comparing a classifier's performance achieved with the continuous data, and the performance with the anonymized data. The privacy level is expressed by k within the context of k-anonymity, which serves as an input parameter for the DET as well.

The results of the evaluation show that with only three out of the fifteen data sets, there is a significant difference in classification accuracy when comparing the continuous and anonymized attributes. In addition, in ten out of fifteen cases, a k-value of at least 10 achieves the highest classification accuracy.

It can be concluded that in most cases, the anonymization approach that is introduced succeeds to create an accurate representation of the continuous attributes that preserves data utility. In addition, it does so while providing privacy guarantees through k-anonymity for relatively high k-values.

Acknowledgements

Starting a master thesis project can be daunting, at least it was for me: an eight month period of conducting a research project and writing the thesis itself. During this time, an initial conception became a problem statement, which became a project proposal. The following months, the project continuously progressed and evolved. And now, after these eight months, all of that work resulted in this final version of my master thesis. I hereby would like to thank those who were closely involved throughout this whole process, without whom this project would not have been possible.

First of all, I would like to thank Georg Krempl, for his advice, guidance and enthusiasm throughout the project. Especially in the more difficult times, his feedback allowed me to progress and move on with the project. Whether I had simple questions or difficult questions, he was always ready to help and decompose the problem.

I would also like to thank Arno Siebes, for sharing his extensive knowledge and views on the topic. Whenever I proposed some ideas, his feedback reassured me in the approach that was taken.

Furthermore, I want to thank Joop Snijder, for sharing his passionate feedback about this research project, from the conception all the way to the final product. Without his trust, support, and confidence in me and the relevance and implications of this research project, this thesis would not have been where it is now.

Lastly, I would like to thank my parents. Their support and care have always been massive, but especially so during these busy months. I can only be grateful to them for providing me everything that I have ever needed and more.

Contents

List of Tables

List of Figures

Part I

Problem investigation

Chapter 1

Introduction

This thesis describes a research project on Privacy Preserving Data Analysis. While interest in advanced analysis of large data sets is only increasing, so is the awareness about the privacy issues that are involved. Section 1.1 contains a detailed description of the problem to be addressed in this thesis. Next, Section 1.2 describes the relevance of the research described in this thesis. Lastly, Section 1.3 provides an outline of the remainder of the report.

1.1 Problem statement

1.1.1 Background

The interest in data analysis has been present for several decades. Although concepts like Data Mining and Machine Learning have theoretical differences, their goal is somewhat similar: to extract useful patterns or information from data (Witten, Frank, Hall, & Pal, 2016). Data analysis is carried out in many fields, e.g. health care, transportation, banking, social media and e-commerce. Learning algorithms yield valuable insights that could not have been gained otherwise. However, data that is collected and analyzed usually involves people and their personal information. This means that personal and sensitive data of individuals is being collected, transferred, analyzed, and sometimes even published.

Although the implementation of privacy regulations and laws is not a recent development, new laws are being implemented to fit the challenges and characteristics of the modern information era. An example of a relatively new law is the General Data Protection Regulation (GDPR), which is active since April 2016 for all members of the European Union. This regulation creates more conditions for data processors, more rights for the individuals whose data is being processed, and larger fines for violations (European Union, 2016). It holds for all data processors residing in EU states, and for all processors of data involving citizens of the European Union. The law thereby accounts for the extra-territorial nature of the internet and removes all ambiguity and importance of server locations and the placement of data centers (EUGDPR.org, 2016). Since the GDPR holds for a large population and is more strict in general, organizations that process personal information do care about complying to it. It was possible to take two years of preparation to comply to the GDPR. Organizations were therefore expected and required to do so since May 2018.

Strict regulations concerning privacy is a positive development, but organizations do not always have the knowledge on what is and is not allowed. This could potentially lead to a reserved attitude, which could impede advanced data analysis. For example, organizations could be restrictive towards sharing their data with external parties for data analysis, to avoid the risk of violating privacy regulations. Solutions to share data while respecting confidential or personal information of individuals are generally not one-size-fits-all.

Apart from new regulations, the awareness of privacy and issues with collecting and sharing data is increasing. At the time of writing, a large privacy scandal concerning Facebook came to light. Specifically, personal information of up to 87 million Facebook users was improperly shared with a data analysis company, Cambridge Analytica (Ingram, 2018). Apart from the initial hit Facebook took, Cambridge Analytica declared bankruptcy within a few months after the incident hit the news (Reuters, 2018). Due to the extensive media attention, most organizations would want to take measures to avoid such scandals, and individuals become aware of the risks of sharing their information with these organizations.

The notion of privacy in data analysis is not restricted to organizations or governments. Processing or publishing data is an integral part in a scientific environment as well. For example, statistical analyses to test hypotheses are performed on responses of a questionnaire or survey. Publishing medical data provides insights in medicine as well, but this type of data naturally contains very sensitive information on individuals. When publishing this data, it is imperative, and in most cases required by law, that individuals in the data remain anonymous and could not be identified by any means.

Another use of data sets is in education. When educating students in statistics, data analysis, Data Mining, or Machine Learning, using real world data sets is worth more than using dummy or mockup data (Neumann, Hood, & Neumann, 2013). However, the public availability of real world data sets leaves something to be desired (Wixom et al., 2011), and is also restricted by privacy regulations.

An answer to the issues of organizations, science, and education is to anonymize data in such a way that the privacy of individuals is respected, while data analysis is still possible. This is not a trivial task however, as several cases show how individuals could be identified from anonymized data with minimal effort (De Montjoye, Hidalgo, Verleysen, & Blondel, 2013; Malin & Sweeney, 2004; Narayanan & Shmatikov, 2008; Samarati, 2001; Sweeney, 2002).

1.1.2 Privacy Preserving Data Analysis

To avoid de-anonymization and to guarantee the privacy of individuals while enabling data analysis, research is conducted under terms like *Privacy Preserving Data Mining* and *Privacy Preserving Data Analysis*. The latter will be used throughout this thesis. The core concept of this field is to use data to our benefit through analysis, without compromising the privacy of individuals in the data. Various techniques, models, and metrics exist (Mendes & Vilela, 2017). Although these techniques are very effective when used correctly, their use in practice is difficult to estimate. There are many different techniques and models that require tight integration (Prasser & Kohlmayer, 2015). The techniques to choose depend on the application and the nature and structure of data. Moreover, privacy models require parameters set by the user. Data anonymization requires at least some amount of preparation, as well as knowledge on the topic.

A common distinction when it comes to privacy in data sets is that between *explicit* identifiers and *quasi* identifiers. The former refers to data that can directly identify a specific individual, like names and social security numbers. The latter refers to data that cannot *directly* identify a specific individual, but combined with auxiliary information could reveal one's identity. For example, data containing street name and age could be enough to identify one specific individual, when an adversary has auxiliary knowledge on someone's age and street. Golle (2006) shows that 63% of US citizens can be uniquely identified just by the combination of their gender, zip code, and birthdate.

Intuitively, explicit identifiers pose a larger threat for privacy. However, these are also the least complex to deal with. When sharing or publishing data for analysis, explicit identifiers are typically removed, either for legal reasons, or due to the fact that these are usually not the most interesting for data analysis tasks. The challenge lies with data that is interesting for analysis or learning tasks, like classification or regression. This data needs to be preserved for effective analysis. However, its disclosure has potential privacy risks.

This research project addresses this problem, and tries to find a comprehensive method to allow data mining or other analysis techniques to be performed, while individuals cannot be identified from the data, and thus stay anonymous. More specifically, the aim is to transform a set of continuous numerical features and instances into an anonymized set. This is achieved through discretization and sampling of the original set to create an accurate summary of the data. This summary should be representative of the original data's distribution to allow for effective data analysis, and at the same time preserve the privacy of individual records in the data. Moreover, the resulting level of data utility and privacy should be measurable to ensure that both satisfy the defined requirements.

1.2 Relevance

1.2.1 Practical relevance

Over the past two decades, many attempts were made to formulate models providing privacy guarantees. In addition, several algorithms were proposed that satisfy a privacy model while data utility is maximized. However, each model has its drawbacks, and the lack of a silver bullet poses difficulties for practitioners. For example, the GDPR encourages employing methods for anonymization and pseudonymization. Doing so, data processors are allowed to handle data more freely, or might even fall outside of the regulation's scope. However, no approaches were suggested for anonymization, or what criteria can be used to determine anonymity of data (Wes, 2017). This is understandable, since it is unlikely that a method can be developed that supports all possible use cases, is free of ambiguity, provides an absolute privacy guarantee, and preserves maximal data utility. However, organizations and practitioners would benefit from a comprehensive approach that can be used to anonymize a set of features and instances, while preserving the statistical properties of the data. The proposed approach tries to achieve anonymization through discretization and sampling. Since these techniques are substantial parts of preprocessing in most data analysis projects (Kotsiantis, Kanellopoulos, & Pintelas, 2006), using an anonymization approach based on these techniques would fit in well, and does not require a radical change in preparing data for analysis.

1.2.2 Scientific relevance

Although many methods, models, and techniques have been studied for achieving privacy, experimental or empirical research on this topic is limited. This project contributes to the body of knowledge on Privacy Preserving Data Analysis by means of an experimental evaluation of applying such a privacy model. By applying an anonymization approach on multiple data sets, using a number of different sample sizes, the effect of anonymization on privacy and utility could potentially be quantified and compared. This makes it possible to evaluate such an algorithm for anonymizing a set of continuous features and instances.

1.3 Report outline

This thesis is structured as follows. Chapter 2 presents the research design followed in this project. Chapter 3 contains a theoretical background on the main concepts, models and techniques on Privacy Preserving Data Analysis, discretization techniques and resampling. Chapter 4 describes the proposed approach that is evaluated, and its rationale, while Chapter 5 describes the experimental setup. The results of the experimental evaluation are presented in Chapter 6. Finally, the conclusion is presented in Chapter in 7.

Chapter 2

Research design

This chapter describes the research design that is used in this thesis, including the research questions, research method and literature research protocol.

2.1 Research questions

This research project aims to find techniques that allow for data analysis on a data set, thereby incorporating mechanisms that protect the privacy of individuals. The starting point is a set of continuous numerical features and instances. This set is summarized through discretization and sampling, and should be representative of the distribution in the original set, while preserving the privacy of individual records.

Techniques are studied to create an accurate summary that has the same statistical properties and characteristics as the original data, to ensure a desired level of data utiliy. At the same time, the method incorporates privacy preserving mechanisms to ensure that the privacy of individuals is not harmed. This goal can be summarized by formulating the following research question:

- **RQ**: *How can data be accurately summarized by as few instances as possible to support data analysis, while preserving the privacy of individuals?*

To provide a structured approach for answering the research question, the following subquestions were formulated:

- **SQ1**: *What privacy preserving mechanisms exist, and what techniques can be used to provide guarantees about privacy?*
- **SQ2**: *How can data be partitioned and sampled, while the distribution is representative of the distribution of the original data?*
- **SQ3**: *How are privacy and utility measured, and what levels of privacy and utility could be considered sufficient?*

- **SQ4**: *What partitioning approach should be used that could be related to the privacy preserving mechanisms?*
- **SQ5**: *How does the proposed approach perform, i.e.: are the achieved levels of privacy and utility in accordance with the required levels?*
- **SQ6**: *How are privacy and utility related, and is it possible to reach a balance between them?*
- **SQ7**: *What is the minimal bucket size, i.e.: how many instances are needed to create an accurate summary?*

The purpose of this thesis is to provide an answer to these subquestions, and ultimately to the main research question. The remainder of this chapter describes the systematic approach to answer these questions.

2.2 Research approach

2.2.1 Design science

The goal of this project is to provide an answer to the research question as defined in the previous section. A suitable research method should be used in order to do so. In information science, projects are usually concerned with solving a particular problem or improving current solutions. In this context, (Wieringa, 2014) defines *design science* to be the *"design and investigation of artifacts in context"* (p. 3). In this sense, artifacts are designed to interact with a problem context, with the intention to improve something in the context.

Within design science, a distinction can be made between *design problems* and *knowledge questions*. The former is concerned with the (re)design of an artifact to improve a context or to find a solution for a problem that meets some goals. On the other hand, knowledge questions ask for knowledge about the world, with the purpose of finding the truth. In contrast to design problems, knowledge questions are assumed to have only one answer, even if there is uncertainty about the answer, or if there only exists a partial answer. Design problems and knowledge questions thus require different questions and approaches, but they are related as well. Knowledge questions can be used to provide an understanding about the problem context, or to evaluate the designed artifact.

This project's main research question poses a typical design problem. In terms of design science, the problem context consists of a set of continuous features and instances that needs to be transformed so that it is representative of the original distribution, while privacy is preserved, with a minimal bucket size. The artifact to be designed is the approach or algorithm to execute this transformation.

In the context of information or computer science, it is common to use experiments. According to Wohlin et al. (2012), experiments are used for the investigation of various aspects, including:

- Confirm theories, i.e. to test existing theories.
- Explore relationships, i.e. to test that a certain relationship holds.
- Evaluate the accuracy of models, i.e. to test that the accuracy of certain models is as expected.
- Validate measures, i.e. to ensure that a measure actually measures what it is supposed to.

In this sense, the design and validation of the artifact (algorithm) could be described as an *experimental evaluation*, in which the artifact is applied, followed by an evaluation of the effects of applying the artifact. Tichy, Lukowicz, Prechelt, and Heinz (1995) stressed the importance of using experiments in computer science. In addition, Hooker (1994) identified two ways of studying the performance of algorithms: (1) using deductive mathematics and (2) using computational experiments. He argues that the second approach is more adequate when dealing with practical problems. In this sense, experiments can be used to actually test and validate an algorithm in a context. Wieringa's concept of design science, in which an artifact is designed and investigated in a context, allows for the execution of an experimental evaluation.

2.2.2 Design cycle

To address design problems, Wieringa (2014) introduced the *design cycle*, which can be used for solving most real-world problems or to improve their solutions. The design cycle consists of three main tasks. *Problem investigation* is concerned with providing a thorough understanding of the problem at hand. Next, during *treatment design*, the requirements and objectives of the artifact to be developed are specified, and a treatment is designed. In this sense, *treatment* refers to the interaction between the designed artifact and the problem context, in order to treat the problem. The third task in the design cycle is *treatment validation*. The purpose of this task is to validate whether the treatment has the desired effect, i.e. whether it meets the specified goals and satisfies the requirements. The validation process should ensure that the proposed treatment does solve the problem context, or that the treatment is an improvement over current solutions. These tasks together are referred to as a cycle since it is common that multiple iterations of each task are performed during a design science project. Table 2.1 provides an overview of the tasks and typical concerns during each task.

The design cycle is part of a larger cycle, called the *engineering cycle*. This larger cycle consists of two extra tasks, concerned with the implementation of the treatment. In this sense, implementation means the application of the treatment to the original problem context. Figure 2.1 shows the engineering cycle, in which the first three tasks form the design cycle.

Table 2.1: Design cycle tasks

Task	**Concerns**
Problem investigation	Stakeholders
	Goals
	Conceptual problem framework
Treatment design	Treatment requirements
	Contributions to goals
	Available treatments
	New treatment design
Treatment validation	Desired effect of treatment
	Compliance to requirements
	Trade-offs for different artifacts

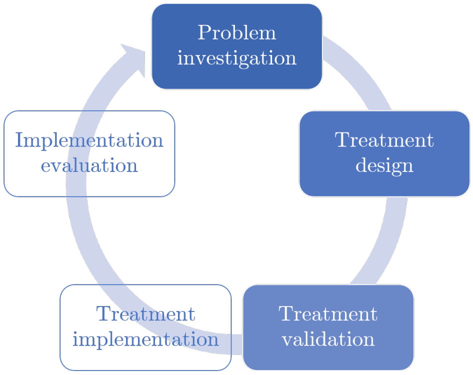


Figure 2.1: Engineering cycle

2.3 Research method

Since the research question of this project is a design problem, the main research activities during this project are summarized by the design cycle tasks. This project can therefore be structured in accordance to the three tasks of the design cycle: (1) problem investigation, (2) treatment design, and (3) treatment validation.

2.3.1 Problem investigation

When designing an artifact for a problem context, it is important to first get a thorough understanding of this problem context. Figure 2.2 contains a simplified overview of the problem context. It starts with a set of continuous features and instances. A summary of the data is generated through discretization and resampling, representing the original distribution. Discretization and resampling form the artifact to be designed, with respect to privacy preserving mechanisms. The treatment is the interaction between this artifact and the problem context:

the set of features and instances.

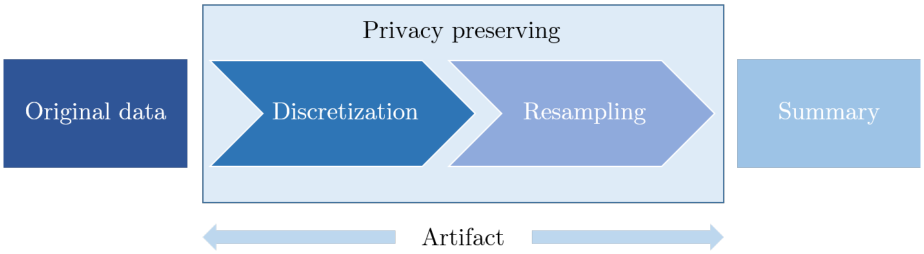


Figure 2.2: Problem context

This problem context provides us with the concepts to be investigated. To understand discretization and resampling with respect to privacy preserving mechanisms, each of these concepts needs to be studied separately. To relate this to the research questions described in Section 2.1, these concepts are studied by providing an answer to **SQ1** and **SQ2**.

When it comes to Privacy Preserving Data Analysis, the aim is to transform the data in such a way that there is some guarantee or requirement in terms of privacy. In this project, we are concerned with the privacy of individuals in a set of continuous features and instances. To provide such guarantees, or to satisfy some privacy constraints, there is a need for quantifying the level of privacy in such a set. Similarly, part of the problem context is to preserve the original distribution. In order to do so, there should be a measure or criterion for determining whether the anonymized data represents the original distribution. This could be called a data utility measure. The first part of **SQ3** is concerned with ways of quantifying both the level of privacy and the level of utility.

2.3.2 Treatment design

The next task of the design cycle is treatment design. As discussed earlier, the treatment refers to the artifact to be designed, with respect to the interaction with the problem context. In other words, the treatment involves the anonymization approach, and the interaction with the data it is applied to. Part of the treatment design phase is to specify the treatment's requirements. These are important during the later validation phase, where the proposed treatment is evaluated. Specifying requirements for the levels of privacy and utility makes it possible to compare the results with these requirements during the validation phase.

During the problem investigation, measures are studied for expressing privacy and utility. The objective of this project is to obtain sufficient levels for both of these measures. This translates into the treatment design. Requirements for the treatment involve incorporating sufficient levels of privacy and utility. Following the question of how to measure these levels, the second part of **SQ3** is concerned with sufficient levels for both.

During the problem investigation, privacy preserving mechanisms are studied, as well as methods for partitioning data. Part of the treatment is the integration and relationship between mechanisms for providing privacy and partitioning and discretization techniques. Ways of doing this are studied during the treatment design phase, by providing an answer to **SQ4**.

2.3.3 Treatment validation

During the treatment validation task, it is assessed whether the designed treatment would contribute to the defined goals if it would be implemented. According to Wieringa (2014), validation *"consists of investigating the effects of the interaction between a prototype of an artifact and a model of the problem context and of comparing these with requirements on the treatment"* (p. 31). In other words, we can apply the artifact to a set of continuous features and instances, and validate whether the effects are in accordance with the requirements specified during treatment design.

One aspect of the validation is how the treatment performs, especially in terms of the levels of privacy and utility. During treatment validation, the effects of applying the artifact are compared with the requirements, so that it can be assessed whether these requirements are satisfied. This assessment would provide an answer to **SQ5**.

Another important aspect of the validation is the trade-off between the level of privacy and the data's utility. To provide very strict privacy guarantees, the data needs to be transformed in such a way that the risk of reconstructing the original data is close to zero. However, such alterations result in a reduction of utility as well, which in turn hurts data analysis. It is therefore valuable to study the effects of the treatment regarding the balance between privacy and utility. More specifically, the effect of increasing and reducing the required level of privacy with respect to the utility is observed. This is addressed by **SQ6**.

The last area of interest during validation is the effect of the bucket size on the levels of privacy and utility. This can be studied by reducing the number of instances to create a summary of the original data, and measuring how it affects privacy and utility. This could potentially be used to determine the minimum amount of instances that are needed to generate an accurate summary of the data. This part of the validation is addressed by **SQ7**.

A concise overview of the application of the design cycle and the connection with the subquestions is illustrated by Figure 2.3.

2.3.4 Thesis structure

The use of design cycle and the allocation of the subquestions to the tasks allows for a convenient structure of the main elements of this research project. The *problem investigation* task is mainly concerned with a thorough understanding

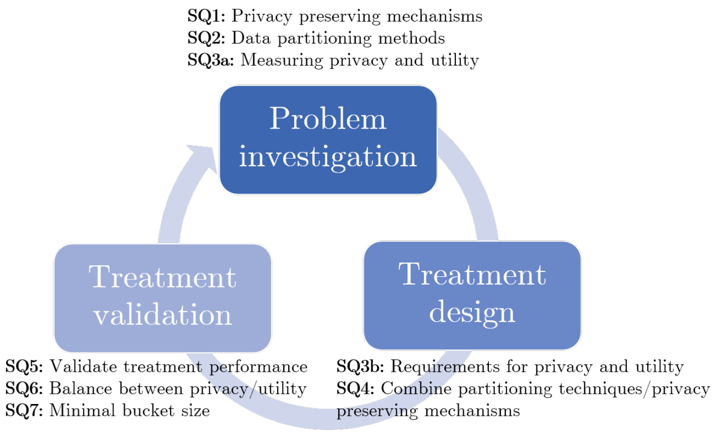

SQ1: Privacy preserving mechanisms
SQ2: Data partitioning methods
SQ3a: Measuring privacy and utility
Problem investigation
Treatment validation
Treatment design
SQ5: Validate treatment performance
SQ6: Balance between privacy/utility
SQ7: Minimal bucket size
SQ3b: Requirements for privacy and utility
SQ4: Combine partitioning techniques/privacy preserving mechanisms

Figure 2.3: Design cycle application

of the problem domain, current practices and the state of the art. This is studied through a literature review. The literature research protocol is explained in Section 2.4, while the resulting theoretical background is discussed in Chapter 3. The *treatment design* involves the design of an algorithm that can be used to transform a data set for the sake of providing privacy guarantees while preserving utility. This is discussed in Chapter 4. The *treatment validation* is performed through an experimental evaluation of this algorithm, of which the setup is described in Chapter 5. The results of the experimental evaluation are discussed in Chapter 6. The relationship between the subquestions, design cycle tasks and the chapters addressing these questions is provided in Table 2.2.

Table 2.2: Report overview

SQ	Design cycle task	Section	Chapter
SQ1 SQ2 SQ3a	Problem investigation	Theoretical background	3
SQ3b SQ4	Treatment design	Treatment design	4
SQ5 SQ6 SQ7	Treatment validation	Results	6

2.4 Literature research protocol

The main concepts that are studied in this project are *Privacy Preserving Data Analysis* and *discretization*. This is also reflected by the subquestions, and their main concern is to study the models and techniques that are available in current literature, as well as approaches to quantify and measure the level of privacy and data utility. To study these concepts, a snowball approach is used to obtain an overview of the available techniques. More specifically, the starting point of the literature search is a recent survey providing a structured overview of the available literature on the concepts. For Privacy Preserving Data Analysis, such a survey is conducted by Mendes and Vilela (2017). For discretization techniques, Garcia, Luengo, Sáez, Lopez, and Herrera (2013) and Ramírez-Gallego et al. (2016) provide extensive overviews of the available methods. From these surveys, the concepts are further studied through the cited literature. This snowballing results in a selection of literature that provide an overview of the developments and state of the art of Privacy Preserving Data Analysis and discretization.

Chapter 3

Theoretical background

This chapter contains a literature review with an overview of the concepts that are relevant for this thesis, including state-of-the-art practices.

3.1 Privacy

To get a thorough understanding of Privacy Preserving Data Analysis (PPDA), the concept of privacy is important to define. Although privacy has definitions beyond the scope of information technology, the used definition in this research project is focused on this perspective. Schoeman (1984) identifies three possible ways to think about privacy from an information perspective:

1. An individual's *claim*, *entitlement* or *right* to determine what information may be communicated to others.

2. A measure of *control* an individual has over the information about himself, and who has access to that information.

3. A *state* or *condition* of limited access to a person.

From these definitions, it follows that what is considered to be private information is determined by an individual. Moreover, an individual should have some *control* over what happens with this information. However, it can be argued whether individuals have any control over the collection and analysis of their personal information, especially in the current information age. In PPDA, the responsibility for the privacy of individuals lies with the actors that process personal or sensitive information. In this sense, Bertino, Lin, and Jiang (2008, p. 3) define informational privacy as:

> *"The right of an entity to be secure from unauthorized disclosure of sensible information that are contained in an electronic repository or that can be derived as aggregate and complex information from data stored in an electronic repository".*

In this definition, whether an individual participates in a data set or not, and whether this is by choice or not, the individual's personal, private, or sensitive information must be protected from unauthorized disclosure.

3.2 Privacy Preserving Data Analysis

In the analysis of large data sets, issues can arise with confidentiality or privacy. Various techniques and concepts exist to alter the data for the sake of anonymity of the records of individuals in a data set. However, these alterations result in a reduction in data quality, or utility. PPDA is concerned with techniques, methods and models to anonymize a data set while keeping a desired amount of data utility (Aggarwal & Philip, 2008).

A taxonomy can be used to classify different types of personal information that could exist in a data set (Bezzi, 2010). The underlying concept is the possibility to identify a specific individual from the data. The most obvious information that could identify an individual is data containing names or social security numbers. These are known as explicit identifiers, as the possibility that this information reveals the identity of an individual is large. Another type of identifiable information is known as quasi identifiers (QIDs). These do not specifically reveal one's identity, but in combination with background information or additional, publicly available information, they could potentially lead to the identification of an individual. Examples are attributes containing age, zip code, or city of residence. The last type of attributes are typically known as sensitive attributes, which themselves do not identify individuals, but contain information that one would not want to be disclosed or known publicly. For example, people might want to keep their income or medical condition a secret. An overview of these concepts and examples of them are shown in Table 3.1. It is worth noting that this taxonomy provides some way of thinking about different types of information, although the distinction between them is not always as clear-cut. For example, it is not unthinkable that the *income* and *disease* attributes can be used as identifying attributes.

Table 3.1: Attribute types and examples

Explicit identifier	**Quasi identifier**	**Sensitive attribute**
Name	Age	Income
Social security number	Zip code	Disease
	City	Political orientation
	State	
	Country	

When publishing data sets, dealing with explicit identifiers is a trivial task, as names or identification numbers can and should simply be removed from data. However, only removing explicit identifiers has been shown to be insufficient. For example, Samarati (2001) shows how an anonymized data set with medical information could be linked to publicly available information to identify individuals in the data set. This is known as a *linkage attack*. Moreover, Sweeney (2002) used a public voter list to identify the governor of Massachusetts in the medical data, thereby revealing his medical information. In this example, the QIDs from the medical data were coupled with those from the voter list to gain additional information, and to reveal sensitive attributes. Similarly, Narayanan and Shmatikov (2008) illustrate how auxiliary information can be used to de-anonymize a data set containing anonymous movie ratings by Netflix users. Lastly, De Montjoye et al. (2013) show that even in a data set containing more than a million records, individuals could be uniquely identified with 95% certainty with a small amount of additional information.

Although it is unlikely that the risk of disclosing personal information could be removed completely (while maintaining a reasonable amount of data utility), various models exist to ensure a certain level of privacy, to avoid linkage attacks, and to create a common ground for quantifying privacy.

3.2.1 *k*-anonymity

Related to linkage attacks, k-anonymity is concerned with sets of quasi identifiers. A data set is said to be k-anonymous if a record's identifiable attributes cannot be distinguished from at least k-1 other records (Sweeney, 2002). Take for example a data set containing the QIDs *age* and *city*. If there is a record containing *49* and *Amsterdam* for these attributes respectively, with k set to 2, this model states that there should be at least one other record with the values *49* and *Amsterdam*. This set of records with identical values is known as an *equivalence class*. In a k-anonymous data set, there are at least k records in each equivalence class in the data. The advantage of this model is that when the combination of QIDs is known, it cannot directly be used to uniquely identify one individual from the data. Instead, at least k records are returned from such a query.

3.2.2 *l*-diversity

The l-diversity model takes k-anonymity as a starting point, but then requires that for every equivalence class there exists at least l 'well represented' values for each sensitive attribute (Machanavajjhala, Gehrke, Kifer, & Venkitasubramaniam, 2006). There are different methods to determine whether values are well represented. In any case, in the example, if we would require our data to be 2-diverse, both records having an *income* of 30,000 would not satisfy 2-diversity. Instead, there should be two different values for *income* and any other sensitive attribute. This ensures that an adversary could not find sensitive information on individuals in the data set, just by having background knowledge or publicly available information on the combination of QIDs.

3.2.3 t-closeness

Intuitively, l-diversity provides more security by extending the k-anonymity model. However, it has its own drawbacks. The first is the consideration of the distribution in the data. For example, consider a sensitive attribute containing people's voting behavior. In this data set, 99% of the individuals in the data voted for Party A, while 1% voted for Party B. The l-diversity principle could enforce that for a certain equivalence class, 50% of the records contain Party A, and 50% contain Party B for this attribute. As the *overall* distribution of values in the data for this attribute is highly skewed, having an equivalence class where both values are equally represented reveals more information on an individual. By applying l-diversity, the probability of someone in this equivalence class voting for Party B increases from 1% to 50%. This principle is known as *skewness attack*.

Another threat of the l-diversity model is the *similarity attack*. Consider an equivalence class where records contain values for the *income* attribute of 28k, 29k, 30k, and 31k. As these are all distinct values, they could satisfy 4-diversity. However, an adversary can still gain knowledge from these values, as they are relatively close together.

The t-closeness principle deals with these types of attacks, and is concerned with the distribution of values for sensitive attributes within equivalence classes compared with the overall population (Li, Li, & Venkatasubramanian, 2007). It makes sure an adversary does not gain much information by knowing the equivalence class of an individual, and consequently gaining knowledge through the distribution of sensitive values in that equivalence class compared to the overall distribution. More precisely, an equivalence class has t-closeness if the distance between the distribution in this class and the distribution in the whole data set is smaller than threshold t. In our example where 99% would vote for Party A and 1% for Party B, t-closeness would ensure that in equivalence classes, there would also be a division similar to 99%/1% for that particular sensitive attribute.

3.2.4 ϵ-differential privacy

Another principle is that of ϵ-differential privacy (or just differential privacy). This means that the inclusion or absence of a record in a database does not substantially change the outcome of the analysis. In other words, the difference between analyses on a data set *with* and a data set *without* that record should be smaller than ϵ (Dwork, 2008). Contrasting the previously discussed mechanisms, differential privacy has a strong mathematical foundation. Consider a mechanism $\mathcal{M}$, that answers queries on a data set $\mathbf{x}$. A neighbouring set $\mathbf{x}'$ is different from $\mathbf{x}$ in one record. The mechanism $\mathcal{M}$ is ϵ-differentially private if for all neighboring data sets $\mathbf{x}$, $\mathbf{x}'$, and for all events (measurable sets) S in the space of outputs of $\mathcal{M}$ (Dwork, McSherry, Nissim, & Smith, 2017):

$$\Pr(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\epsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S)$$

In other words, when querying two sets that differ in just one record, the difference in outcome should be smaller than ϵ. The advantage of this principle over the ones mentioned earlier is that differential privacy does not take into account any background knowledge or knowledge gained from other data sources. This is especially useful since auxiliary knowledge does not just exist in the past, any potential future knowledge should be considered as well. As auxiliary knowledge is not a concern with differential privacy, it is not necessary to take knowledge that could possibly exist in the future into account, or to make any model of an adversary's knowledge for that matter. Differential privacy also provides a more intuitive sense of privacy for the individual than previously discussed privacy models. Whether an individual participates in a data set or not, the statistics of the data should not change by a large amount.

Another way of looking at differential privacy is that an adversary should not gain any posterior knowledge on individuals in the data set that is greater than its prior beliefs. This poses the fundamental challenge of data analysis while preserving privacy. Gaining knowledge is one of the goals of storing and analyzing data. However, data that is too accurate harms the privacy of individuals. Differential privacy supports the approach to limit the knowledge gained on *individuals* from the data, while knowledge is gained on the *population* in the data.

The earlier definition of differential privacy is not an algorithm to achieve this privacy model. Instead, it places constraints on the mechanism $\mathcal{M}$. If $\mathbf{x}$ and $\mathbf{x}'$ differ in just one row, the difference in output of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ should be minimal, with ϵ limiting their difference. This allows for different approaches to achieve differential privacy.

One possible approach is that of *randomized response*, introduced by Warner (1965) to provide anonymity in surveys with sensitive questions. Consider the case where the question is posed whether someone voted for Party A. Before replying with YES or NO, the respondent flips a coin. If it's heads, the respondent answers truthfully. When it is tails, the respondent flips another coin, and replies with YES or NO based on the outcome of the second coin flip. This technique works well at the data collection stage, but could also be used with learning techniques like classifiers, without losing too much accuracy (Du & Zhan, 2003).

A popular technique that works with numeric attributes is the Laplace mechanism. Based on the magnitude by which an individual's data can change an outcome, the mechanism adds noise randomly drawn from a Laplace distribution. Since the outcome of $\mathcal{M}(\mathbf{x})$ contains random noise, the information gain by comparing the outcomes of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ is limited (Dwork & Roth, 2014; Dwork et al., 2017).

Dwork et al. (2017) make a distinction between two scenarios for PPDA: *interactive* and *non-interactive*. In the latter, a data set to be published is altered through some algorithm, resulting in a 'sanitized' or anonymous data set which can be used for analysis. In an interactive setting, the underlying data set is not released. Instead, a mechanism is employed that answers queries on the data set. These queries typically involve aggregates like the sum, average, or count of instances in a data set that have some value x for attribute f. Achieving differential privacy would be more difficult in a non-interactive setting, since the algorithm has to take into account all possible queries on the data. In an interactive setting, a differentially private result can be generated based on the query at that time. Most literature on the subject therefore discuss mechanisms to achieve differential privacy in an interactive setting.

Differential privacy is the most recent influential development in PPDA, and is considered the standard when attempting to achieve information privacy. Some organizations have created their own implementation of differential privacy. For example, Google developed RAPPOR, a technology that allows for the analysis of the activity and usage of client-side software by end-users (Erlingsson, Pihur, & Korolova, 2014). In the spirit of differential privacy, RAPPOR achieves high data utility that allows for the analysis of a group of users, while the information on individual users is limited. To provide strong privacy guarantees, RAPPOR achieves ϵ-differential privacy with ϵ set to $\ln(3)$.

Another attempt is made by Apple, with the goal of analyzing user behaviour of Apple customers. Examples are the most popular emojis, new and emerging words, and which web sites cause high energy and memory usage when using Safari (Apple, 2017). Although the attempt to achieve differential privacy by organizations with such a large user base is a positive development, the effectiveness of the implementation can be questioned. By reverse engineering the privacy components in MacOS 10.12, Tang, Korolova, Bai, Wang, and Wang (2017) argue whether Apple's implementation actually provides strong guarantees about a user's privacy. According to Apple (2017), the ϵ-values range from 2 to 8. Although the choice of ϵ is in essence a social question, Dwork (2008) considers reasonable ϵ-values to be 0.01, 0.1, or $\ln(2)$ to $\ln(3)$ in some cases. Whether the implemented values for ϵ are sufficient in a certain environment is difficult to assess. However, it is clear that higher ϵ-values provide less privacy guarantees. In addition, the differences between small values (between 0 and 1) of ϵ are much more significant than differences between larger values.

Although the advocators of differential privacy are confident about the guarantees and promises it offers due to its mathematical foundations, some skepticism exists. Sarathy and Muralidhar (2010) show that adding noise from a Laplacian distribution to achieve differential privacy could potentially reduce data quality to such an extent that the data becomes meaningless. Moreover, Bambauer, Muralidhar, and Sarathy (2013) question the practical use of differential privacy altogether. They claim that the answers of simple statistical questions like averages or correlations would be *"gibberish"*, and the application in regression and other complex analyses would be very difficult. To take full advantage of differential privacy, the release or exchange of traditional data sets would be impossible as well. These statements caused a discussion between proponents and the authors of the article (McSherry, 2016a, 2016b; Bambauer &

Muralidhar, 2016). Although many are attracted by the promises of differential privacy, it is clearly not universally accepted. In addition, there are only a few cases where differential privacy is used in practice. The most notable attempts are made by Erlingsson et al. (2014) and Apple (2017).

3.3 Discretization

Data exists in many forms, and the format of data can be classified in different ways. It is common to differentiate between nominal, discrete, and continuous data. With nominal data, there is no useful order between values, while there is an order in values with discrete and continuous data. While continuous data can take an infinite amount of possible values, the set of possible values is finite with discrete data.

In data analysis, it is useful to avoid the use of continuous attributes. For example, using continuous attributes in a decision tree algorithm reduces the speed and accuracy of learning. Moreover, discrete values are easier to understand, use, and explain for both users and experts (Liu, Hussain, Tan, & Dash, 2002). Various advanced data analysis algorithms can therefore only deal with discrete attributes, thereby requiring or implementing an additional step of discretization. As continuous features are common, there is a wide variety of discretization techniques that transform a continuous feature to a discrete one.

In the context of PPDA, continuous attributes are *by nature* less private than discrete attributes. For example, consider a data set containing 100 residents of a certain street, and a continuous attribute for *age*. If there would be only one resident with an age of *45*, and an adversary with auxiliary knowledge on the age of that resident, the individual could be identified from the data, along with possible sensitive information. If instead, the individual's age would be presented as the interval $[41, 50]$, the identification risk of that individual is reduced. This is because it is likely that more residents have an age within $[41, 50]$, so it becomes more difficult to identify just one individual from the data. This generalization approach is used to achieve privacy models including k-anonymity (Mendes & Vilela, 2017), and is closely related to discretization.

Discretization can be considered as a preprocessing technique for data analysis. As previously described, discretization functions as a data preparation step, since some learning algorithms require discrete instead of continuous values, or have a higher performance with discretized attributes. Additionally, discretization functions as a data reduction step as well (García, Luengo, & Herrera, 2015). This is due to the fact that a continous attribute can contain a large spectrum of possible values. Discretization algorithms can reduce this spectrum to a limited number of intervals, preferrably the smallest number of intervals while still representing the original values well.

The result of a discretization process is a finite set of values. To arrive at this point, each value from a continuous attribute is assigned to one of the possible discrete values. The general discretization approach can be described in four steps (Ramírez-Gallego et al., 2016). First, the continuous values are sorted in descending or ascending order. In the second step, cut points are evaluated to either split or merge adjacent intervals, based on some evaluation measure. In the next step, the candidates that satisfy a condition are split or merged, depending on the discretization technique. Lastly, the discretization process stops at some point, with the goal of limiting the resulting number of intervals. At the two extremes of discretization, either all values are assigned to one interval, or there are intervals for every continuous value. More intervals means more precision but less understandability, while fewer intervals sacrifices accuracy for understandability. The stopping criterion is used to control this trade-off.

Discretization techniques can be categorized in different ways. Similar to the categorization of machine learning algorithms, there are supervised and unsupervised discretization methods. Supervised methods take the class labels of instances into account, while unsupervised methods do not (Dougherty, Kohavi, & Sahami, 1995). In addition, discretizers can be either univariate or multivariate. Univariate approaches look at one attribute at a time when deciding cut points and deciding on intervals. On the other hand, multivariate approaches take the whole set of attributes to be discretized when deciding on cut points and intervals. These approaches are especially useful in the context of machine learning algorithms, since the interdependence between attributes is particularly interesting (Garcia et al., 2013). Lastly, discretizers can either split or merge intervals to arrive at the desired set of intervals. In the former case, the algorithm starts with an interval containing all attribute values and splits until some stopping criterion. In the latter, there is an interval for all continuous values, and intervals are merged until some stopping criterion. Taxonomies of discretizers are described in more detail by Dougherty et al. (1995); Garcia et al. (2013); Kotsiantis and Kanellopoulos (2006); Liu et al. (2002); Ramírez-Gallego et al. (2016).

A convenient discretization method is the *equal-width* discretizer. In this approach, all intervals or bins have the same value range or width. With k bins, the bin size δ for an attribute x with values between x_{min} and x_{max} is defined as (Dougherty et al., 1995):

$$\delta = \frac{x_{max} - x_{min}}{k}$$

In contrast, when each bin has the same number of instances, it is called an *equal-frequency* discretizer. These are both examples of unsupervised approaches, since class labels are not taken into account.

Holte's 1R discretizer (Holte, 1993) is a common supervised approach. Bins are created with instances that preferably have the same class label. Since this could result in very small and too many bins, a restriction is imposed on the minimum number of instances that a bin can contain.

Another supervised discretizer is the Chi2 algorithm, which is an improved version of ChiMerge (Kerber, 1992). This method uses the statistical test χ^2 to evaluate which intervals should be merged (Liu & Setiono, 1997). The algorithm makes sure the discretized values represent the original, continuous distribution.

Apart from discretization, Chi2 checks for consistency in an attribute with regards to the class label, and inconsistent features are discarded. Chi2 therefore functions as a feature selection algorithm as well.

Similar to Chi2, the Omega algorithm functions as both a discretization and a feature selection technique (Ribeiro, Ferreira, Traina Jr., & Traina, 2008). Like 1R, Omega employs a parameter to limit the minimum interval size. It also uses an inconsistency rate to determine which intervals are merged. For feature selection, a global inconsistency rate is used to determine which features should be discarded. Compared to other discretization and feature selection methods, Omega results in a small error rate when training a C4.5 classifier, while the number of nodes of the resulting decision tree is relatively small.

Discretization can also be achieved using tree based density estimation. With this technique, the best cut point is found in a step-wise fashion (Schmidberger & Frank, 2005). Similar to the construction of classification trees, a tree is created which initially contains one node, containing all training instances. The cut point that maximizes the likelihood based on the training data and the ranges of the current node is chosen, and two new nodes are created. To prevent overfitting, 10-fold cross validation is used to determine the optimal number of splits, and this number is used as a stopping criterion for the construction of the tree. The density estimation tree results in a number of intervals with varying width, and does not take class labels into account.

Fayyad and Irani (1993) propose a supervised discretization approach that uses information entropy as an evaluation measure for discretization. In addition, their approach uses the Minimum Description Length Principle (MDLP) (Rissanen, 1978) as a stopping criterion. This widely used discretization technique is among the better performing discretizers, and one of the most compared techniques in literature (Garcia et al., 2013; Ramírez-Gallego et al., 2016).

So far, only univariate approaches were discussed, since most of the existing discretization techniques consider only one variable. This means that variables are discretized separately without regarding other possible variables. In data analysis tasks however, we are interested in finding patterns, clusters or relationships between variables and possibly some class label. For example, consider the dimensions X_1 and X_2 forming a cluster together. However, in isolation, no meaningful patterns can be extracted from both the dimensions. Discretizing X_1 and X_2 separately would ignore the relationship between the two, so this relationship could potentially be lost. Multivariate or multidimensional discretization methods do regard multiple variables, with the objective of preserving the relationships between them.

A common multivariate approach is to first partition each continuous attribute in n basic regions, and then to merge intervals of instances that have a similar distribution over multiple dimensions (Bay, 2000). In other words, intervals X and Y are merged if $F_x \sim F_y$. Any attempt at such a multivariate approach requires a way of comparing F_x and F_y. Bay (2000) describes a Multivariate Discretization (MVD) approach, where a test of differences called STUCCO is implemented. This test involves determining the *support* of a combination of attributes. In this case, support simply means the percentage of observations where a combination of attribute values holds, with respect to the total number of observations (Bay & Pazzani, 1999).

Since the continuous attributes were already partitioned to some extent, it is possible to compare the support of the different groups of instances G_1 and G_2, for a specified attribute-value pair C. That is, we compare $P(C|G_1)$ and $P(C|G_2)$, for example, $P(\text{sector} = \text{IT} \mid \text{male})$ and $P(\text{sector} = \text{IT} \mid \text{female})$. So, the support is a measure for defining the difference between two groups, and is used to determine whether or not these groups should be merged. When the support of the two groups are similar, i.e. if $F_x \sim F_y$, the groups are merged. When the difference in support is too large, a discretization boundary remains between the two groups. It should be decided how large the differences in support can be while claiming that the groups have a similar distribution.

Bay's MVD shows the ingredients for multivariate discretization approaches. Instead of using measures like χ^2 or an inconsistency rate, which works in univariate cases, merging is done by determining and comparing the distribution of *groups* of multivariate instances. Comparing the interdependence of attributes between groups is a major task of multivariate discretization, since its goal is to preserve this interdependence. Since determining a multivariate distribution is not a trivial task, different discretization approaches make use of different measures.

In MVD, *support* is essentially a measure for the distribution over multiple variables. Mehta, Parthasarathy, and Yang (2005) use Principal Component Analysis (PCA) to identify the interdependence between continuous attributes, along with a similarity measure based on Association Mining principles. Wei (2009) adopts a density-based clustering algorithm by Ankerst, Breunig, Kriegel, and Sander (1999) to find regions that could hide possible patterns in the data. In their effort to find a way to measure distances between multivariate distributions, Nguyen, Müller, Vreeken, and Böhm (2014) propose the *Interaction Distance* (ID), which doesn't require any assumptions on the distribution of variables. Their discretization algorithm uses MDLP (Rissanen, 1978) to find the optimal number of cut points for a dimension X_i. Then, X_i is discretized using ID, so the interaction between X_i and the other dimensions is preserved.

3.4 Sampling

From a statistics point of view, sampling is required for making inferences about a population. Additionally, most data sets are in fact already a sample from some population. If there would be access to a whole population, it does not make sense to construct statistical or machine learning models. Instead, we would just be querying the available data. However, we usually do not have access or knowledge about the whole population. In addition, with classifiers or regression models, we want to make statements about *future* examples. This is why samples from a population are taken, and statistical tests are performed to make inferences about the population. When it becomes expensive or impractical to analyze a sample, we can sample from an artificial population. As this is usually the data set itself, it can be seen as taking a sample from a sample of the real population, hence technically called *resampling* (Kaplan, 1999). In the data mining context however, creating a subset of a data set can also be referred to as data sampling (Aggarwal, 2015), or instance selection (Olvera-López, Carrasco-Ochoa, Martínez-Trinidad, & Kittler, 2010; García et al., 2015).

There are multiple reasons for using resampling. For example, the Big Data era comes with a number of challenges. Properties of Big Data include its variety, volume, velocity, variability, complexity, and value (Katal, Wazid, & Goudar, 2013). Most data mining techniques are not able to process such large volumes, in combination with the velocity of data (Wu, Zhu, Wu, & Ding, 2014). This brings up the need to process just a subset of large data sets, or the implementation of Big Data solutions including distributed and decentralized data processing.

In many learning algorithms, it is common to divide the available data in two parts: a training set and a test set (García et al., 2015). This prevents both overfitting and underfitting, and helps ensuring that the model generalizes to new cases. One approach is to use a random majority of instances as a training set, and to use the remaining instances to validate the trained model. Another common technique is k-fold cross validation, in which the data is randomly partitioned in k sets of equal size. Then, k-1 folds are used for training and one is used to validate the model. This is repeated until every fold is used to validate the model (Kohavi, 1995). Taking this even further, *leave one out* requires k to be equal to N, the number of instances in the data. This results in a very high computational complexity, especially in large data sets.

Sampling can be related to privacy as well. If there is a data set containing records with personal information, creating a smaller subset intuitively reduces the risk of information leakage (Ebadi, Antignac, & Sands, 2016). Joy and Gerla (2017) also show how differential privacy could be achieved just by using a sampling technique. Lin, Wang, and Rane (2013) apply an approach to sampling to preserve privacy in the aggregation of distributed databases. By using random sampling, Li, Qardaji, and Su (2012) show how a k-anonymity algorithm achieves differential privacy, since the random sampling step adds uncertainty about whether or not an individual is included in the subset.

Part II

Treatment design

Chapter 4

Treatment design

4.1 Goals

The goal of this research is to find an answer to the research question stated in Section 2.1: *How can data be accurately summarized by as few instances as possible to support data analysis, while preserving the privacy of individuals?* To do so, an algorithm is developed and validated through experimental evaluation. The objective of this algorithm is to transform a set of continuous features and instances in such a way that both the privacy of individuals and the utility are preserved. An additional part of the evaluation is to achieve these objectives with as few instances as possible. The anonymization approach is a combination of privacy preserving mechanisms and discretization techniques. The result of the transformation is a set of features and instances that provides guarantees on the privacy of individual records, while mostly maintaining the utility of the original data.

The remainder of this chapter is dedicated to describing the proposed approach to anonymization through discretization and the rationale behind it, relating to **SQ3b** and **SQ4**.

4.2 Privacy preservation

Over the past two decades, many privacy preserving models have been proposed to achieve privacy in data sets. The most well-known were discussed in Section 3.2, but there are many possible variations for these models. Differential privacy is considered to be the most favorable model these days, mainly due to its strong mathematical foundations. In addition, there is no need to consider auxiliary information that can be used to perform linkage attacks. However, as discussed in Section 3.2.4, differential privacy has some drawbacks as well. In this project, aiming for an implementation of differential privacy is not suitable to achieve the goals, for a number of reasons. Firstly, this project's aim involves the transformation of a data set in such a way that it still represents the original data, while the privacy of individuals in the data is not harmed. This makes it possible to release or share the data with external parties, and to use the data with commonly used mining and learning tasks, without compromising the privacy of any of the individuals in the data. These objectives call for an approach that works in a non-interactive setting, while differential privacy's strenghts are mainly found in the interactive setting.

Another difficulty is the need for a privacy property that can be placed on the anonymized data, so the privacy level can be measured. With differential privacy, this property holds for the mechanism instead of the data.

In the third place, a differentially private mechanism is specifically designed for a data set, since the amount of noise that is added to the query answers heavily depends on the characteristics of the data. For this project, the aim is to have a general-purpose algorithm that can be applied to different data sets, making differential privacy not a suitable model to pursue.

A model that does work in these circumstances is k-anonymity. Although it has its drawbacks as well (Domingo-Ferrer & Torra, 2008), k-anonymity is a suitable model to achieve privacy through discretization. First of all, it is one of the more intuitive and easy to understand models that exist. Secondly, the *re-identification risk*, the risk that a specific individual can be identified from the data, can be determined intuitively as well. If we require that for every equivalence class there are at least k records in the data, the *maximum* re-identification risk is $1/k$. Thus, higher values for k reduces the re-identification risk, and an acceptable *threshold* can be decided on. In addition, more sophisticated measures can also be used to determine the risk of re-identification more precisely (El Emam & Dankar, 2008; Rebollo-Monedero, Parra-Arnau, Diaz, & Forné, 2013). Another consideration is the way k-anonymity is achieved. To make sure that there are at least k records for each equivalence class, *generalization* and *suppression* of attribute values are common techniques (Samarati, 2001). Suppression simply involves removing an attribute's value. On the other hand, generalization would replace an attribute value with a more general value.

Different approaches for generalization with respect to k-anonymity were proposed. A possible approach is to create a generalization hierarchy for the attribute to be generalized (Samarati, 2001). In the *ARX Data Anonymization Tool* by Prasser and Kohlmayer (2015), such a hierarchy can be created for both continuous and categorical attributes. An example of such a hierarchy for continuous attributes is shown in Figure 4.1. The hierarchy in this example represents an *age* attribute in a data set. To satisfy k-anonymity for some level of k, values of this continuous attribute can be generalized bottom-up. For example, if a record would contain the value *34* for this attribute, generalization would replace this value with $[17, 39]$. If more generalization is required in order to satisfy k-anonymity, this would be done by going up a level in the hierarchy, and transform into $[17, 61]$. The highest level interval contains the whole range of the attribute. With the ARX tool, it is possible to create such a hierarchy manually for each attribute. However, in this project, the idea is to incorporate this in the anonymization algorithm. Doing so eliminates the tedious task of creating generalization hierarchies by hand. In addition, as discussed in Section

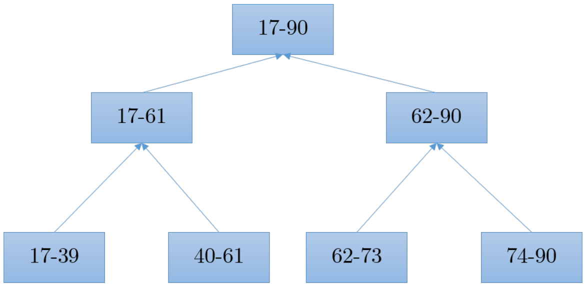


Figure 4.1: Generalization hierarchy example

3.3, generalization of continuous attributes is closely related to discretization. Using discretization techniques to achieve k-anonymity could therefore ensure that the intervals that are created represent the distribution of the data.

As discussed in Chapter 3, the concept of k-anonymity is to combat *linkage attacks*, in which auxiliary knowledge is used to identify specific individuals by linking quasi-identifying attributes from different sources. As critics of k-anonymity have pointed out, this requires a model of the auxiliary knowledge available to the adversary. Theoretically, this requires a model of all available knowledge to any adversary, now and in the future. It is very unlikely that such a model would ever be complete. Instead, it would be safer to assume that an adversary knows everything, and that any attribute in a data set could potentially identify an individual. Therefore, every continuous attribute in a data set is included in the transformation process to evaluate the discretization algorithm in this research project.

4.3 Discretization

The scope of this project involves anonymizing a set of continuous attributes and instances. A continuous numeric variable can be transformed into a discrete set of bins. Different approaches to do so were discussed in Section 3.3. In general, a discretization approach would sort attribute values, and either split or merge consecutive intervals based on some criteria. The aim of these criteria is to preserve the characteristics of the variable, and to decide which split or merge candidate would yield the best results. Discretization typically results in a set of cutpoints or intervals.

Ram and Gray (2011) introduced the concept of *Density Estimation Trees* (DETs). DETs can be used for the fundamental task of density estimation of multivariate data, and are analog to classification or regression trees, commonly used in learning tasks. Similarly, Schmidberger and Frank (2005) explored the possibility to perform discretization using tree-based density estimation. The underlying idea of this approach is to iteratively partition the data, while the resulting decision tree represents an estimation of the *probability density function* of the data. Based on this concept, the approach to discretization that is used in this project is a *tree-based* discretization approach.

One advantage of using a tree-based approach is that it is easy to interpret and visualize. Decision trees are widely used in learning tasks, for example classification and regression trees, which are especially powerful with Random Forest (Breiman, 2001). It is a flexible and intuitive concept that practitioners are familiar with.

The main reason for using tree-based discretization is that it yields a structure that is similar to a generalization hierarchy like the one shown in Figure 4.1. This means that the starting point is the complete value range, which then is iteratively partitioned. In other words, the root node would contain all of the instances, while lower level nodes contain progressively fewer instances. This is a useful property in the current context. Lower level nodes with smaller interval ranges potentially make for a more accurate model of the data, but it does not benefit data privacy. Instead, more privacy is achieved at the higher level nodes, since these intervals have a wider range, and a larger number of instances would fall into this range. However, wider, more general intervals naturally leads to a less accurate representation of the data.

One of the essential ingredients of tree-based models is some evaluation criterion to decide on the best values (and attributes) to split a node. With classification trees, examples of evaluation criteria include the misclassification error, Gini index and entropy (Moisen, 2008). DETs need evaluation criteria to determine the best partitioning as well. In this case, the goal of such criteria is to find a structure that best represents the distribution of the data. Schmidberger and Frank (2005) use the log-likelihood to select the best cutpoints in their univariate approach of discretization through tree-based density estimation. Ram and Gray (2011) describe a multivariate approach, where the best split point is chosen by minimizing the Integrated Squared Error (ISE), an approach also used by Anderlini (2016). This multivariate approach would be suitable for the efforts to achieve data privacy through discretization, especially when applying k-anonymity.

4.4 Privacy preserving discretization

The approach to achieve privacy in a set of continuous features and instances involves discretization through Density Estimation Trees to achieve k-anonymity for a desirable level of k, while aiming to preserve data utility. This is a novel approach, since DETs have not been used before for achieving data privacy. It takes the existing concept of discretization through DETs (Schmidberger & Frank, 2005; Ram & Gray, 2011; Anderlini, 2016), but incorporates k as a minimum leaf stopping rule during the construction of a DET. Since the DET's leaf nodes effectively represent equivalence classes, this novelty can be used to partition and assign data to their respective equivalence classes, thereby achieving k-anonymity.

The DET would yield a discretization structure similar to a generalization hierarchy. This is used to assign the continuous numeric attributes to a discrete number of bins, thereby transforming each value to an interval. Consider the hierarchy shown in Figure 4.1. Each of the continuous values of this *age* attribute would be discretized by assigning them to the *leaf nodes* of the hierarchy, i.e. $[17, 39]$, $[40, 61]$, $[62, 73]$, and $[74, 90]$.

Constructing a DET involves iteratively partitioning node t in two new child nodes t_L and t_R, both containing a subrange of t. To determine the best split, the approach by Ram and Gray (2011) is used. They use the integrated squared error (ISE), and show that the error for any node t can be defined as:

$$R(t) = -\frac{|t|^2}{N^2 V_t}$$

where $|t|$ is the number of instances in node t, N is the total number of instances, and V_t is the proportion of the volume of the data associated with node t. A DET is then constructed top-down by maximizing the error's reduction in each node, i.e. $R(t) - R(t_L) - R(t_R)$.

In terms of discretization, using DETs provide an *unsupervised*, *multidimensional* approach. As a result, it can be applied to most data sets containing continuous attributes, not just in the context of supervised learning. In addition, the multidimensional nature of it means that it tries to preserve the interaction between attributes, which is important in the context data analysis like learning tasks. The assumption is that potential patterns not only reside within each dimension individually, but also come from the interplay between dimensions. A multidimensional approach takes the multidimensional structure into account, thereby aiming to preserve the multivariate distribution. Additionally, DETs are *nonparametric* density estimators. This means that the method does not require assumptions on the distribution of the data.

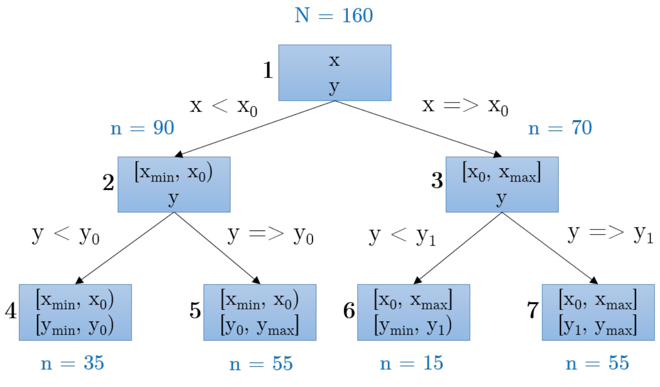


Figure 4.2: Discretization tree example

Figure 4.2 shows how DETs are used for multidimensional discretization. The tree starts with the root node, containing all instances with values for the attributes x and y. The first division is performed by splitting on x, where instances with values lower than x_0 will go to the left node, and those with higher values to the right. This is illustrated by the intervals $[x_{min}, x_0)$ and $[x_0, x_{max}]$, indicating all values of x until x_0, or from x_0 onwards respectively. These nodes are split further on y_0 and y_1. The result of this process is a Density Estimation Tree that can be used as a discretization structure. The leaf nodes of this tree represent the bins to which the continuous values can be assigned to. In this example, there are four leaf nodes, so the data would be partitioned in four exclusive bins. The range of these bins is composed of the *cut point* value on one side, and either the *minimum* or *maximum* value of the attribute value among the instances in that bin. If an attribute is not divided into multiple intervals, the interval would simply be the complete range of that attribute.

So far, one essential aspect of a tree-based algorithm, the evaluation criterion is discussed. Another important part of such an algorithm, and of discretization techniques in general, is deciding how many nodes or bins are needed to obtain an accurate representation of the data. In theory, a decision tree that best fits the available instances could contain nodes for each individual instance. However, such a model would not generalize well for new instances, but it would be overfitting the available observations. In the construction of decision tree models like classification trees or DETs, the algorithm should contain an approach to avoid overfitting models. For example, cross-validation is widely used to estimate the optimal number of cuts that best represents the available data (Schmidberger & Frank, 2005; Ram & Gray, 2011). However, Anderlini (2016) argues that cross-validation is too expensive, and prefers an *a priori* solution to prevent overfitting in DETs, namely a *minimum leaf* constraint. By specifying how many instances a node should contain at least, nodes containing only a few instances are never created in the first place. Since the aim of k-anonymity is to end up with at least k instances per equivalence class, this can be achieved a priori by specifying a minleaf that is equal to k. Increasing k (minleaf) then means that fewer leaf nodes are created, containing more instances. This would lead to a less accurate model, but results in a higher privacy level according to k-anonymity. Lower k-values would allow more splits, containing fewer instances. While this provides less privacy guarantees, it could result in a more accurate model. Table 4.1 shows the resulting data when the tree from Figure 4.2 is used to discretize the hypothetical data.

Table 4.1: Discretized data

#	**x**	**y**
1..35	x_{min} - x_0	y_{min} - y_0
36..90	x_{min} - x_0	y_0 - y_{max}
91..105	x_0 - x_{max}	y_{min} - y_1
106..160	x_0 - x_{max}	y_1 - y_{max}

This example shows how k-anonymity is achieved through DET discretization. The DET is constructed with $minleaf = k$. The original instances are then assigned to the leaf nodes or bins. These values are then essentially generalized from the more unique, continuous value, to a more general one. This means that all the instances in a bin have the same values. In the terminology of k-anonymity, we can consider the attributes $\{x, y\}$ as the quasi-identifier. In the example, we have four equivalence classes, since there are four combinations of $\{x, y\}$, represented by the leaf nodes. The number of instances in each leaf node represents the number of records within each equivalence class. Since the *minleaf* constraint is set beforehand, each equivalence class contains at least that amount of instances. In this example, leaf node 6 is the smallest bin with 15 instances, which means that k was set to 15 or less.

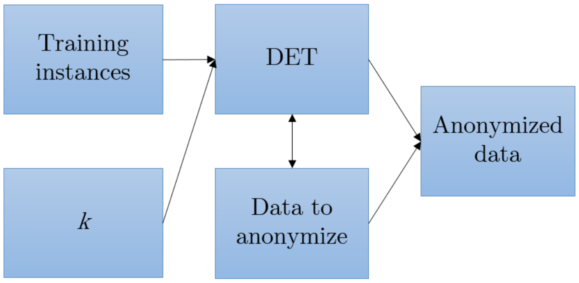


Figure 4.3: Anonymization approach

This approach is essentially comprised of two parts. The first one is constructing the DET, requiring the data set and k as input parameter. The second is to use this discretization structure to discretize instances for the sake of achieving k-anonymity. It is very much possible to use the entire data set to be anonymized as input data for the DET, and then discretize this data. However, this approach also allows for the construction of the DET on a subset of the data, and then use this model to discretize new instances. This could be useful when using larger data sets to improve the performance. It can also be useful when data is continually generated, and new instances need to be anonymized. At the same time, newly generated data can be used to further improve the model. The approach is illustrated by Figure 4.3. This approach allows for the evaluation of the amount of training instances that are needed to create an accurate representation of the data as well. The evaluation can be done on the complete data, and on subsets of it, to evaluate whether less instances can still provide an accurate representation of the data.

4.5 Measures

As discussed in Sections 3.2 and 4.2, k-anonymity requires that for every equivalence class, i.e. for every combination of attribute values, there exists at least k records. Additionally, the re-identification risk is $1/k$ at maximum. With k-anonymity, k is the level of privacy. Higher values for k increases the level of privacy, and reduces the re-identification risk. However, k is an input parameter as well, so it should be set to a desirable level. Unfortunately, there is no optimal level of k. A desirable level of k depends on the context, and how hard it is to achieve this depends on the data set. In some data sets, achieving some level of k could require more alterations, resulting in larger utility reductions, than other data sets.

It can be assumed that higher k-values generally provide more privacy guarantees. The algorithm will therefore be evaluated for a range of k-values. Considering the re-identification risk $1/k$, k-values between 10 and 20 would result in a re-identification risk of 5-10%. This would make it highly unlikely to identify a specific individual from the data. At the same time, these values should not be unrealistically low so that it constraints data utility too much. However, part of the evaluation is to see whether higher k-values can still mostly preserve the utility.

Apart from measures for achieving privacy in the data, there should be some expression of the data utility. The main interest is the comparison of the utility of the transformed data and the original data. To do so, a Random Forest classifier is used to assess whether the utility is preserved in the transformed data. To allow for a comparison, a classifier is trained on both the original data and the transformed data, following the approach of Nguyen et al. (2014).

The performance of a classifier can be expressed through various measures. To provide a complete view of the performance of the classifier, the following performance measures are used:

- Accuracy
- F1
- Recall
- Precision
- ROC score (for binary classification cases)

The difference between the performance of the classifier trained on the original data and the classifier trained on the transformed data then acts as a utility measure. If the performance of the classifier on the transformed data is much worse than the performance of the classifier on the original data, this is an indication that the discretized attributes lack the interaction present in the continuous attributes. To determine whether these differences are acceptable or not, significance testing on the classifiers' accuracies will determine if there is a significant difference between the accuracy obtained with the anonymized data, and the accuracy obtained with the continuous data. If there is a significant difference between the two, this indicates that the utility is not preserved.

Part III

Treatment validation

Chapter 5

Experimental setup

5.1 Setup

The approach to achieve k-anonymity through DET discretization was discussed in Section 4.4. To be able to conduct an experimental evaluation, this approach was implemented in Python. The implementation consists of a tree-based algorithm that iteratively subdivides nodes in two new nodes, based on the approach by Ram and Gray (2011) to determine the best local splits. Applying this to a data set results in a list of bins. These bins describe the ranges of each dimension, which are defined by the training instances that belong to this bin. This list of bins is used to discretize a (sub)set of instances. The measures discussed in Section 4.5 are then evaluated for different k-values and amounts of training instances.

The aim of the experiments is to find an answer to subquestions **SQ5**, **SQ6**, and **SQ7** from Section 2.1. **SQ5** is concerned with the levels of privacy and utility that are achieved from the output of the algorithm. Subsequently, **SQ6** is concerned with the trade-off between privacy and utility, and addresses the effect of higher and lower privacy levels on the level of utility. The aim of this is ultimately to find a point where the levels of privacy and utility can both be considered satisfactory. Lastly, **SQ7** is concerned with finding the minimal amount of instances that are needed to still provide an accurate and privacy preserving summary of the data.

5.2 Measures

During the experimental evaluation, both the level of privacy and utility are expressed through some measures. The level of privacy is measured through the value of k in accordance with the k-anonymity model. This k-value serves as an input parameter as well. The approach is evaluated for a range of k, from $k = 5$ up until $k = 100$, with steps of 5. This range should include reasonable values for k. Evaluating over this range also provides insights into the effect of increasing k regarding the resulting data utility.

Since it is infeasible to express the difference between two multivariate distributions, one continuous and one discrete, the utility of the output of the algorithm is expressed in terms of its classifier performance. A classifier is trained on both the original and the transformed data. This classifier therefore acts as a goodness of fit regarding the discretized data and the preservation of the properties of the original data, and thereby as an indicator for utility.

Considering the amount of instances that are needed to provide an accurate model of the data, the another measure is the bucket size, expressed as the proportion of the set of *training instances*. This is evaluated by creating the DET on different subsets of the training set, and evaluating the performance on the same test set. The proportions of training set sizes that are evaluated range from 10% to 90%, with steps of 10. This is in addition to the evaluation where the complete data is used and divided into training and test set.

Lastly, the runtime is reported for each data set. This provides an indication of the complexity of creating the discretization structure on the training instances. Table 5.1 provides an overview of the measures that are used during the evaluation.

Table 5.1: Evaluation measures

Measure	**Metric**
Privacy level	k
Utility	Classifier performance: - Accuracy - F1 - Recall - Precision - ROC
Minimal bucket size	Proportion of the training set
Computational complexity	Runtime in seconds

5.3 Data

In order to validate the performance and applicability of the use of DET discretization for anonymization, various data sets are used for the evaluation of the approach. All data sets used for the evaluation contain class label information. This is the case since a Random Forest classifier is used to assess the goodness of fit of the transformation process as part of the evaluation. The class label information is only used by the classifier during the evaluation, and entirely disregarded by the DET discretization technique. When it comes to evaluating discretization or machine learning algorithms, it is common to test the algorithm on data sets from the UCI Machine Learning Repository (Dheeru & Karra Taniskidou, 2017). This conveniently allows for the comparison of classifiers or discretization techniques.

In addition to the evaluation on UCI data sets, the approach is also evaluated on synthetically generated data. This can be used to amplify certain characteristics of the data, and to specify shapes and dimensions.

5.3.1 Real-world data

The discretization for the sake of anonymity approach is tested on some common UCI repository data sets. Although anonymization is usually performed on sensitive and personal information, the discretization algorithm can be used on any data set containing continuous attributes, and the levels of privacy and utility can be measured. From the UCI repository, four well known data sets containing only continuous attributes are used. These data sets are described in Table 5.2.

Table 5.2: Real-world data sets

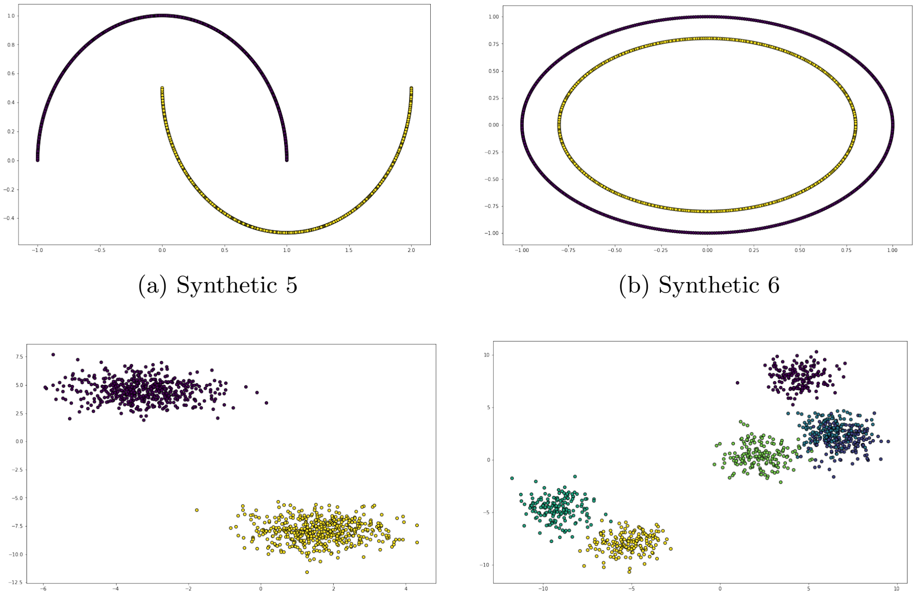
Data set	**Instances**	**Attributes**	**Classes**
breast	682	9	2
glass	213	9	6
iris	150	4	3
wine	178	13	3

5.3.2 Synthetic data

A distinction can be made between three types of generated synthetic data sets for the evaluation of the anonymization approach. At the core of discretization through DETs, the objective is to model and partition the data through density estimation. This can be considered as an unsupervised discretization approach, so possible class labels are not considered when partitioning the data. This could mean that partitions are being made in such a way that the potential interactions between attributes contributing to the class information are lost. To validate whether the discretization algorithm is able to distinguish 'meaningless' attributes, the *iris* set is extended with additional attributes, with values randomly drawn from a Gaussian distribution. Three variations are evaluated, with one, five, and ten additional attributes respectively.

The discretization algorithm is also validated on completely generated synthetic data, created with the *datasets* module from *scikit-learn*. Eight synthetic data sets are used, with varying numbers of instances and attributes. The first four sets are constructed with *scikit-learn*'s *make_classification* utility from the *datasets* module. These sets pose a typical classification problem with interdependent attributes. The first two sets contain the same number of instances, but differ in the number of attributes. The other two sets contain the same number of attributes, while the number of instances differ.

The remaining four synthetic data sets are similar in the amount of instances and attributes. However, these sets differ in the shape of the clusters that are placed in the data, as illustrated by Figure 5.1. Synthetic data set 5 and 6 are generated with *scikit-learn*'s utilities *make_moons* and *make_circles*, and are illustrated by Figure 5.1a and 5.1b respectively. Synthetic sets 7 and 8 are illustrated by Figure 5.1c and 5.1d, and are comprised of a number of distinctable clusters that are placed around a center. Table 5.3 provides a description of the generated data sets and their dimensions. All eight data sets is a binary classification problem, with the exception of Synthetic 8, which has six distinct class labels.



(a) Synthetic 5

(b) Synthetic 6

(c) Synthetic 7

(d) Synthetic 8

Figure 5.1: Synthetic data sets 5-8

Table 5.3: Synthetic data sets

Data set	**Instances**	**Attributes**	**sklearn type**
Synthetic 1	1000	5	*make_classification*
Synthetic 2	1000	10	*make_classification*
Synthetic 3	10000	2	*make_classification*
Synthetic 4	20000	2	*make_classification*
Synthetic 5	1000	2	*make_moons*
Synthetic 6	1000	2	*make_circles*
Synthetic 7	1000	2	*make_blobs*
Synthetic 8	1000	2	*make_blobs*

5.4 Experimental procedure

The experimental evaluation involve applying the anonymization approach to different data sets, both real-world and synthetic. Table 5.4 provides an overview of all data sets that are part of the evaluation. Depending on the size of the data set, two approaches are used to evaluate the goodness of fit of the anonymization process. With a larger number of instances, the data set is randomly split in two parts: 70% training instances and 30% test instances. The training instances are anonymized through DET discretization. The transformed set is then used to train a Random Forest classifier. The remaining 30% is then discretized by assigning them to their respective bins according to the DET created on the training set. The performance of the classifier is then expressed in terms of its performance on the test set.

Table 5.4: Data sets

Data set	**Instances**	**Attributes**	**Classes**	**Evaluation**
breast	682	9	2	10-fold CV
glass	213	9	6	10-fold CV
iris	150	4	3	10-fold CV
wine	178	13	3	10-fold CV
iris + 1	150	5	3	10-fold CV
iris + 5	150	9	3	10-fold CV
iris + 10	150	14	3	10-fold CV
Synthetic 1	1000	5	2	Train/test
Synthetic 2	1000	10	2	Train/test
Synthetic 3	10000	2	2	Train/test
Synthetic 4	20000	2	2	Train/test
Synthetic 5	1000	2	2	Train/test
Synthetic 6	1000	2	2	Train/test
Synthetic 7	1000	2	2	Train/test
Synthetic 8	1000	2	6	Train/test

For smaller sets, 10-fold cross validation is used. In this case, the data is randomly split in ten folds. For ten iterations, nine folds are used as training set, and the remaining as test set. Every fold is used as test set exactly once. For every iteration, the training set is anonymized through DET discretization, and then used as training set for the Random Forest classifiers. The classifier performance is evaluated on the test fold, and the performance measures are averaged over all iterations.

To compare the results, four other approaches are used and evaluated through their classification performance. In the first place, a classifier is trained and tested on the original, continuous attributes, without using discretization. These results serve as the *baseline* classification performance results. In addition, the *equal-width* and *equal-frequency* discretization methods are used. These are unsupervised, univariate approaches, so all attributes are discretized individually. Both methods require the number of bins as input parameter. Similar to the evaluation of anonymization through DET discretization, both methods are evaluated over a range of this parameter. Finally, the results are compared with the supervised MDLP discretization approach by Fayyad and Irani (1993), which does not require an input parameter.

Studying the effect of the number of instances that are needed to obtain an accurate summary requires an additional approach. When using the train-test evaluation approach, instead of using the full training set to construct the DET, only a proportion is used. When using cross-validation, a proportion of the instances included in the training folds is used. This proportion is evaluated in the range of 0.1 to 0.9, with steps of 0.1, while the test set remains the same. For each proportion, the same range of k-values is used for the evaluation as with the main evaluation, or a subset of it. This results in the classification accuracy for every combination of k-value and training set proportion.

Chapter 6

Results

The experimental evaluation yields many results, and these are presented in this chapter. When it comes to the performance of the anonymization approach, the results regarding utility and privacy are of particular interest. For every data set that is part of the evaluation, the same output is generated. The results for each set can roughly be divided in three categories:

1. Diagnostic results. These are mainly focused on the anonymization approach and the performance measures of the classifier indicating the utility. The classification performance results are illustrated in this chapter. Appendix A contains the corresponding tables with the exact values, and the performance measures' standard deviations when cross-validation is used for the evaluation.

2. Comparative results. These results provide a comparison between DET discretization for anonymization, and the discretization techniques equal-width, equal-frequency, and MDLP. The comparison of the classification accuracy for these different techniques are illustrated in this chapter. Appendix A contains a similar comparison, for F_1 and the ROC AUC score. The runtime of the techniques is also compared in Appendix A.

3. Results concerning the training set proportion. These results are included in this chapter, and are illustrated as a colored heatmap containing the classification accuracy for every evaluated combination of k and training set proportion.

6.1 Real-world data

6.1.1 Breast

The first data set is the *breast* set from the UCI repository. The classification performance measures are shown in Figure 6.1, along with the baseline accuracy of the classifier trained on the original data. This effectively illustreates how much of a difference there is between a classifier trained on the original data, and one on the anonymized data. Additional results are shown in Appendix A.1. The results show that with a k-value of 15, the highest cross-validated classification accuracy of 0.95 is achieved, with a baseline accuracy of 0.96. In addition, Figure 6.1 shows that the classification performance is relatively stable up until a k-value of 95.

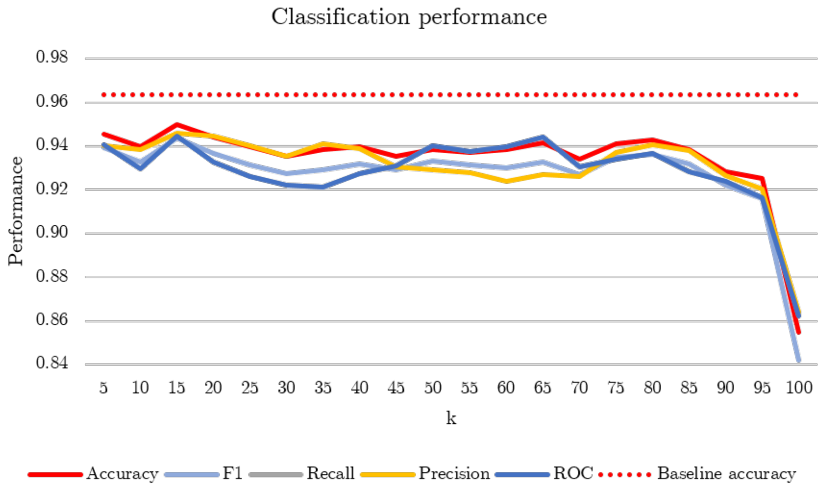


Figure 6.1: Classification performance breast

The performance of the DET anonymization approach is compared with other discretization techniques as well. Figure 6.2 shows the classification accuracy for these techniques. For the DET approach, the input parameter k on the x-axis ranges from 5 to 100. The input parameter equal-width and equal-frequency, the number of bins, is inverted on this axis, and ranges from 100 to 5. This is the case since low k-values with the DET approach result in a higher number of bins, while high k-values result in less bins. Inverting the axis for equal-width and equal-frequency makes sure that with these three techniques, a high number of bins is on the left side of the graph, and a lower number on the right. MDLP and no discretization do not require an input parameter, so these approaches resulted in one accuracy measure instead of one for each parameter value. This comparison shows that EW could achieve 0.97 accuracy at 80 bins, EF 0.97 at 60 bins, while MDLP also resulted in 0.97 classification accuracy.

Although DET performs slightly worse with 0.95, it is relatively close to the other approaches up until a k-value of 95.

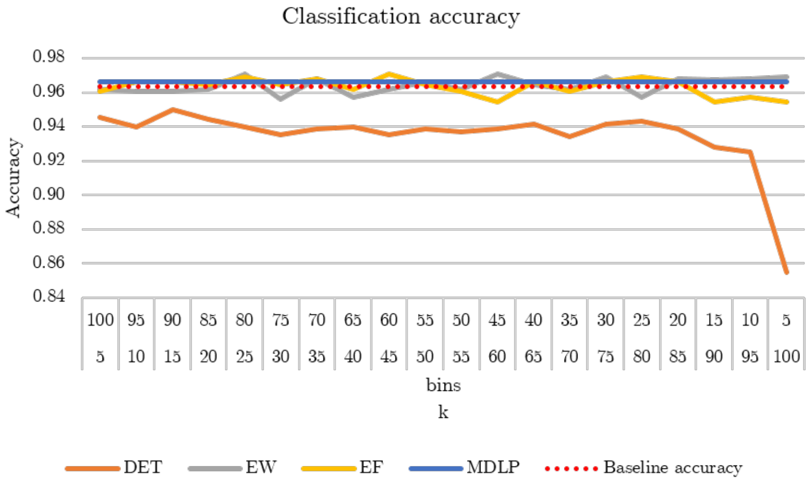
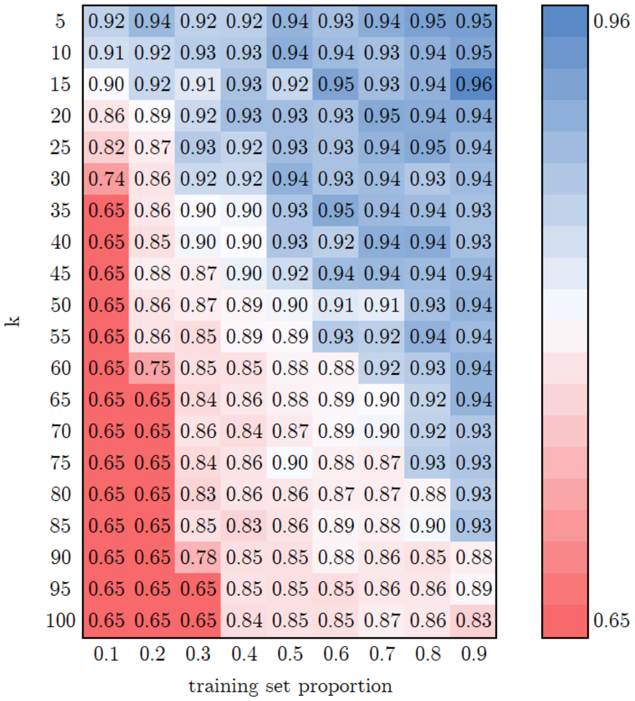


Figure 6.2: Classification accuracy comparison breast

The DET anonymization approach is evaluated for different proportions of the training set as well. Figure 6.3 shows the classification accuracy for the various combinations of k and the training set proportion. The cells highlighted in blue indicate relatively high accuracy, while the red highlighted cells contain relatively low measures. The best classification accuracy results are found in the upper right corner, with low k-values and a larger training set. However, it is still possible to obtain high accuracy with less data and reasonably high k-values. For example, with only 50% of the training instances and $k = 30$, an accuracy of 0.94 is achieved.



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.92	0.94	0.92	0.92	0.94	0.93	0.94	0.95	0.95
10	0.91	0.92	0.93	0.93	0.94	0.94	0.93	0.94	0.95
15	0.90	0.92	0.91	0.93	0.92	0.95	0.93	0.94	0.96
20	0.86	0.89	0.92	0.93	0.93	0.93	0.95	0.94	0.94
25	0.82	0.87	0.93	0.92	0.93	0.93	0.94	0.95	0.94
30	0.74	0.86	0.92	0.92	0.94	0.93	0.94	0.93	0.94
35	0.65	0.86	0.90	0.90	0.93	0.95	0.94	0.94	0.93
40	0.65	0.85	0.90	0.90	0.93	0.92	0.94	0.94	0.93
45	0.65	0.88	0.87	0.90	0.92	0.94	0.94	0.94	0.94
50	0.65	0.86	0.87	0.89	0.90	0.91	0.91	0.93	0.94
55	0.65	0.86	0.85	0.89	0.89	0.93	0.92	0.94	0.94
60	0.65	0.75	0.85	0.85	0.88	0.88	0.92	0.93	0.94
65	0.65	0.65	0.84	0.86	0.88	0.89	0.90	0.92	0.94
70	0.65	0.65	0.86	0.84	0.87	0.89	0.90	0.92	0.93
75	0.65	0.65	0.84	0.86	0.90	0.88	0.87	0.93	0.93
80	0.65	0.65	0.83	0.86	0.86	0.87	0.87	0.88	0.93
85	0.65	0.65	0.85	0.83	0.86	0.89	0.88	0.90	0.93
90	0.65	0.65	0.78	0.85	0.85	0.88	0.86	0.85	0.88
95	0.65	0.65	0.65	0.85	0.85	0.85	0.86	0.86	0.89
100	0.65	0.65	0.65	0.84	0.85	0.85	0.87	0.86	0.83

Figure 6.3: Classification accuracy heatmap breast

6.1.2 Glass

The classification performance results for the *glass* set are shown in Figure 6.4. This shows that the highest classification accuracy that is achieved is 0.58, with $k = 5$. The accuracy without discretization is 0.65. Compared to the *breast* set, there is a larger difference in classification performance between the anonymized and the original data. In addition, the performance quickly declines beyond low k-values.

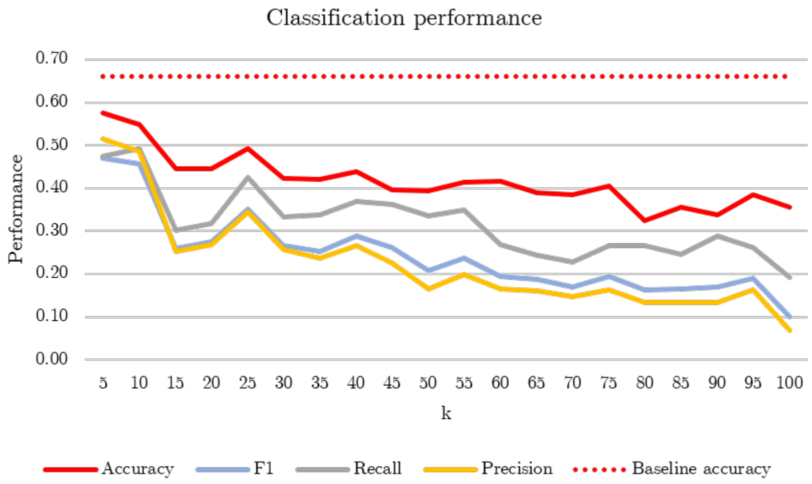


Figure 6.4: Classification performance glass

A comparison with the other discretization methods is illustrated by Figure 6.5. The other techniques perform better than without discretization. EW achieves 0.77 accuracy with 45 bins, EF obtains 0.76 with 70 bins, while MDLP results in an accuracy of 0.75.

Lastly, Figure 6.6 shows a heatmap of the classification accuracy for the combinations of training set proportion and k. It shows a general degradation in performance when increasing k or decreasing the amount of training instances, although an accuracy of 0.57 is achieved with only 50% of the training instances and $k = 5$. It is clear that the DET approach has some difficulties with the *glass* data set when it comes to classification performance. This could be caused by the fact that this data set is smaller than the *breast* set, is a relatively difficult classification problem, and has six distinct class labels. Additional results are included in Appendix A.2.

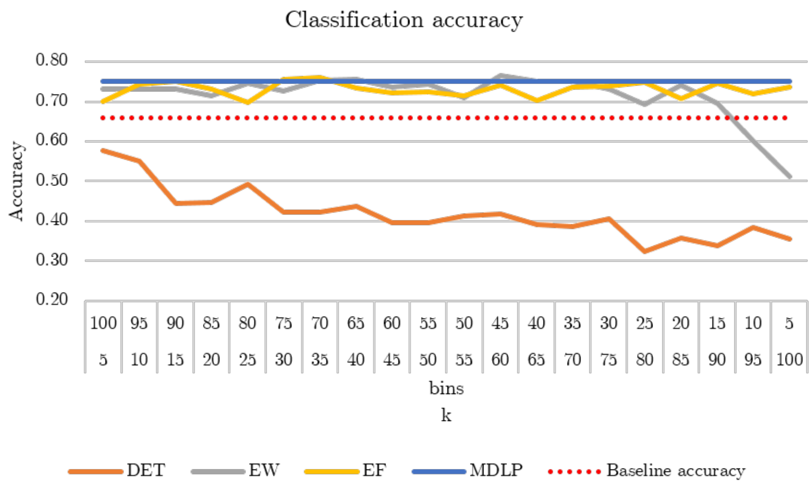
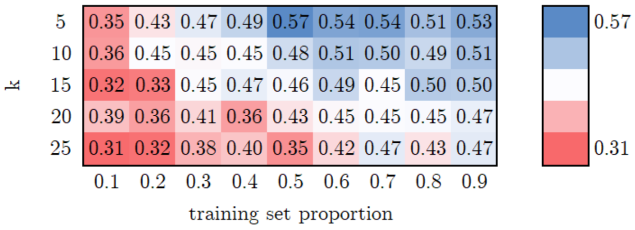


Figure 6.5: Classification accuracy comparison glass



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.35	0.43	0.47	0.49	0.57	0.54	0.54	0.51	0.53
10	0.36	0.45	0.45	0.45	0.48	0.51	0.50	0.49	0.51
15	0.32	0.33	0.45	0.47	0.46	0.49	0.45	0.50	0.50
20	0.39	0.36	0.41	0.36	0.43	0.45	0.45	0.45	0.47
25	0.31	0.32	0.38	0.40	0.35	0.42	0.47	0.43	0.47

Figure 6.6: Classification accuracy heatmap glass

6.1.3 Iris

The classification performance results concerning the *iris* set are shown in Figure 6.7. The highest classification accuracy that is achieved is 0.95, at $k = 5$. The accuracy that is achieved without discretization is 0.96. The results show decreasing performance around the k-values 45 and 70. This corresponds with changes in the number of bins that are created by the algorithm, which are significant with this particular data set. The *iris* data set is small, and has three class labels. Up until a partitioning of the data in three parts, the class labels are relatively straightforward to predict. However, partitioning the data in two parts, which happens around $k = 45$, doesn't give the classifier enough information to predict the three class labels. The same is naturally true when there is just one bin, which is the case from $k = 70$ onwards. Figure 6.7 shows how the classification performance increases around $k = 30$ after an initial decline, and is still high at $k = 40$. Table A.3 shows that at this k-value, exactly three bins are created, while lower k-values naturally result in a slightly larger number of bins. These results indicate that in terms of utility, discretizing into three bins are actually preferable over more than three bins. In terms of privacy, increasing k does not *always* result in a decline in utility.

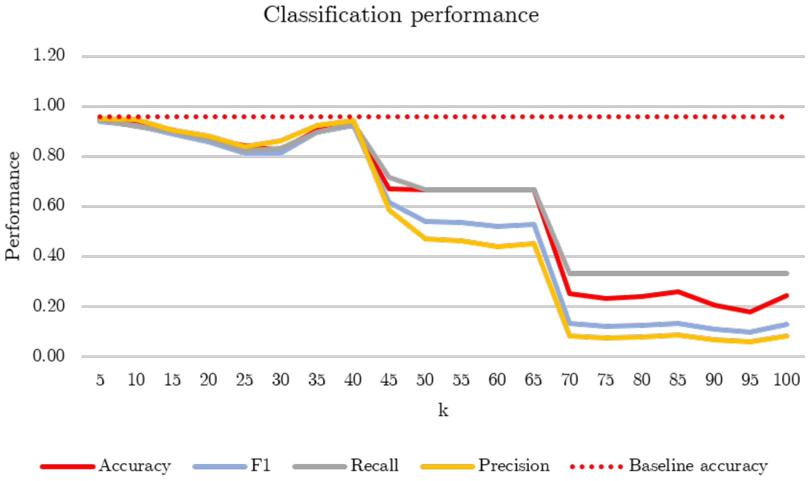


Figure 6.7: Classification performance iris

A comparison in performance with the other discretization techniques is provided by Figure 6.8. With EW, an accuracy of 0.96 is achieved with 55 bins, EF results in 0.97 with 80 bins, while MDLP obtains an accuracy of 0.96. This shows that all methods can obtain similar classification performance.

Figure 6.9 shows the accuracy heatmap for the *iris* set. Again, the highest accuracy is obtained in the top right part of the heatmap, with low k-values and a larger number of training instances. However, relatively high accuracy can be

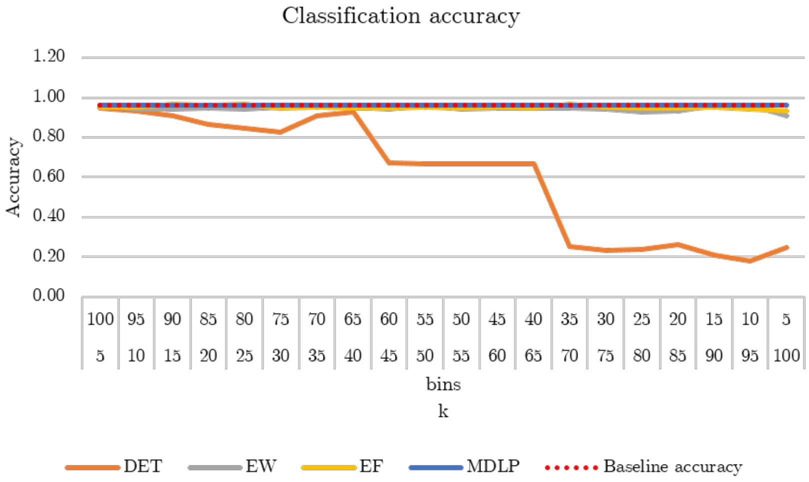
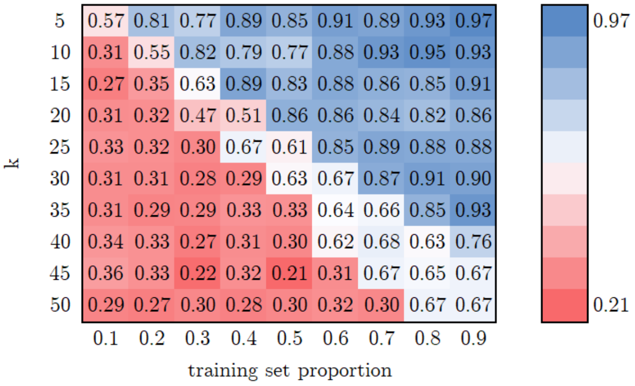


Figure 6.8: Classification accuracy comparison iris

obtained with higher *k*-values and less training instances. As long as the data is partitioned in three parts or more, the utility of the *iris* set is still mostly preserved. Appendix A.3 contains additional results for this data set.



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.57	0.81	0.77	0.89	0.85	0.91	0.89	0.93	0.97
10	0.31	0.55	0.82	0.79	0.77	0.88	0.93	0.95	0.93
15	0.27	0.35	0.63	0.89	0.83	0.88	0.86	0.85	0.91
20	0.31	0.32	0.47	0.51	0.86	0.86	0.84	0.82	0.86
25	0.33	0.32	0.30	0.67	0.61	0.85	0.89	0.88	0.88
30	0.31	0.31	0.28	0.29	0.63	0.67	0.87	0.91	0.90
35	0.31	0.29	0.29	0.33	0.33	0.64	0.66	0.85	0.93
40	0.34	0.33	0.27	0.31	0.30	0.62	0.68	0.63	0.76
45	0.36	0.33	0.22	0.32	0.21	0.31	0.67	0.65	0.67
50	0.29	0.27	0.30	0.28	0.30	0.32	0.30	0.67	0.67

Figure 6.9: Classification accuracy heatmap iris

6.1.4 Wine

The results for the *wine* data set are shown in Figure 6.10. Similar to the *iris* set, *wine* has a small number of instances and distinct class labels, but a larger number of attributes. The highest classification accuracy that is achieved at $k = 5$ is 0.90, with a baseline accuracy of 0.97. With this data, the performance drops noticeably at $k = 55$ and $k = 85$. As Table A.4 of Appendix A.4 shows, at these values, the discretization results in two bins and one bin respectively, which makes it impossible for the classifier to predict the three distinct class labels.

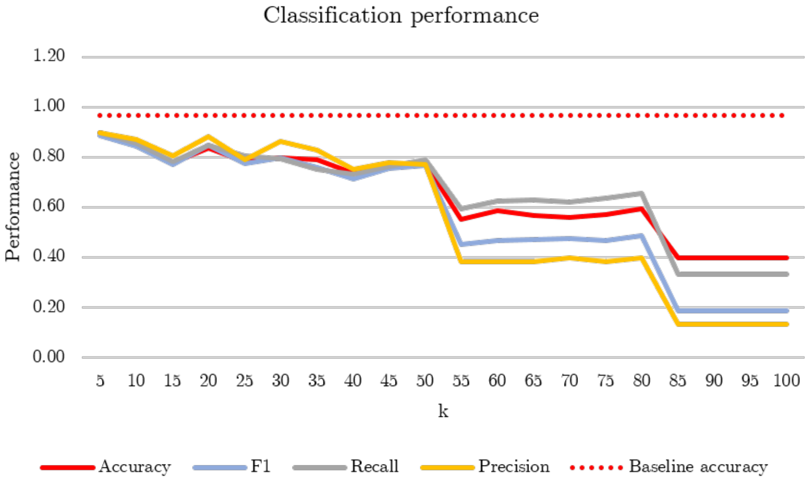


Figure 6.10: Classification performance wine

Figure 6.11 shows a comparison of the classification accuracy between the different discretization methods. EW, EF, and MDLP are all able to achieve 0.98 accuracy.

The heatmap with classification accuracy for DET discretization is provided by Figure 6.12. The higher accuracy values in the top right are all examples where at least three bins are created. The clear drop in performance, illustrated by the light blue cells, occurs when the data is partitioned in less than three parts.

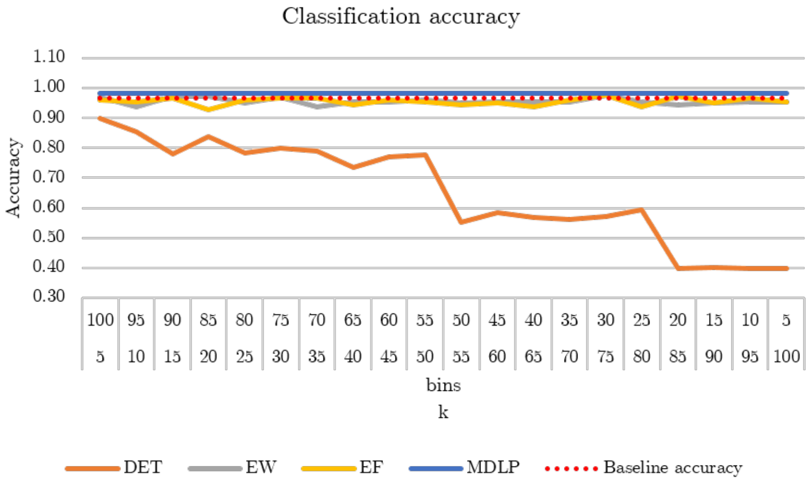
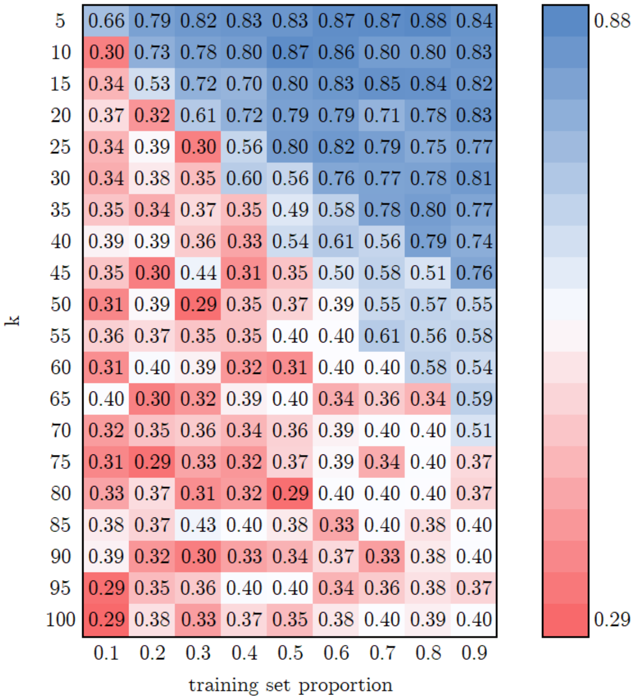


Figure 6.11: Classification accuracy comparison wine



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.66	0.79	0.82	0.83	0.83	0.87	0.87	0.88	0.84
10	0.30	0.73	0.78	0.80	0.87	0.86	0.80	0.80	0.83
15	0.34	0.53	0.72	0.70	0.80	0.83	0.85	0.84	0.82
20	0.37	0.32	0.61	0.72	0.79	0.79	0.71	0.78	0.83
25	0.34	0.39	0.30	0.56	0.80	0.82	0.79	0.75	0.77
30	0.34	0.38	0.35	0.60	0.56	0.76	0.77	0.78	0.81
35	0.35	0.34	0.37	0.35	0.49	0.58	0.78	0.80	0.77
40	0.39	0.39	0.36	0.33	0.54	0.61	0.56	0.79	0.74
45	0.35	0.30	0.44	0.31	0.35	0.50	0.58	0.51	0.76
50	0.31	0.39	0.29	0.35	0.37	0.39	0.55	0.57	0.55
55	0.36	0.37	0.35	0.35	0.40	0.40	0.61	0.56	0.58
60	0.31	0.40	0.39	0.32	0.31	0.40	0.40	0.58	0.54
65	0.40	0.30	0.32	0.39	0.40	0.34	0.36	0.34	0.59
70	0.32	0.35	0.36	0.34	0.36	0.39	0.40	0.40	0.51
75	0.31	0.29	0.33	0.32	0.37	0.39	0.34	0.40	0.37
80	0.33	0.37	0.31	0.32	0.29	0.40	0.40	0.40	0.37
85	0.38	0.37	0.43	0.40	0.38	0.33	0.40	0.38	0.40
90	0.39	0.32	0.30	0.33	0.34	0.37	0.33	0.38	0.40
95	0.29	0.35	0.36	0.40	0.40	0.34	0.36	0.38	0.37
100	0.29	0.38	0.33	0.37	0.35	0.38	0.40	0.39	0.40

Figure 6.12: Classification accuracy heatmap wine

6.2 Synthetic data

As discussed in Section 5.3.2, there are three types of synthetic data to evaluate the anonymization algorithm. The first type involves three variations of the *iris* set, in which one, five and ten additional, random attributes are added to the data. These results are discussed first, followed by the results on the eight full synthetic data sets.

6.2.1 Iris + 1

The first variation of *iris* is that with one additional attribute. The classification performance results are shown in Figure 6.13. These show that the highest cross-validated accuracy of 0.94 is achieved with $k = 40$, while the baseline accuracy is 0.97. This is slightly less than the accuracy of 0.95 that is achieved with the original *iris* set. In general, Figure 6.13 shows a similar trend to the original set, with a clear performance decrease around $k = 45$ and $k = 70$, caused by the discretization in less than three bins. There is an increase towards $k = 40$ after an initial decline as well. Appendix A.5 contains additional results for this set.

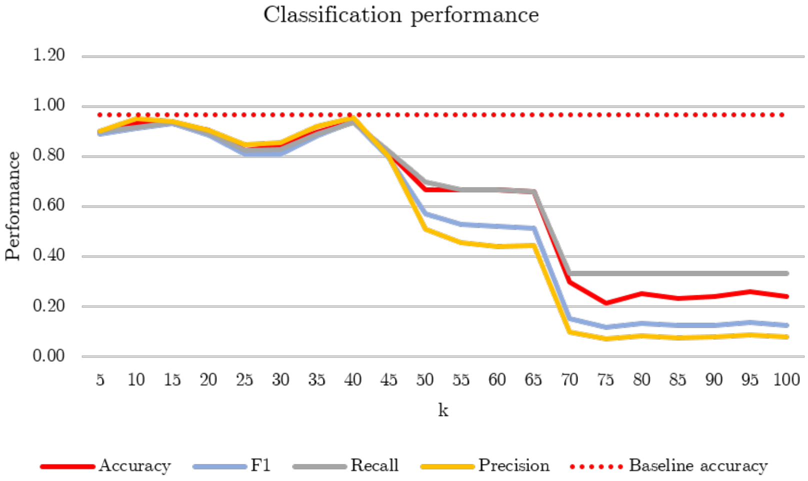


Figure 6.13: Classification performance iris + 1

Figure 6.14 illustrates a comparison with the other discretization methods. Both EF and EW result in an accuracy of 0.97, while MDLP obtains an accuracy of 0.94.

The accuracy results for the various combinations of k and training set proportion is shown by Figure 6.15. When three bins or more are created, the accuracy is high, indicated by the blue top right area. With less than two bins, the performance naturally drops.

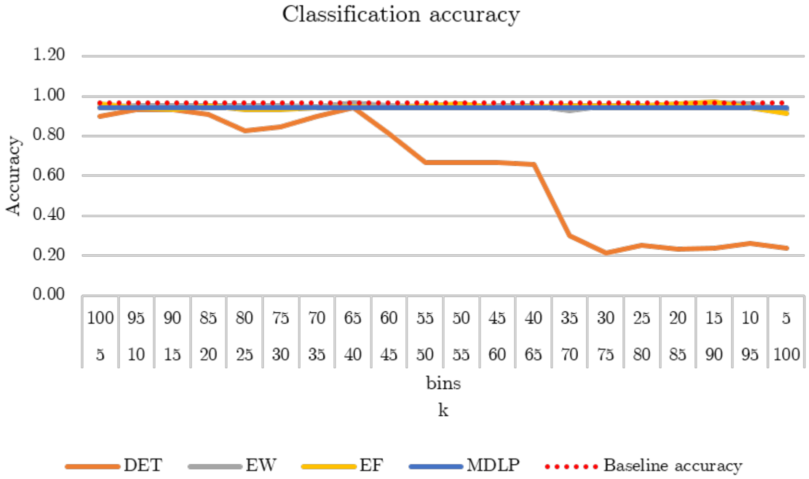
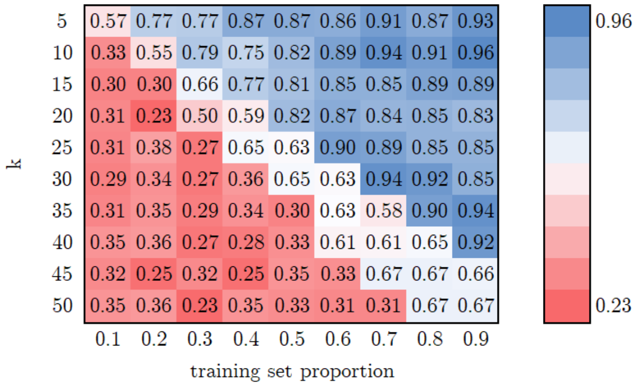


Figure 6.14: Classification accuracy comparison iris + 1



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.57	0.77	0.77	0.87	0.87	0.86	0.91	0.87	0.93
10	0.33	0.55	0.79	0.75	0.82	0.89	0.94	0.91	0.96
15	0.30	0.30	0.66	0.77	0.81	0.85	0.85	0.89	0.89
20	0.31	0.23	0.50	0.59	0.82	0.87	0.84	0.85	0.83
25	0.31	0.38	0.27	0.65	0.63	0.90	0.89	0.85	0.85
30	0.29	0.34	0.27	0.36	0.65	0.63	0.94	0.92	0.85
35	0.31	0.35	0.29	0.34	0.30	0.63	0.58	0.90	0.94
40	0.35	0.36	0.27	0.28	0.33	0.61	0.61	0.65	0.92
45	0.32	0.25	0.32	0.25	0.35	0.33	0.67	0.67	0.66
50	0.35	0.36	0.23	0.35	0.33	0.31	0.31	0.67	0.67

Figure 6.15: Classification accuracy heatmap iris + 1

6.2.2 Iris + 5

The second set is the *iris* data with 5 additional attributes. The classification performance is shown by Figure 6.16. The cross-validated accuracy of 0.92 is achieved at $k = 40$. The baseline cross-validated accuracy is 0.93. Again, the results are very similar to the original *iris* set and iris + 1, but slightly lower. Additional results are included in Appendix A.6.

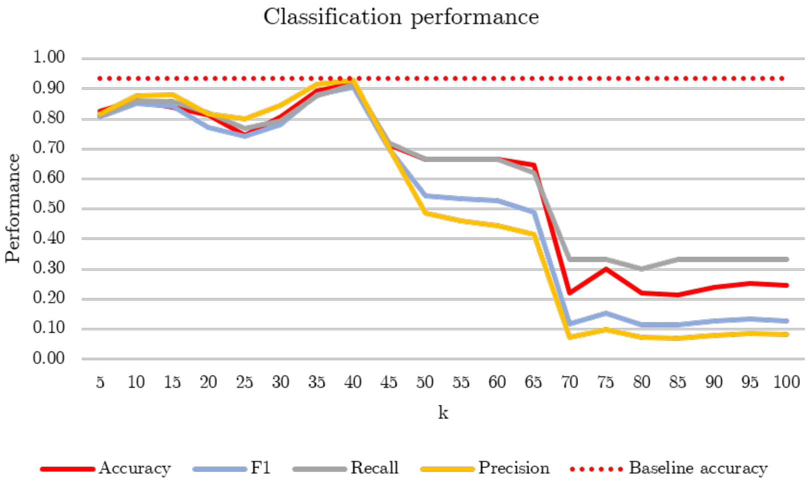


Figure 6.16: Classification performance iris + 5

The comparison with other discretization methods in terms of classification accuracy is shown in Figure 6.17. Their best accuracy results are very similar. EW and EF both achieve an accuracy of 0.96, while MDLP is able to achieve 0.93 accuracy.

Figure 6.18 shows the classification accuracy heatmap. Although the same pattern applies as with the other *iris* variations, it is less clear. This is illustrated by the fact that the top right cell, with the lowest k-value and the largest number of training instances, the classification accuracy is far from the highest. Instead, better results are achieved at higher k-values.

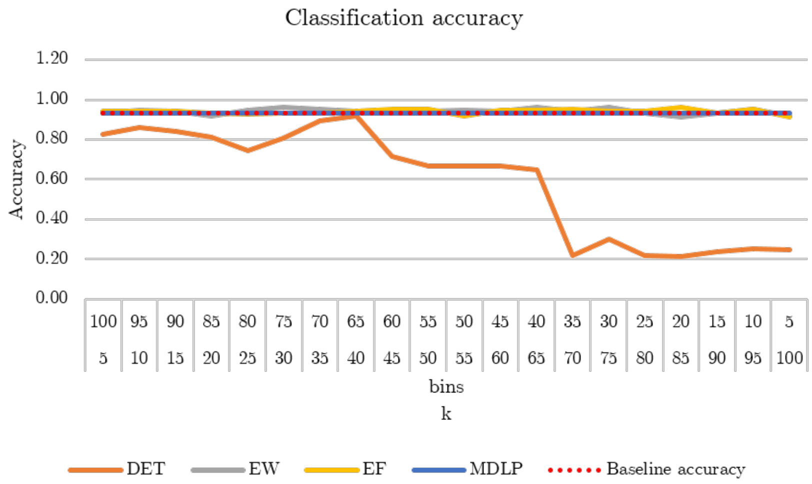
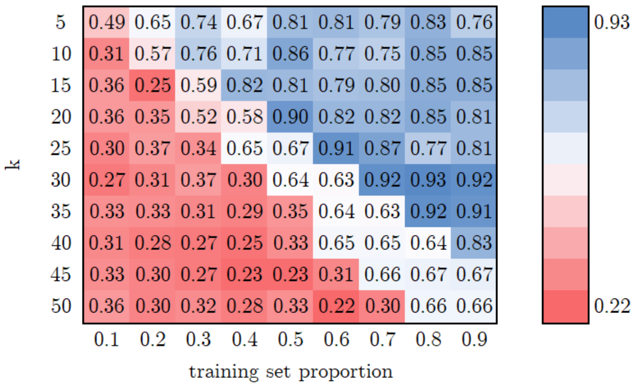


Figure 6.17: Classification accuracy comparison iris + 5



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.49	0.65	0.74	0.67	0.81	0.81	0.79	0.83	0.76
10	0.31	0.57	0.76	0.71	0.86	0.77	0.75	0.85	0.85
15	0.36	0.25	0.59	0.82	0.81	0.79	0.80	0.85	0.85
20	0.36	0.35	0.52	0.58	0.90	0.82	0.82	0.85	0.81
25	0.30	0.37	0.34	0.65	0.67	0.91	0.87	0.77	0.81
30	0.27	0.31	0.37	0.30	0.64	0.63	0.92	0.93	0.92
35	0.33	0.33	0.31	0.29	0.35	0.64	0.63	0.92	0.91
40	0.31	0.28	0.27	0.25	0.33	0.65	0.65	0.64	0.83
45	0.33	0.30	0.27	0.23	0.23	0.31	0.66	0.67	0.67
50	0.36	0.30	0.32	0.28	0.33	0.22	0.30	0.66	0.66

Figure 6.18: Classification accuracy heatmap iris + 5

6.2.3 Iris + 10

The last *iris* variation has 10 additional attributes. Figure 6.19 shows the classification performance for this data. The k-value of 40 results in the highest cross-validated accuracy of 0.94, while the baseline accuracy is 0.92. The performance results show the same trend as with the other *iris* variations, although the increase around $k = 40$ is even more apparent. In addition, the performance in general is slightly worse than with the previous variations. Additional results are included in Appendix A.7.

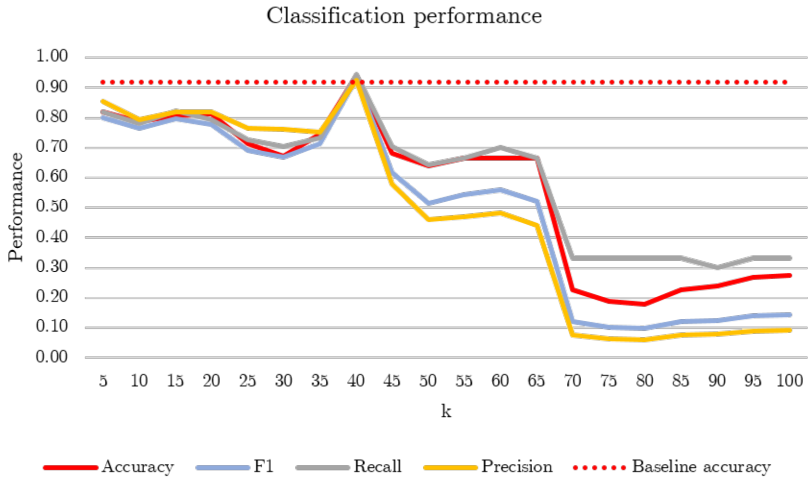


Figure 6.19: Classification performance iris + 10

Figure 6.20 shows a comparison with the other discretization techniques. With EW and EF, an accuracy of 0.97 is achieved, while using MDLP obtains 0.95 accuracy.

The classification accuracy heatmap is shown in Figure 6.21. The pattern is somewhat similar to iris + 5, in that the highest accuracy results are not obtained at the lowest k-values for each training set size. Instead, higher k-values result in a higher classification accuracy, which is also the case with the regular results, where $k = 40$ obtains the best performance.

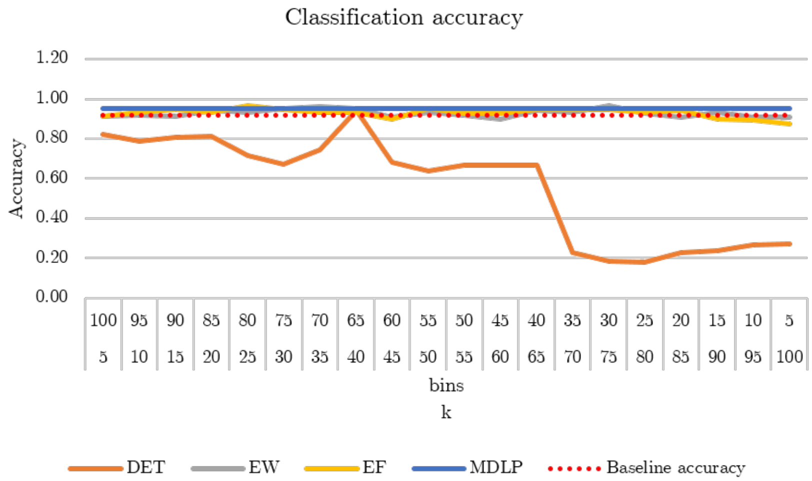
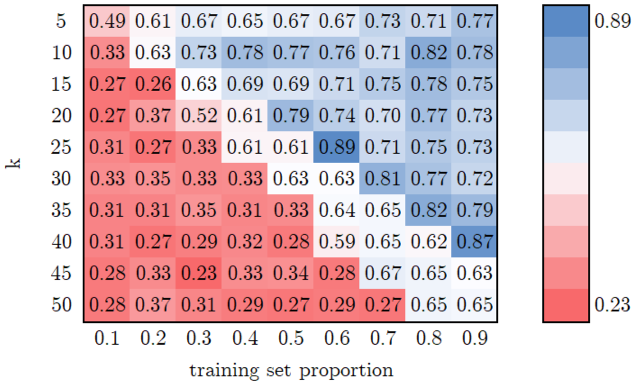


Figure 6.20: Classification accuracy comparison iris + 10



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.49	0.61	0.67	0.65	0.67	0.67	0.73	0.71	0.77
10	0.33	0.63	0.73	0.78	0.77	0.76	0.71	0.82	0.78
15	0.27	0.26	0.63	0.69	0.69	0.71	0.75	0.78	0.75
20	0.27	0.37	0.52	0.61	0.79	0.74	0.70	0.77	0.73
25	0.31	0.27	0.33	0.61	0.61	0.89	0.71	0.75	0.73
30	0.33	0.35	0.33	0.33	0.63	0.63	0.81	0.77	0.72
35	0.31	0.31	0.35	0.31	0.33	0.64	0.65	0.82	0.79
40	0.31	0.27	0.29	0.32	0.28	0.59	0.65	0.62	0.87
45	0.28	0.33	0.23	0.33	0.34	0.28	0.67	0.65	0.63
50	0.28	0.37	0.31	0.29	0.27	0.29	0.27	0.65	0.65

Figure 6.21: Classification accuracy heatmap iris + 10

6.2.4 Synthetic 1

Apart from the extended *iris* data, the DET anonymization approach is evaluated on full synthetic data as well. In this case, the data is split in a training set and a test set, instead of using cross-validation.

The first synthetic set contains 1000 instances and five attributes. The classification performance results that were obtained are shown in Figure 6.22. With k-values of 30 and 40, the highest classification accuracy of 0.77 was obtained, with a baseline accuracy of 0.89. In this case, the performance on the original data is substantially higher than on the anonymized data. Appendix A.8 contains additional results for this data set.

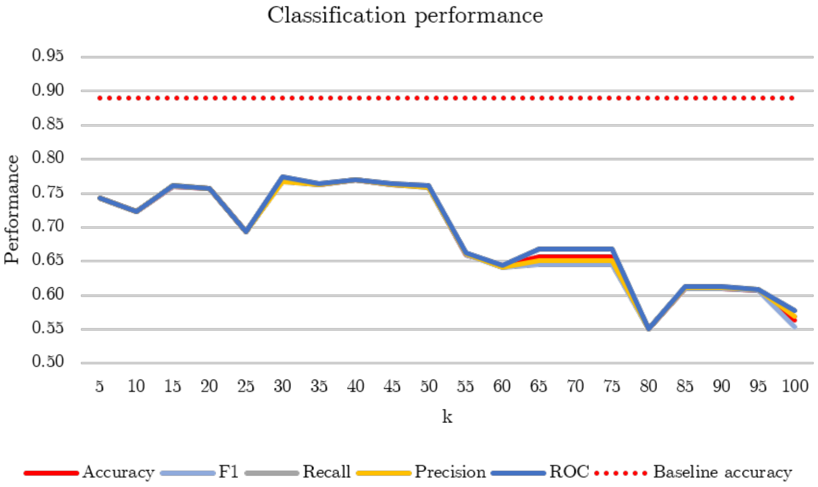


Figure 6.22: Classification performance synthetic 1

Figure 6.23 shows the comparison with other discretization methods. The accuracy results obtained with these methods are all close to the accuracy without discretization. EW results in 0.88 accuracy, EF in 0.89, and MDLP obtains an accuracy of 0.84.

The classification accuracy heatmap is shown in Figure 6.24. The distinction between high and low accuracy values is not as clear as with the real-world data. The highest accuracy values of 0.80 are obtained with relatively higher k-values and a smaller set of training instances. In general however, the bottom left part of the heatmap shows very low performance results, while the top right contains accuracy values around 0.70.

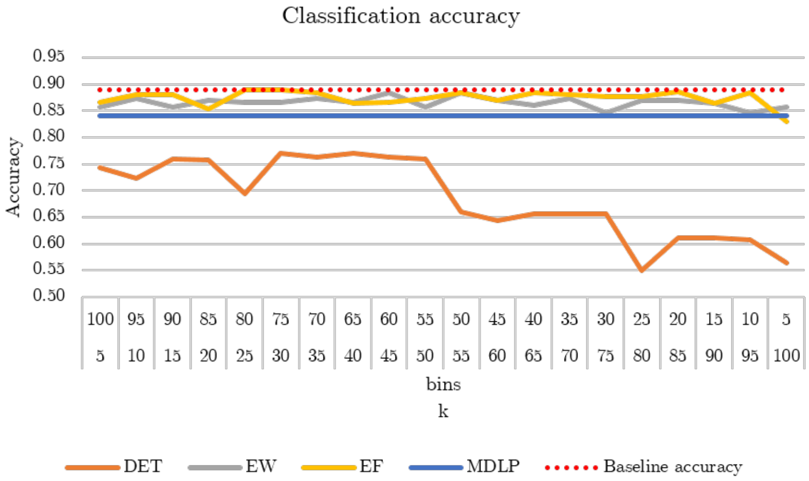
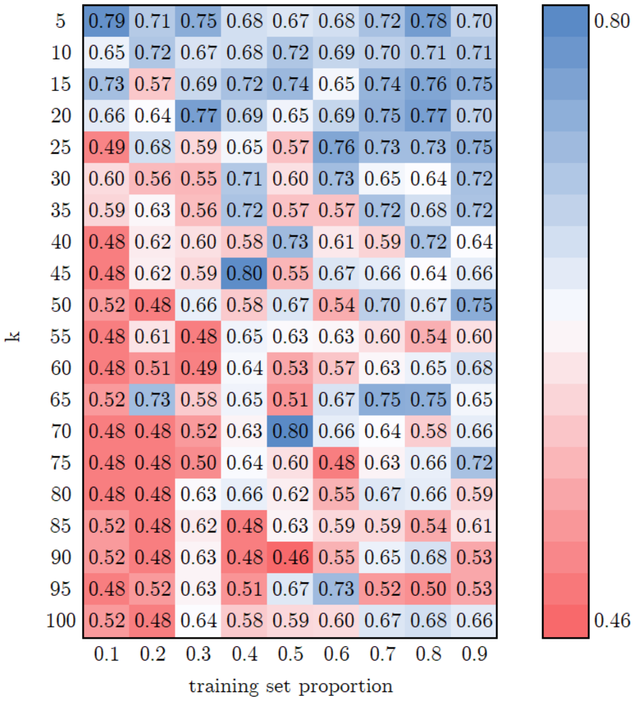


Figure 6.23: Classification accuracy comparison synthetic 1



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.79	0.71	0.75	0.68	0.67	0.68	0.72	0.78	0.70
10	0.65	0.72	0.67	0.68	0.72	0.69	0.70	0.71	0.71
15	0.73	0.57	0.69	0.72	0.74	0.65	0.74	0.76	0.75
20	0.66	0.64	0.77	0.69	0.65	0.69	0.75	0.77	0.70
25	0.49	0.68	0.59	0.65	0.57	0.76	0.73	0.73	0.75
30	0.60	0.56	0.55	0.71	0.60	0.73	0.65	0.64	0.72
35	0.59	0.63	0.56	0.72	0.57	0.57	0.72	0.68	0.72
40	0.48	0.62	0.60	0.58	0.73	0.61	0.59	0.72	0.64
45	0.48	0.62	0.59	0.80	0.55	0.67	0.66	0.64	0.66
50	0.52	0.48	0.66	0.58	0.67	0.54	0.70	0.67	0.75
55	0.48	0.61	0.48	0.65	0.63	0.63	0.60	0.54	0.60
60	0.48	0.51	0.49	0.64	0.53	0.57	0.63	0.65	0.68
65	0.52	0.73	0.58	0.65	0.51	0.67	0.75	0.75	0.65
70	0.48	0.48	0.52	0.63	0.80	0.66	0.64	0.58	0.66
75	0.48	0.48	0.50	0.64	0.60	0.48	0.63	0.66	0.72
80	0.48	0.48	0.63	0.66	0.62	0.55	0.67	0.66	0.59
85	0.52	0.48	0.62	0.48	0.63	0.59	0.59	0.54	0.61
90	0.52	0.48	0.63	0.48	0.46	0.55	0.65	0.68	0.53
95	0.48	0.52	0.63	0.51	0.67	0.73	0.52	0.50	0.53
100	0.52	0.48	0.64	0.58	0.59	0.60	0.67	0.68	0.66

Figure 6.24: Classification accuracy heatmap synthetic 1

6.2.5 Synthetic 2

The second synthetic data set also contains 1000 instances, but ten attributes. The results of the classification performance on the test set are shown in Figure 6.25. The highest classification accuracy that is achieved on the test set is 0.67, corresponding to $k = 10$. Further results are included in Appendix A.9. In this case, the performance difference between the original data and the anonymized data is larger than with the previous data sets.

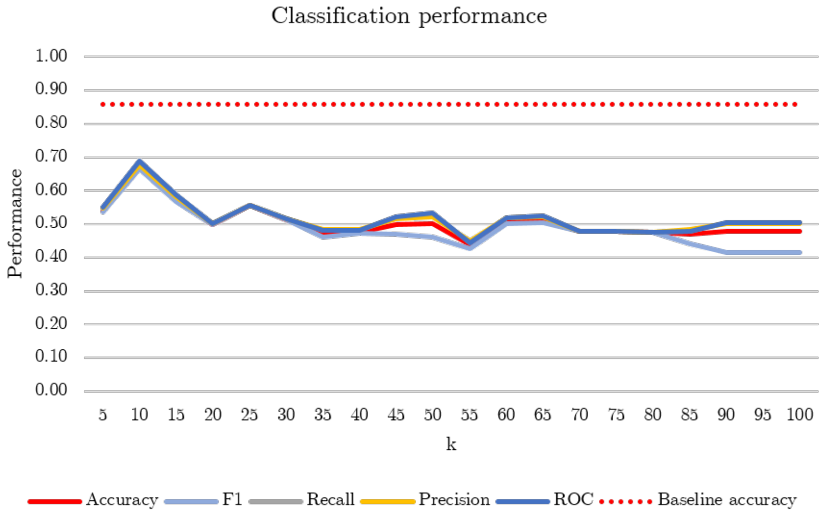


Figure 6.25: Classification performance synthetic 2

A comparison in classification accuracy with other discretization methods is shown in Figure 6.26. With both EW and EF, an accuracy of 0.88 can be obtained, while MDLP results in an accuracy of 0.80.

Figure 6.27 shows the classification accuracy heatmap for this data set. Similar to the first synthetic data set, there is no clear pattern to detect regarding high and low accuracy values. In addition, with only 30% of the training instances, an accuracy of 0.78 is achieved, which is much higher than the accuracy of 0.67 that was originally obtained with the complete training set.

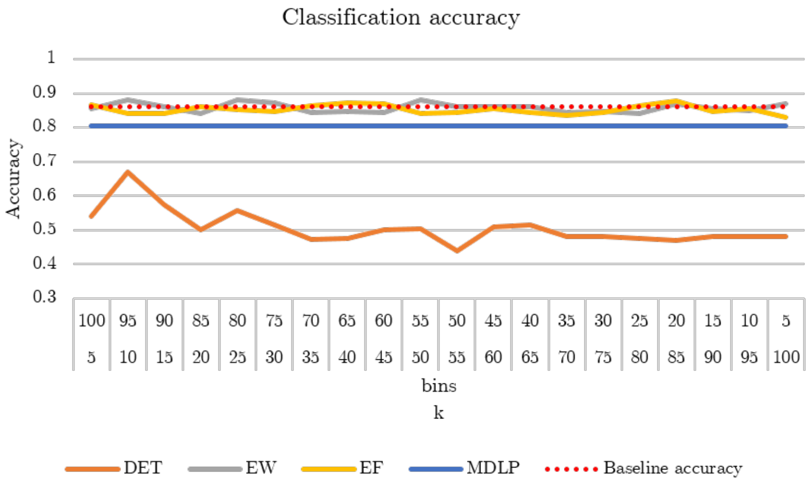
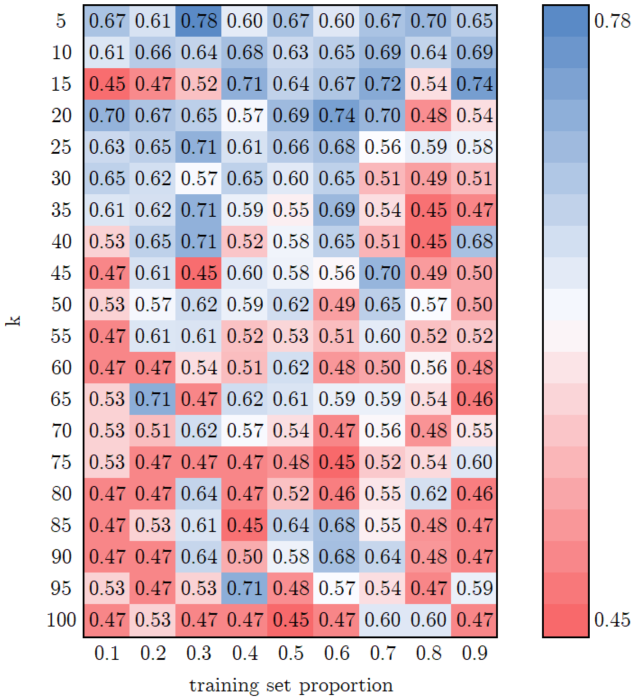


Figure 6.26: Classification accuracy comparison synthetic 2



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.67	0.61	0.78	0.60	0.67	0.60	0.67	0.70	0.65
10	0.61	0.66	0.64	0.68	0.63	0.65	0.69	0.64	0.69
15	0.45	0.47	0.52	0.71	0.64	0.67	0.72	0.54	0.74
20	0.70	0.67	0.65	0.57	0.69	0.74	0.70	0.48	0.54
25	0.63	0.65	0.71	0.61	0.66	0.68	0.56	0.59	0.58
30	0.65	0.62	0.57	0.65	0.60	0.65	0.51	0.49	0.51
35	0.61	0.62	0.71	0.59	0.55	0.69	0.54	0.45	0.47
40	0.53	0.65	0.71	0.52	0.58	0.65	0.51	0.45	0.68
45	0.47	0.61	0.45	0.60	0.58	0.56	0.70	0.49	0.50
50	0.53	0.57	0.62	0.59	0.62	0.49	0.65	0.57	0.50
55	0.47	0.61	0.61	0.52	0.53	0.51	0.60	0.52	0.52
60	0.47	0.47	0.54	0.51	0.62	0.48	0.50	0.56	0.48
65	0.53	0.71	0.47	0.62	0.61	0.59	0.59	0.54	0.46
70	0.53	0.51	0.62	0.57	0.54	0.47	0.56	0.48	0.55
75	0.53	0.47	0.47	0.47	0.48	0.45	0.52	0.54	0.60
80	0.47	0.47	0.64	0.47	0.52	0.46	0.55	0.62	0.46
85	0.47	0.53	0.61	0.45	0.64	0.68	0.55	0.48	0.47
90	0.47	0.47	0.64	0.50	0.58	0.68	0.64	0.48	0.47
95	0.53	0.47	0.53	0.71	0.48	0.57	0.54	0.47	0.59
100	0.47	0.53	0.47	0.47	0.45	0.47	0.60	0.60	0.47

Figure 6.27: Classification accuracy heatmap synthetic 2

6.2.6 Synthetic 3

The third synthetic set consists of 10,000 instances and two attributes. The classification performance is shown in Figure 6.28. With a k-value of 5, the highest classification accuracy of 0.94 is achieved, with a baseline accuracy of 0.96. The performance results show a clear decline in performance when increasing k. Additional results are included in Appendix A.10.

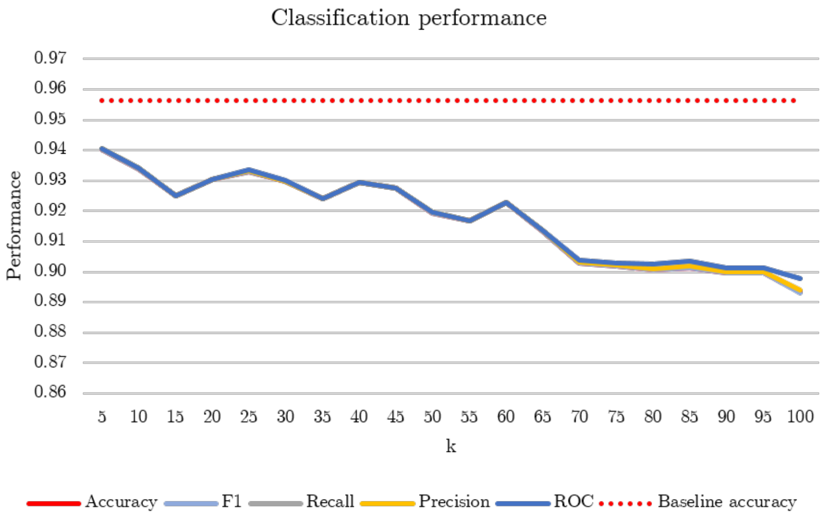


Figure 6.28: Classification performance synthetic 3

Figure 6.23 shows a comparison with the other discretization methods when it comes to the classification accuracy. EW results in an accuracy of 0.95, EF obtains 0.96, while MDLP results in an accuracy of 0.92.

The classification accuracy heatmap is shown in Figure 6.30. Compared to the first two synthetic data sets, it is more clear that the top right part, with lower k-values and a larger training set, results in a higher accuracy. However, high accuracy results are still achieved with less training instances.

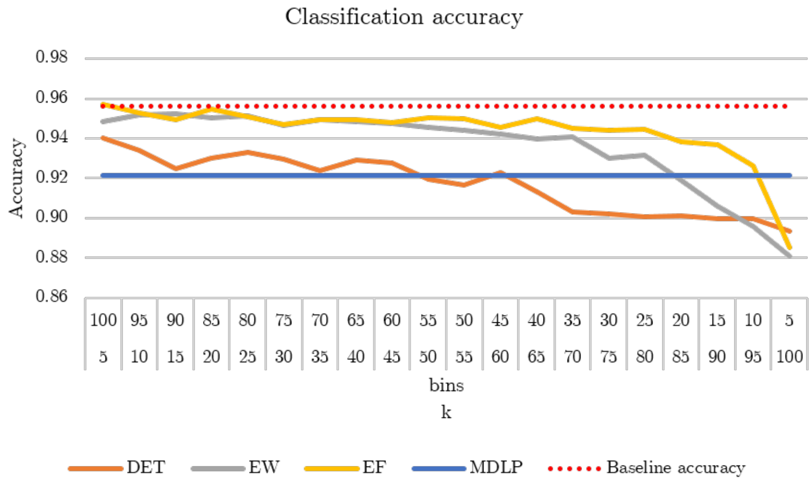
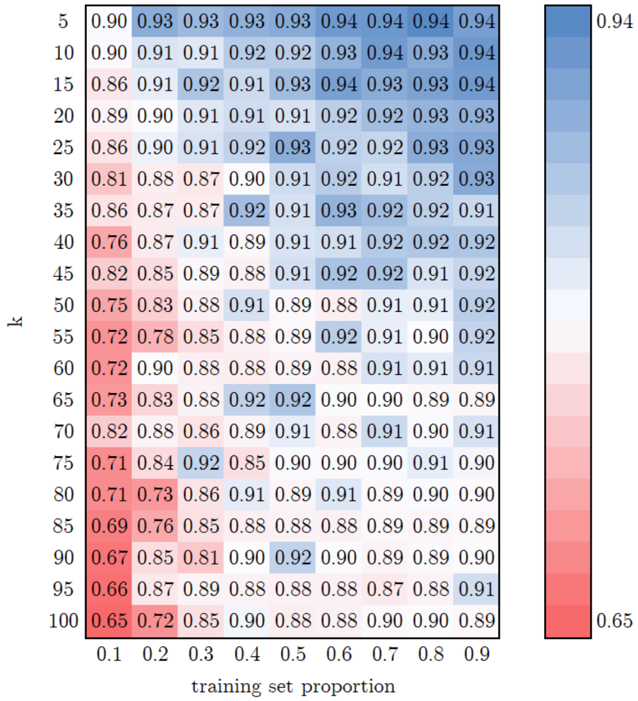


Figure 6.29: Classification accuracy comparison synthetic 3



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.90	0.93	0.93	0.93	0.93	0.94	0.94	0.94	0.94
10	0.90	0.91	0.91	0.92	0.92	0.93	0.94	0.93	0.94
15	0.86	0.91	0.92	0.91	0.93	0.94	0.93	0.93	0.94
20	0.89	0.90	0.91	0.91	0.91	0.92	0.92	0.93	0.93
25	0.86	0.90	0.91	0.92	0.93	0.92	0.92	0.93	0.93
30	0.81	0.88	0.87	0.90	0.91	0.92	0.91	0.92	0.93
35	0.86	0.87	0.87	0.92	0.91	0.93	0.92	0.92	0.91
40	0.76	0.87	0.91	0.89	0.91	0.91	0.92	0.92	0.92
45	0.82	0.85	0.89	0.88	0.91	0.92	0.92	0.91	0.92
50	0.75	0.83	0.88	0.91	0.89	0.88	0.91	0.91	0.92
55	0.72	0.78	0.85	0.88	0.89	0.92	0.91	0.90	0.92
60	0.72	0.90	0.88	0.88	0.89	0.88	0.91	0.91	0.91
65	0.73	0.83	0.88	0.92	0.92	0.90	0.90	0.89	0.89
70	0.82	0.88	0.86	0.89	0.91	0.88	0.91	0.90	0.91
75	0.71	0.84	0.92	0.85	0.90	0.90	0.90	0.91	0.90
80	0.71	0.73	0.86	0.91	0.89	0.91	0.89	0.90	0.90
85	0.69	0.76	0.85	0.88	0.88	0.88	0.89	0.89	0.89
90	0.67	0.85	0.81	0.90	0.92	0.90	0.89	0.89	0.90
95	0.66	0.87	0.89	0.88	0.88	0.88	0.87	0.88	0.91
100	0.65	0.72	0.85	0.90	0.88	0.88	0.90	0.90	0.89

Figure 6.30: Classification accuracy heatmap synthetic 3

6.2.7 Synthetic 4

Synthetic set 4 has 20,000 instances and two attributes. Figure 6.31 shows the classification performance on the test set. The highest classification accuracy of 0.94 is achieved at $k = 15$, which is slightly higher than the accuracy on the continuous data. The additional results are shown in Appendix A.11.

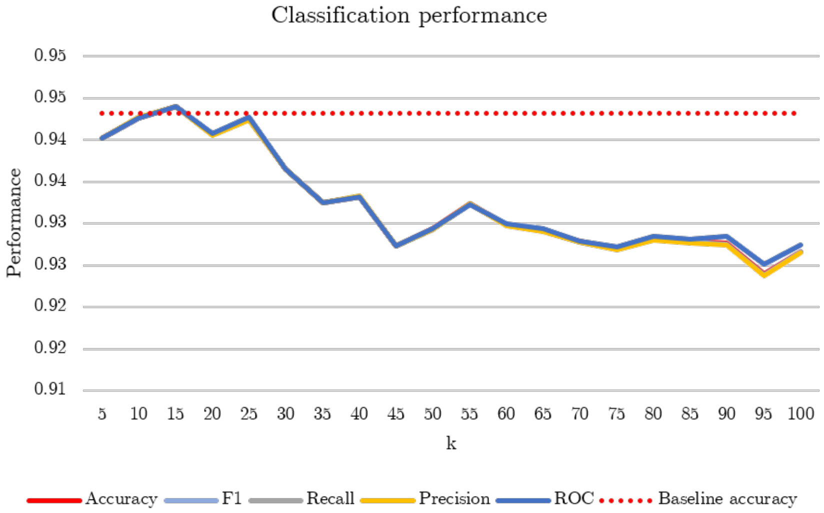


Figure 6.31: Classification performance synthetic 4

Figure 6.32 shows a comparison with the other discretization methods in terms of accuracy. In this case, all discretization methods are able to achieve an accuracy close to the accuracy that is achieved without discretization. Both EW and EF can obtain an accuracy of 0.95, and MDLP an accuracy of 0.94.

The heatmap regarding classification accuracy values is shown in Figure 6.33. Most of the higher accuracy values are obtained in the top right part of the heatmap. In addition, an accuracy of 0.95 is achieved with 90% of the training set, which is slightly higher than the evaluation on the complete training set.

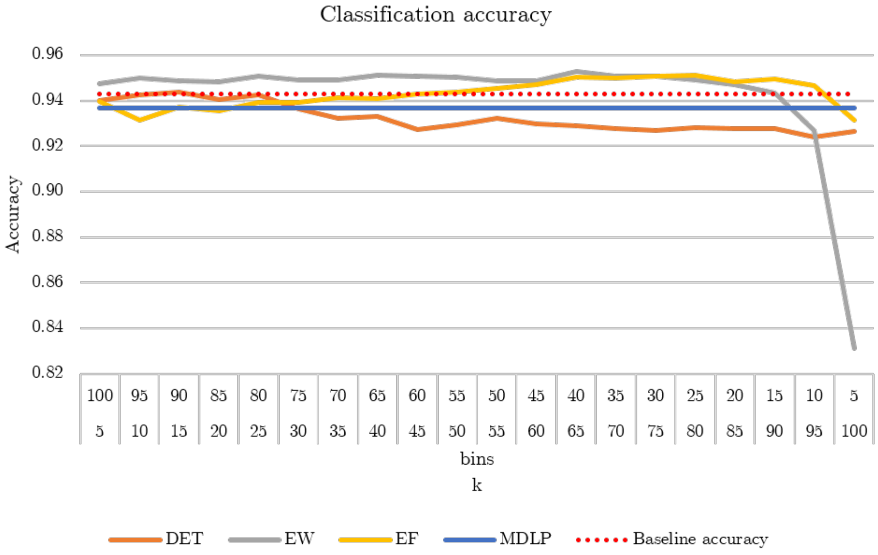
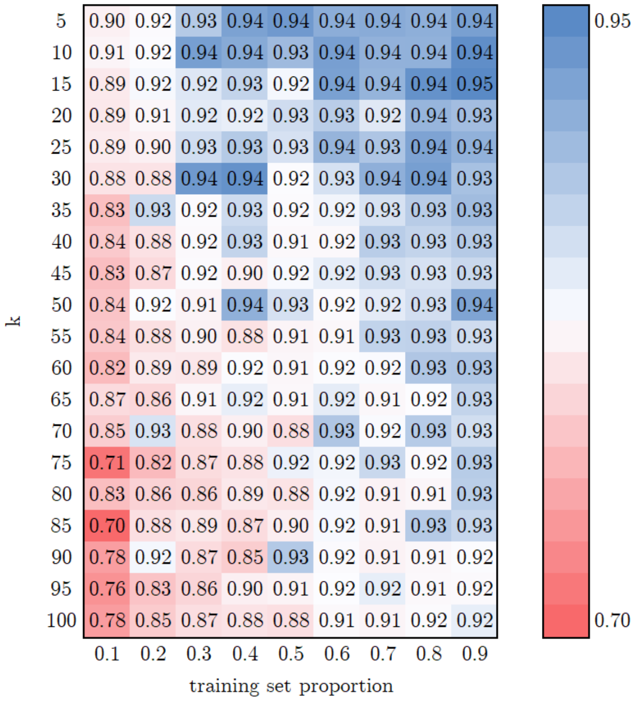


Figure 6.32: Classification accuracy comparison synthetic 4



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.90	0.92	0.93	0.94	0.94	0.94	0.94	0.94	0.94
10	0.91	0.92	0.94	0.94	0.93	0.94	0.94	0.94	0.94
15	0.89	0.92	0.92	0.93	0.92	0.94	0.94	0.94	0.95
20	0.89	0.91	0.92	0.92	0.93	0.93	0.92	0.94	0.93
25	0.89	0.90	0.93	0.93	0.93	0.94	0.93	0.94	0.94
30	0.88	0.88	0.94	0.94	0.92	0.93	0.94	0.94	0.93
35	0.83	0.93	0.92	0.93	0.92	0.92	0.93	0.93	0.93
40	0.84	0.88	0.92	0.93	0.91	0.92	0.93	0.93	0.93
45	0.83	0.87	0.92	0.90	0.92	0.92	0.93	0.93	0.93
50	0.84	0.92	0.91	0.94	0.93	0.92	0.92	0.93	0.94
55	0.84	0.88	0.90	0.88	0.91	0.91	0.93	0.93	0.93
60	0.82	0.89	0.89	0.92	0.91	0.92	0.92	0.93	0.93
65	0.87	0.86	0.91	0.92	0.91	0.92	0.91	0.92	0.93
70	0.85	0.93	0.88	0.90	0.88	0.93	0.92	0.93	0.93
75	0.71	0.82	0.87	0.88	0.92	0.92	0.93	0.92	0.93
80	0.83	0.86	0.86	0.89	0.88	0.92	0.91	0.91	0.93
85	0.70	0.88	0.89	0.87	0.90	0.92	0.91	0.93	0.93
90	0.78	0.92	0.87	0.85	0.93	0.92	0.91	0.91	0.92
95	0.76	0.83	0.86	0.90	0.91	0.92	0.92	0.91	0.92
100	0.78	0.85	0.87	0.88	0.88	0.91	0.91	0.92	0.92

Figure 6.33: Classification accuracy heatmap synthetic 4

6.2.8 Synthetic 5

Synthetic set 5 is created with the *make_moons* utility, and has 1000 instances and two attributes. Figures 6.34 shows the classification performance on the test set, for different k-values. Between $k = 60$ and $k = 85$, the highest classification accuracy of 0.99 is achieved, while the baseline accuracy is 0.99 as well. An interesting observation is that there is no clear decline in accuracy when increasing k, in contrast to the results of the previous data sets. Appendix A.12 contains additional results for this data set.

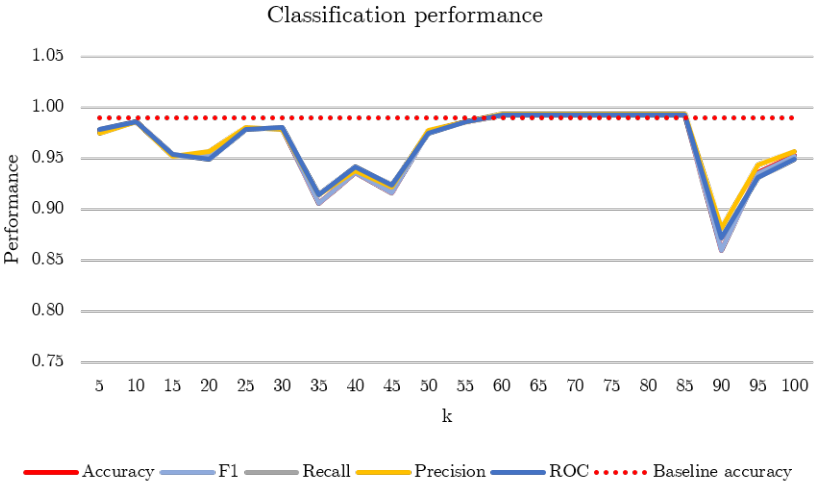


Figure 6.34: Classification performance synthetic 5

Figure 6.35 shows a comparison with other discretization methods. EW, EF and MDLP all achieved an accuracy of 1.

Figure 6.36 shows the classification accuracy heatmap for this data set. This shows that most of the higher accuracy results are obtained in the top right part, although a high accuracy can also be achieved at higher k-values and less training instances.

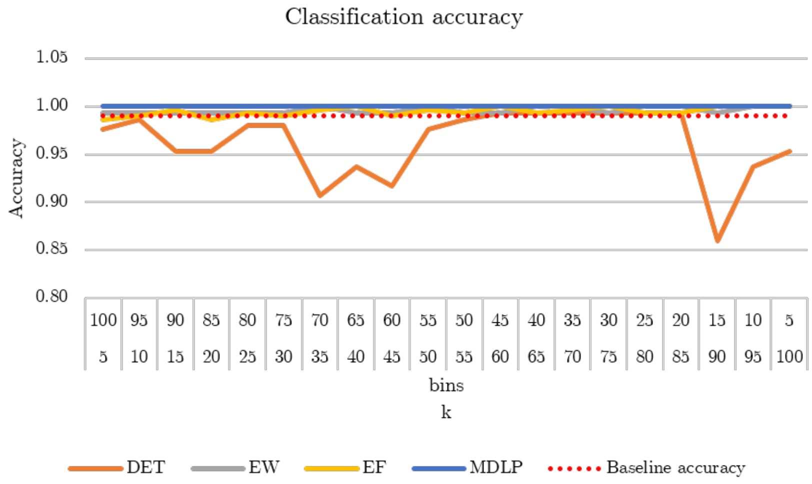
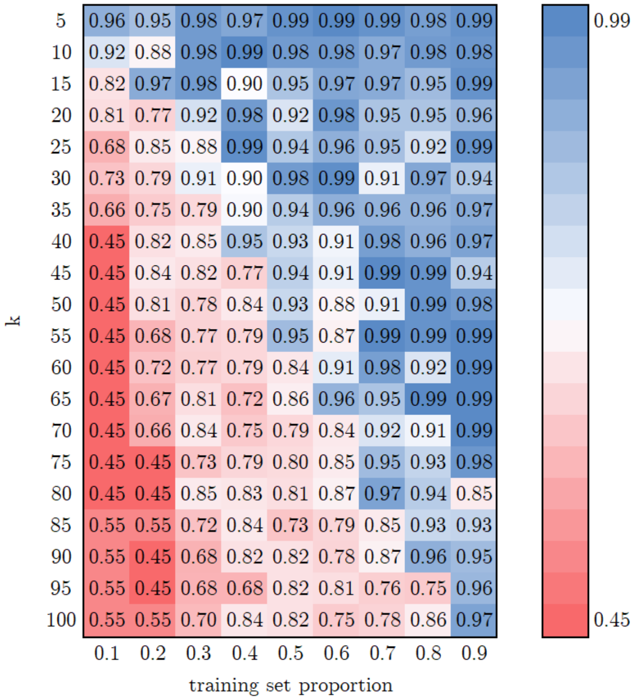


Figure 6.35: Classification accuracy comparison synthetic 5



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.96	0.95	0.98	0.97	0.99	0.99	0.99	0.98	0.99
10	0.92	0.88	0.98	0.99	0.98	0.98	0.97	0.98	0.98
15	0.82	0.97	0.98	0.90	0.95	0.97	0.97	0.95	0.99
20	0.81	0.77	0.92	0.98	0.92	0.98	0.95	0.95	0.96
25	0.68	0.85	0.88	0.99	0.94	0.96	0.95	0.92	0.99
30	0.73	0.79	0.91	0.90	0.98	0.99	0.91	0.97	0.94
35	0.66	0.75	0.79	0.90	0.94	0.96	0.96	0.96	0.97
40	0.45	0.82	0.85	0.95	0.93	0.91	0.98	0.96	0.97
45	0.45	0.84	0.82	0.77	0.94	0.91	0.99	0.99	0.94
50	0.45	0.81	0.78	0.84	0.93	0.88	0.91	0.99	0.98
55	0.45	0.68	0.77	0.79	0.95	0.87	0.99	0.99	0.99
60	0.45	0.72	0.77	0.79	0.84	0.91	0.98	0.92	0.99
65	0.45	0.67	0.81	0.72	0.86	0.96	0.95	0.99	0.99
70	0.45	0.66	0.84	0.75	0.79	0.84	0.92	0.91	0.99
75	0.45	0.45	0.73	0.79	0.80	0.85	0.95	0.93	0.98
80	0.45	0.45	0.85	0.83	0.81	0.87	0.97	0.94	0.85
85	0.55	0.55	0.72	0.84	0.73	0.79	0.85	0.93	0.93
90	0.55	0.45	0.68	0.82	0.82	0.78	0.87	0.96	0.95
95	0.55	0.45	0.68	0.68	0.82	0.81	0.76	0.75	0.96
100	0.55	0.55	0.70	0.84	0.82	0.75	0.78	0.86	0.97

Figure 6.36: Classification accuracy heatmap synthetic 5

6.2.9 Synthetic 6

Synthetic data set 6 is generated with the *make_circles* utility, and consists of 1000 instances and two attributes. Figure 6.37 shows the classification performance for different levels of k. The highest classification accuracy is 0.93, while an accuracy of 1 is achieved with a classifier on the original, not discretized set. In general, a decline in performance is apparent when increasing k, although there is an increase around $k = 50$ after an initial decrease.

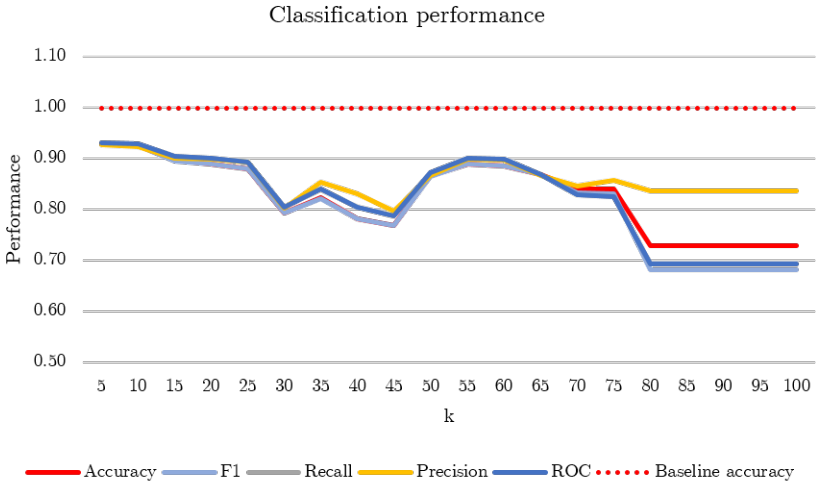


Figure 6.37: Classification performance synthetic 6

Figure 6.38 shows the comparison with other discretization methods. Both EW and EF can achieve an accuracy of 1, just like the results without discretization. MDLP obtains an accuracy of 0.90, which is slightly less than the results for DET discretization for low k-values.

The classification accuracy heatmap is shown in Figure 6.39. The highest accuracy that is achieved is in the top right cell. This accuracy of 0.96 is even higher than the 0.93 that was achieved with the complete set of training instances.

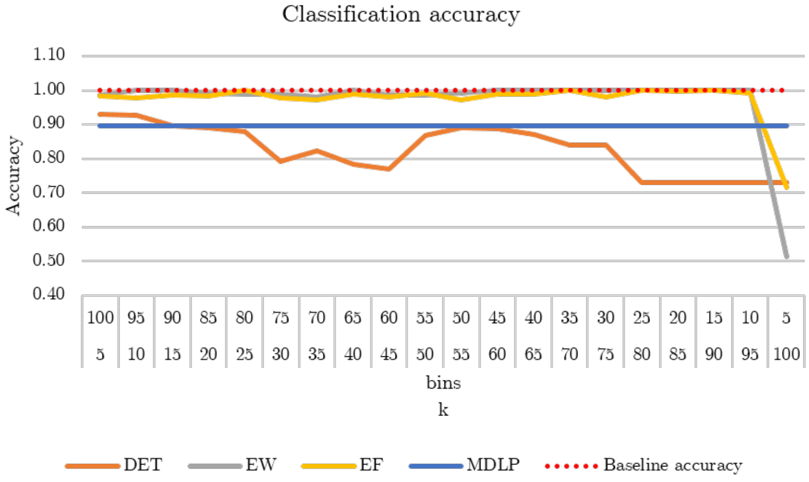
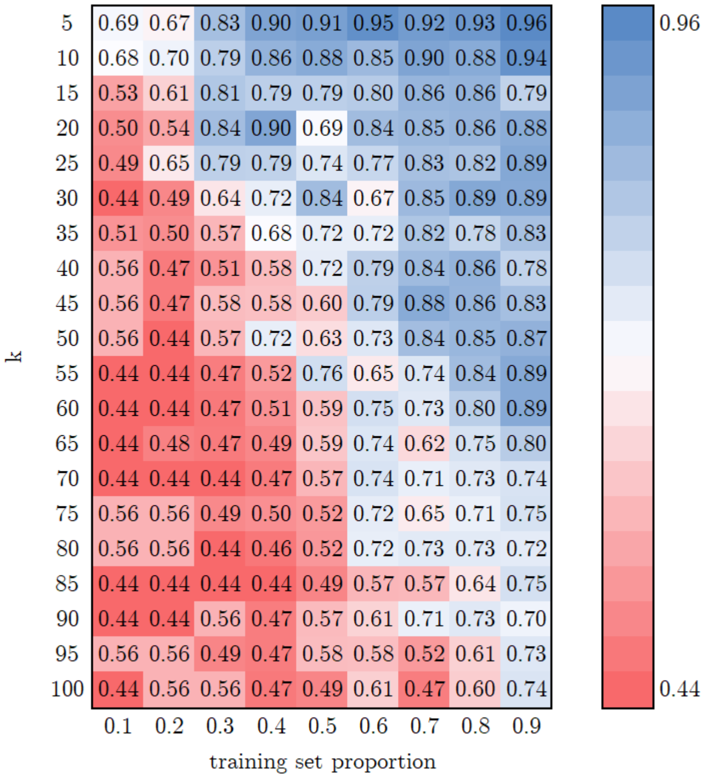


Figure 6.38: Classification accuracy comparison synthetic 6



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.69	0.67	0.83	0.90	0.91	0.95	0.92	0.93	0.96
10	0.68	0.70	0.79	0.86	0.88	0.85	0.90	0.88	0.94
15	0.53	0.61	0.81	0.79	0.79	0.80	0.86	0.86	0.79
20	0.50	0.54	0.84	0.90	0.69	0.84	0.85	0.86	0.88
25	0.49	0.65	0.79	0.79	0.74	0.77	0.83	0.82	0.89
30	0.44	0.49	0.64	0.72	0.84	0.67	0.85	0.89	0.89
35	0.51	0.50	0.57	0.68	0.72	0.72	0.82	0.78	0.83
40	0.56	0.47	0.51	0.58	0.72	0.79	0.84	0.86	0.78
45	0.56	0.47	0.58	0.58	0.60	0.79	0.88	0.86	0.83
50	0.56	0.44	0.57	0.72	0.63	0.73	0.84	0.85	0.87
55	0.44	0.44	0.47	0.52	0.76	0.65	0.74	0.84	0.89
60	0.44	0.44	0.47	0.51	0.59	0.75	0.73	0.80	0.89
65	0.44	0.48	0.47	0.49	0.59	0.74	0.62	0.75	0.80
70	0.44	0.44	0.44	0.47	0.57	0.74	0.71	0.73	0.74
75	0.56	0.56	0.49	0.50	0.52	0.72	0.65	0.71	0.75
80	0.56	0.56	0.44	0.46	0.52	0.72	0.73	0.73	0.72
85	0.44	0.44	0.44	0.44	0.49	0.57	0.57	0.64	0.75
90	0.44	0.44	0.56	0.47	0.57	0.61	0.71	0.73	0.70
95	0.56	0.56	0.49	0.47	0.58	0.58	0.52	0.61	0.73
100	0.44	0.56	0.56	0.47	0.49	0.61	0.47	0.60	0.74

Figure 6.39: Classification accuracy heatmap synthetic 6

6.2.10 Synthetic 7

Synthetic data set 7 is generated through the *make_blobs* utility, and consists of two very distinct clusters in two dimensions, with 1000 instances. The classification performance results are shown in Figure 6.40, while additional results are included in Appendix A.14. The classification accuracy that is achieved without discretization is 1. The same accuracy of 1 is achieved with the anonymization approach, at least with k-values from 5 to 100. In fact, up until a partitioning of the data in two parts, an accuracy of 100% is achieved. Since this set contains 1000 instances, of which 700 are used by the discretization algorithm, the maximum k-value to end up with two bins is $700/2 = 350$. Higher k-values will result in one large interval containing all instances, making the classification task impossible. So, in this hypothetical case, all possible instances belong to one of the two clusters present in the data, so having just two bins would be enough to accurately classify even new instances.

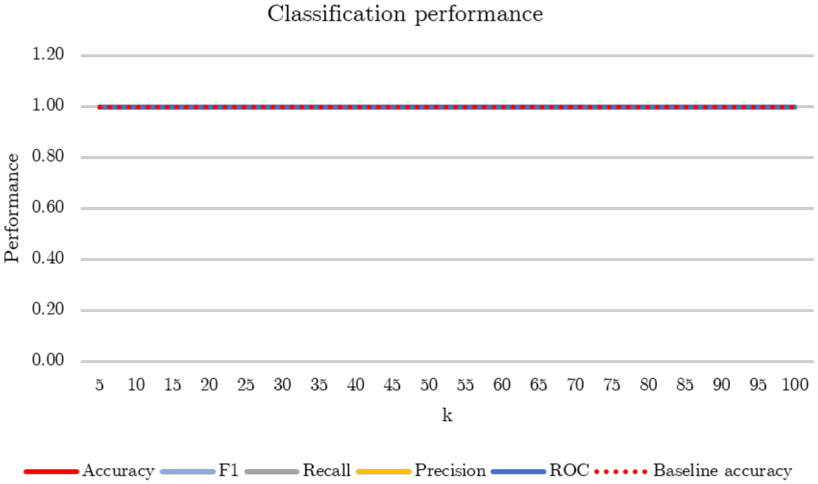


Figure 6.40: Classification performance synthetic 7

Figure 6.41 shows a comparison with the other discretization methods. All methods can achieve an accuracy of 1, as expected. This particular data set is a simple classification problem, so discretization does not hurt the classification performance. This is also illustrated by the classification accuracy heatmap in Figure 6.42. As long as at least two parts are created with the DET discretization approach, the utility is hardly sacrificed. For example, with a proportion of 0.1 and $k = 35$, there are $1000 * 0.7 * 0.1 = 70$ training instances. With $k = 35$, this means that $70/35 = 2$ bins are created, which results in an accuracy of 0.97. However, at $k = 40$, a maximum of $70/40 = 1.75$ bins can be created, which is less than two bins. This means that just one bin is created, which is reflected by the accuracy that is achieved for this k-value.

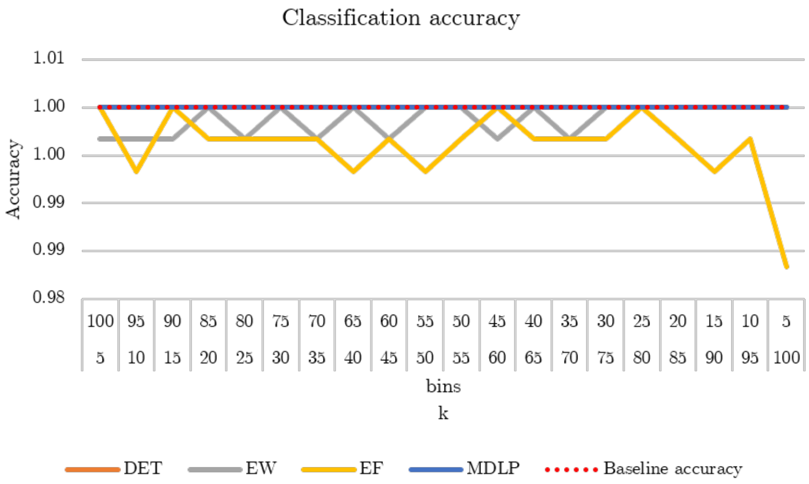
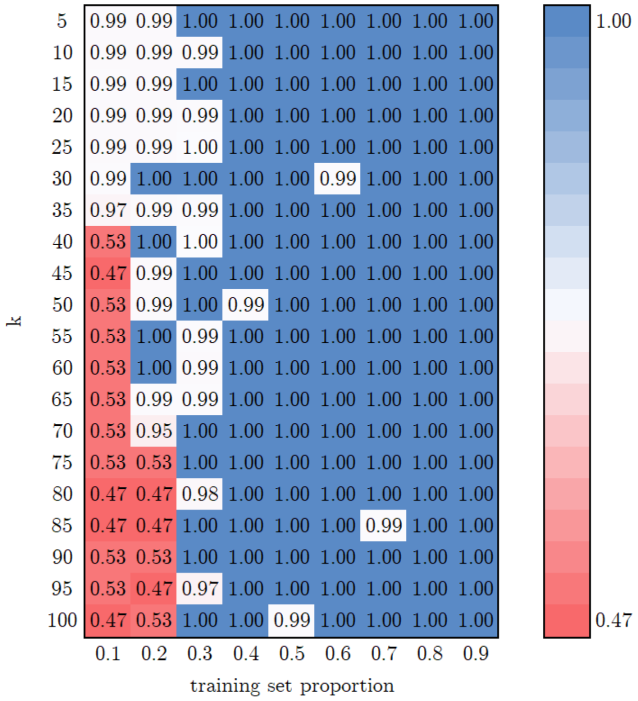


Figure 6.41: Classification accuracy comparison synthetic 7



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
15	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
20	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
25	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
30	0.99	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
35	0.97	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
40	0.53	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
45	0.47	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
50	0.53	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00
55	0.53	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
60	0.53	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
65	0.53	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
70	0.53	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00
75	0.53	0.53	1.00	1.00	1.00	1.00	1.00	1.00	1.00
80	0.47	0.47	0.98	1.00	1.00	1.00	1.00	1.00	1.00
85	0.47	0.47	1.00	1.00	1.00	1.00	0.99	1.00	1.00
90	0.53	0.53	1.00	1.00	1.00	1.00	1.00	1.00	1.00
95	0.53	0.47	0.97	1.00	1.00	1.00	1.00	1.00	1.00
100	0.47	0.53	1.00	1.00	0.99	1.00	1.00	1.00	1.00

Figure 6.42: Classification accuracy heatmap synthetic 7

6.2.11 Synthetic 8

Synthetic data set 8 is similar to 7, in that it is constructed with *make_blobs*, and it contains 1000 instances over two dimensions. However, this set consists of six distinct clusters. The classification performance results are shown in Figure 6.43. The highest classification accuracy that is achieved is 0.89, and the accuracy with classification without discretization is 0.89 as well. Additional results are included in Appendix A.15. In this case, Table A.15 reports k-values ranging from 5 to 150. The results show a trend that is consistent with results obtained with other data sets. Up until a partitioning in six bins, corresponding to a k-value of 110, the performance remains relatively stable. Higher k-values result in fewer bins, and in a decline in classification performance. This particular data set has six classes to predict. Each cluster represents one class. When the discretization algorithm partitions the data in six parts, this is still enough information for the classifier. Having less bins than the number of distinct class labels ultimately results in a substantial performance decline.

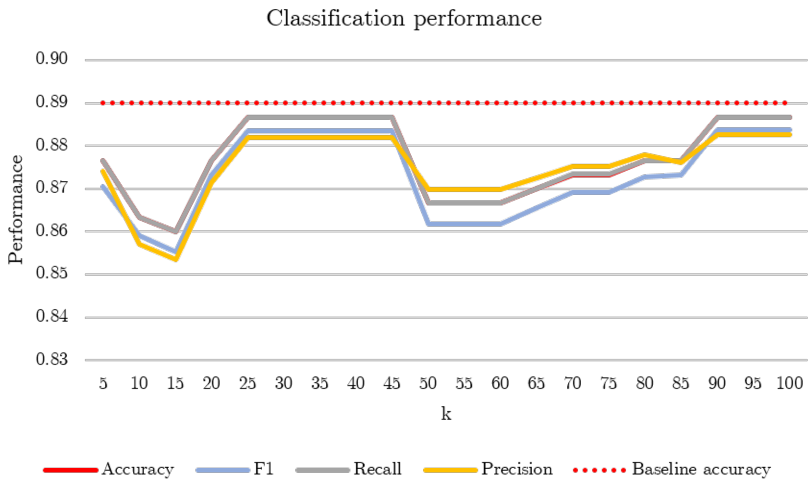


Figure 6.43: Classification performance synthetic 8

Figure 6.44 shows the classification accuracy comparison with other discretization methods. The accuracy that is achieved is similar for the different methods. With EW, an accuracy of 0.90 is achieved, while EF and MDLP resulted in an accuracy of 0.88. The classification accuracy heatmap is shown in Figure 6.45. Most of the higher accuracy scores are in the top right part, although some higher scores are still obtained at higher k-values.

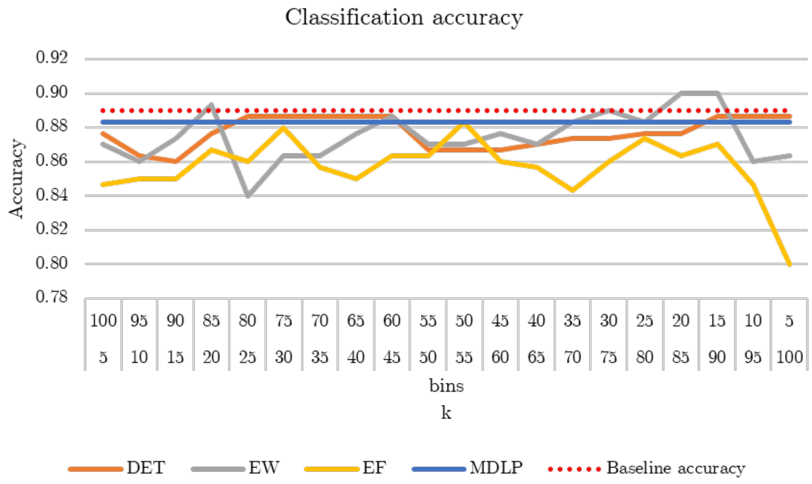
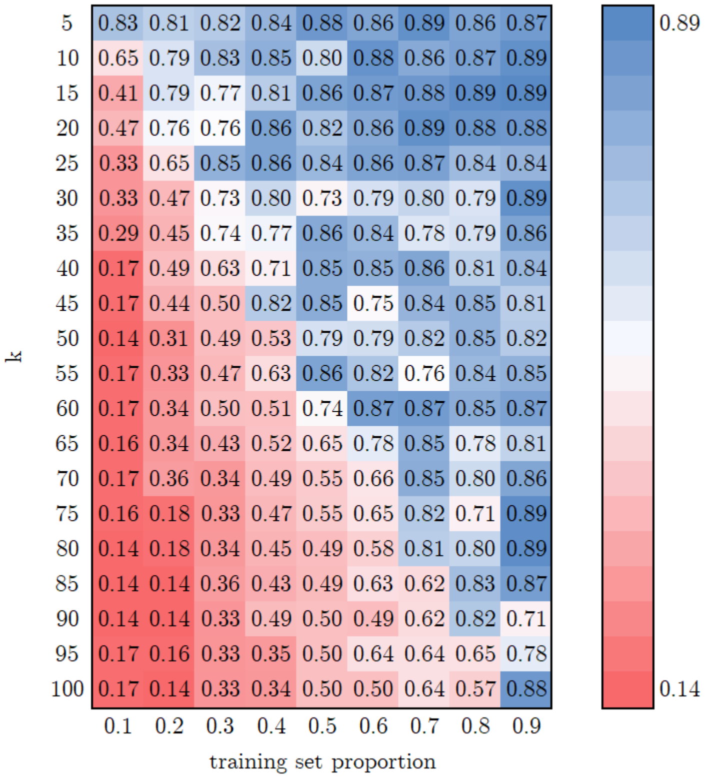


Figure 6.44: Classification accuracy comparison synthetic 8



k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.83	0.81	0.82	0.84	0.88	0.86	0.89	0.86	0.87
10	0.65	0.79	0.83	0.85	0.80	0.88	0.86	0.87	0.89
15	0.41	0.79	0.77	0.81	0.86	0.87	0.88	0.89	0.89
20	0.47	0.76	0.76	0.86	0.82	0.86	0.89	0.88	0.88
25	0.33	0.65	0.85	0.86	0.84	0.86	0.87	0.84	0.84
30	0.33	0.47	0.73	0.80	0.73	0.79	0.80	0.79	0.89
35	0.29	0.45	0.74	0.77	0.86	0.84	0.78	0.79	0.86
40	0.17	0.49	0.63	0.71	0.85	0.85	0.86	0.81	0.84
45	0.17	0.44	0.50	0.82	0.85	0.75	0.84	0.85	0.81
50	0.14	0.31	0.49	0.53	0.79	0.79	0.82	0.85	0.82
55	0.17	0.33	0.47	0.63	0.86	0.82	0.76	0.84	0.85
60	0.17	0.34	0.50	0.51	0.74	0.87	0.87	0.85	0.87
65	0.16	0.34	0.43	0.52	0.65	0.78	0.85	0.78	0.81
70	0.17	0.36	0.34	0.49	0.55	0.66	0.85	0.80	0.86
75	0.16	0.18	0.33	0.47	0.55	0.65	0.82	0.71	0.89
80	0.14	0.18	0.34	0.45	0.49	0.58	0.81	0.80	0.89
85	0.14	0.14	0.36	0.43	0.49	0.63	0.62	0.83	0.87
90	0.14	0.14	0.33	0.49	0.50	0.49	0.62	0.82	0.71
95	0.17	0.16	0.33	0.35	0.50	0.64	0.64	0.65	0.78
100	0.17	0.14	0.33	0.34	0.50	0.50	0.64	0.57	0.88

Figure 6.45: Classification accuracy heatmap synthetic 8

6.3 Runtime

For each data set used in the evaluation and discussed in the previous sections, a comparison between the four discretization methods regarding runtime is included in Appendix A. In this section, the runtime of the DET anonymization approach is compared between a selection of data sets. This could provide more information about how data set dimensions affect the runtime of the algorithm.

Figure 6.46 shows the runtime for the three extended *iris*, of which the results were discussed in Section 6.2, and additional results are shown in Appendices A.5, A.6, and A.7. These data sets consist of 150 instances and five, nine, and fourteen attributes respectively. In all cases, there is an exponential decline in runtime when increasing k. As shown in Tables A.5, A.6, and A.7, this corresponds with the exponential decline in the number of bins that are created for different k-values. In addition, there is a noticeable difference in the time it takes to construct a DET on the three variations of the data, especially at low k-values. This indicates that having more dimensions increases the runtime. In addition, the difference in runtime between iris + 5 and iris + 10 is larger than that between iris + 1 and iris + 5. In this case, the results indicate that the runtime scales exponentially with the amount of dimensions, rather than linearly.

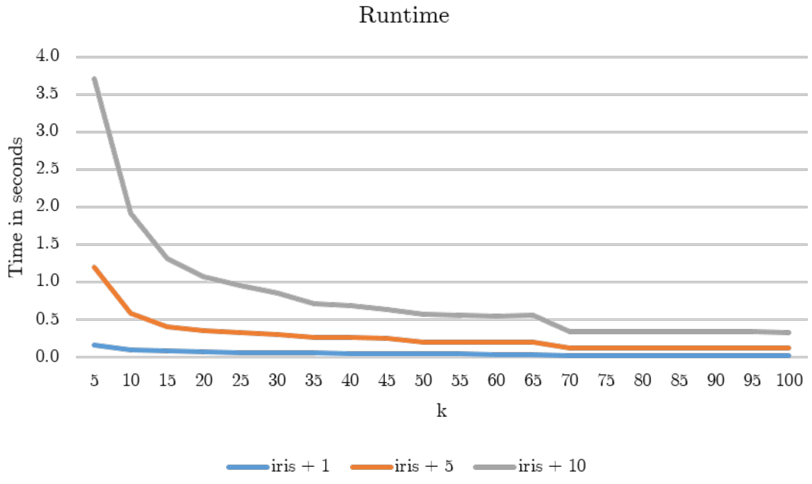


Figure 6.46: Runtime extended iris

A similar comparison can be made between synthetic data sets 1 and 2, which both consist of 1000 instances, but five and ten attributes respectively. Figure 6.47 shows the runtime of both data sets. In this case, there is a large difference between the two sets. At $k = 5$, the runtime for synthetic set 1 is 6.8 seconds, while it is 225 seconds for synthetic set 2. At higher k-values, the difference in runtime decreases. In addition, the runtime declines exponentially over k, although this is most noticeable for synthetic set 2. Tables A.8 and A.9 show the same trend regarding the number of bins that are created for each k. The number of bins is also very similar for both sets, as they have the same amount of instances.

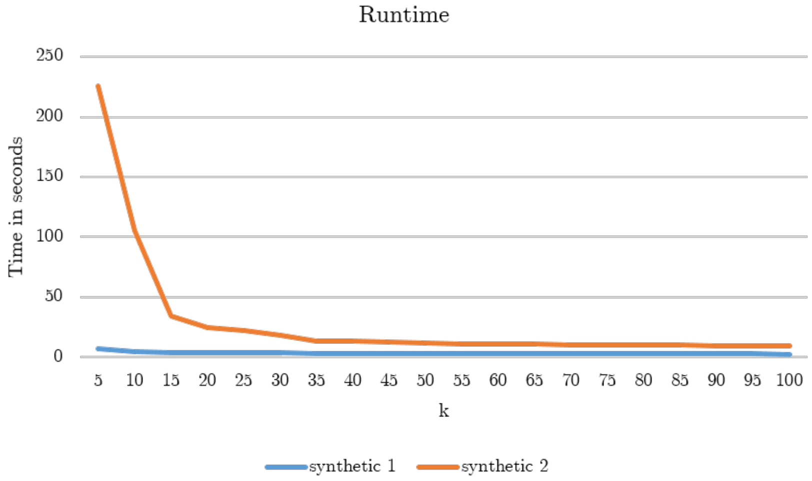


Figure 6.47: Runtime synthetic 1 and 2

Synthetic sets 3 and 4 are similar in that they both consist of two attributes. However, these data sets differ in their number of instances. However, synthetic set 3 contains 10,000 instances, while synthetic 4 has 20,000. Figure 6.48 shows the runtime for these data sets. Synthetic set 3 shows a slight overall decrease in runtime between 105 and 87 seconds, while the runtime of synthetic set 4 remains relatively stable around 340 seconds. This contrasts the more noticeable trends in runtime when increasing k for most of the other data sets. Tables A.10 and A.11 do show an exponential decrease in the amount of bins that are created for both sets. These results also show that generally speaking, the amount of bins that are created for synthetic set 4 is about double the amount of bins that are created for synthetic set 3 for all k-values. This is not unexpected since synthetic 4 contains double the amount of instances as synthetic 3.

The last comparison is between synthetic sets 5, 6, 7 and 8. These all consist of 1000 instances, and two attributes, so their dimensions are exactly the same. However, as explained in section 5.3.2, the clusters in the data are differently shaped. Tables A.12, A.13, A.14 and A.15 show that for each of these data sets, the amounts of created bins are roughly the same for the various k-values. Figure 6.49 shows how the runtime differs for these sets. Synthetic set 7 and 8 were both created using the *make_blobs* utility, and their runtime is very similar. Synthetic sets 5 and 6 were created with *make_moons* and *make_circles* respectively. Their runtimes are higher, most noticeable at lower k-values. This indicates that the runtime is not necessarily directly deducible from the dimensions of the data.

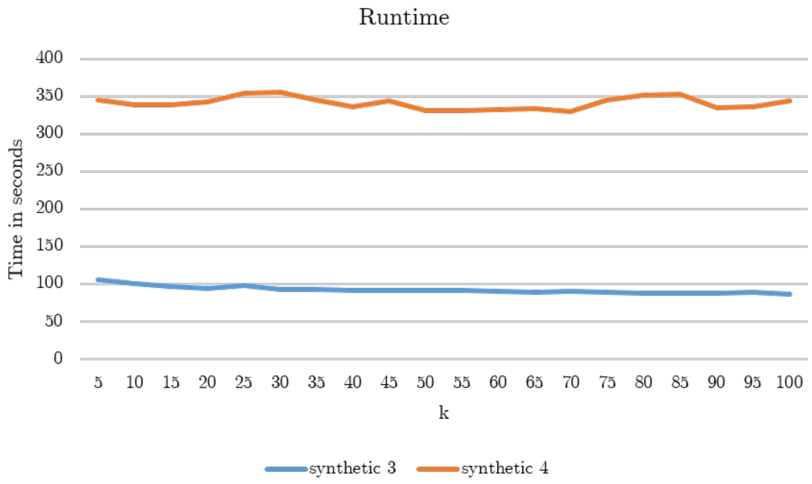


Figure 6.48: Runtime synthetic 3 and 4

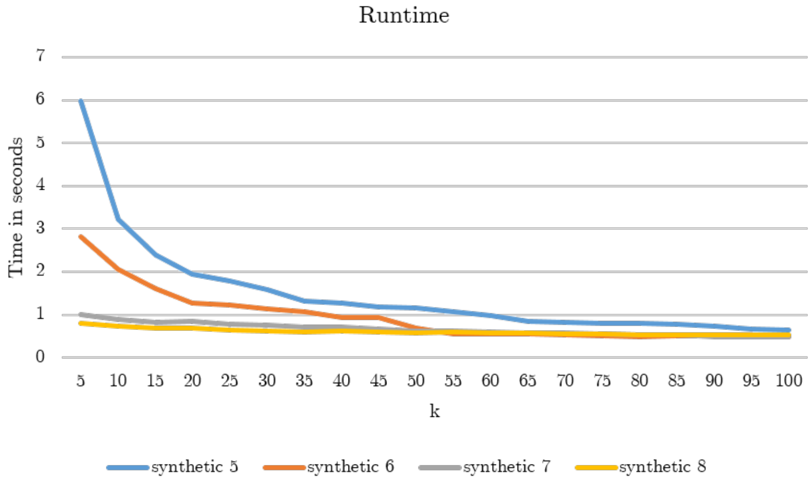


Figure 6.49: Runtime synthetic 5 - 8

6.4 Discussion

The results of all the evaluated data sets were discussed in the previous sections of this chapter. Table 6.1 contains an overview of the highest accuracy that is achieved for each data set for the discretization methods, and the corresponding input parameter values. In the case of DET discretization, when multiple k-values result in the same accuracy, the highest of these k-values is included in the table, since this means a higher privacy level, less bins, and therefore a less complex model. When this is the case for EW and EF, the lowest number of bins is included, since this represents a less complex model as well. The accuracy values in bold indicate the highest accuracy that is achieved for that data set. The DET column indicate whether the accuracy that is achieved is within 0.05 from the highest accuracy of all discretization methods for that data set. Green cells indicate values within that range, red cells indicate that the value is not in that range.

Table 6.1: Classification accuracy comparison

Data set	**Original**	**DET**	**k**	**EW**	**h**	**EF**	**h**	**MDLP**
breast	0.96	0.95	15	**0.97**	80	**0.97**	60	**0.97**
glass	0.66	0.58	5	**0.77**	45	0.76	70	0.75
iris	0.96	0.95	5	0.96	55	**0.97**	80	0.96
wine	0.97	0.90	5	**0.98**	30	**0.98**	30	**0.98**
iris + 1	0.97	0.94	40	**0.97**	15	**0.97**	15	0.94
iris + 5	0.93	0.92	40	**0.96**	30	**0.96**	20	0.93
iris + 10	0.92	0.94	40	**0.97**	30	**0.97**	80	0.95
Synthetic 1	0.89	0.77	40	0.88	50	**0.89**	75	0.84
Synthetic 2	0.86	0.67	10	**0.88**	55	**0.88**	20	0.80
Synthetic 3	0.96	0.94	5	0.95	90	**0.96**	100	0.92
Synthetic 4	0.94	0.94	15	**0.95**	40	**0.95**	25	0.94
Synthetic 5	0.99	0.99	85	**1.00**	5	**1.00**	5	**1.00**
Synthetic 6	1.00	0.93	5	**1.00**	10	**1.00**	15	0.90
Synthetic 7	**1.00**	1.00	100	**1.00**	5	**1.00**	25	**1.00**
Synthetic 8	0.89	0.89	100	**0.90**	15	0.88	50	0.88

This overview shows that the DET discretization method does not achieve the highest accuracy on any of the data sets, except for a shared best accuracy for synthetic set 7. However, for ten data sets, the accuracy with DET discretization falls within a range of 0.05, of which 7 are within 0.02. In some cases, the classification accuracy heatmaps have shown that a higher accuracy is sometimes achieved with less training instances.

The key difference between DET discretization to achieve k-anonymity and the other discretization methods is that the anonymization approach partitions the data *horizontally* in a multivariate fashion, while the other methods do not. For example, equal-width discretization with ten bins divides each dimension into ten bins separately. This means that there are still many possible combinations of attribute values. Instead, with DET discretization, instances in each bin have the exact same values for *each* dimension as the other instances in that bin. This therefore results in the creation of equivalence classes. In other words, DET discretization partitions the whole data over all attributes into bins, while the other methods discretize each attribute into bins. This could result in much more multivariate bins, which could be beneficial for the preservation of utility. In addition, this also means that the other discretization methods can not be used for anonymization through achieving k-anonymity, which is possible with DET discretization. In any case, the comparison between these discretization techniques still provides an indication of the performance of anonymization through DET discretization. Especially with the *glass* data set and synthetic sets 1 and 2, the difference with the best accuracy for that data set is substantially large, compared to the other results.

The differences between the classification accuracy results are tested for significance. This is done by testing difference of proportions – which would be two accuracy results – through computing the z statistic (Kuncheva, 2004). In a two-sided test with a significance level of 0.05, the null hypothesis H_0 that both accuracies are equal is rejected if $|z| > 1.96$. Table 6.2 shows the $|z|$-values for a difference test between the accuracy achieved with DET and the accuracy with the continuous attributes. It also shows these values for a difference test between the accuracy achieved with DET and the best performing discretization method. The values in bold indicate that $|z| > 1.96$, which would reject the null hypothesis that the accuracies are the same, but instead lead to the conclusion that the accuracies are significantly different. For synthetic sets 1, 2, and 6, the differences in accuracy achieved by DET and the accuracy on the continuous data are significantly different. In addition, for *glass*, *wine*, and synthetic sets 1, 2, and 6, the differences in accuracy achieved by DET and the best performing discretization method are significantly different.

Compared to the continuous attributes, the three data sets that achieved a significantly different accuracy with DET discretization are synthetic sets 1, 2, and 6. Synthetic 1 and 2 were both generated with the $make\_classification$ utility, and contain five and ten attributes respectively. The difference in performance could be caused by the higher number of attributes these two sets have, since the other generated synthetic sets all contain two attributes. In addition, synthetic 2, containing ten attributes, has an even larger difference with the continuous data than synthetic 1, indicating that the method does not perform well on higher number of attributes. This is not always the case however, since breast, glass, wine, and the three extended iris sets have at least five attributes as well, and there is no significant difference in accuracy with those data sets.

Synthetic sets 3 and 4 contain 10,000 and 20,000 instances respectively, and both contain two attributes. In both cases, there classification performance is high, and there is no significant difference with the continuous attributes. This indicates that there is no noticeable connection between the amount of instances and the performance of the DET discretization approach.

Synthetic sets 5, 6, 7, and 8 were generated with *scikit-learn*'s utilities, and contain a certain pattern in the data. Synthetic sets 5, 7, and 8 all have an

Table 6.2: Accuracy differences significance testing

Data set	DET	Continuous	$\lvert z \rvert$	Discrete	$\lvert z \rvert$
breast	0.95	0.96	0.47	0.97	0.72
glass	0.58	0.66	1.20	0.77	**2.92**
iris	0.95	0.96	0.45	0.97	0.83
wine	0.90	0.97	1.91	0.98	**2.40**
iris + 1	0.94	0.97	0.89	0.97	1.02
iris + 5	0.92	0.93	0.36	0.96	1.19
iris + 10	0.94	0.92	0.55	0.97	1.02
Synthetic 1	0.77	0.89	**2.26**	0.89	**2.26**
Synthetic 2	0.67	0.86	**3.17**	0.88	**3.56**
Synthetic 3	0.94	0.96	0.51	0.96	0.64
Synthetic 4	0.94	0.94	0.03	0.95	0.19
Synthetic 5	0.99	0.99	0.26	1.00	0.82
Synthetic 6	0.93	1.00	**2.69**	1.00	**2.69**
Synthetic 7	1.00	1.00	-	1.00	-
Synthetic 8	0.89	0.89	0.07	0.90	0.31

accuracy equal to the accuracy achieved on the continuous attributes. These results indicate that the DET discretization approach works well with the data generated with *make_moons* and *make_blobs*. However, the accuracy achieved with synthetic set 6 is significantly different from the accuracy on the continuous attributes, indicating a low performance on this data set with a circular shape, since it was generated with *make_circles*.

The three variations on *iris* that contain a number of attributes with randomly drawn values all perform well compared to the continuous data. Their highest classification accuracy values do not significantly differ from the accuracy on the continuous attributes. Figure 6.19 indicates that overall, the classification performance is lower than with the original *iris* set in Figure 6.7. However, the accuracy results on the continuous data for these variations slightly decrease as well, as more noise is added. Since there is no significant difference in the accuracies of the continuous and anonymzed data for these variations, these results show that the DET approach is able to handle attributes that do not contribute to meaningful interactions.

When it comes to the runtime, the results in Appendix A show that this is much higher than with the other discretization methods. In the case of EW and EF, almost no computation is required, since the intervals are determined solely based on the number of bins as input parameter. MDLP requires a bit more computation, since it is an entropy-based method to determine the best intervals based on the class information. On the other hand, Density Estimation Trees determine the bins based on the amount of instances that are associated with the proportion of the data in that bins. As discussed in Chapter 4.4, this requires the calculation of the volume of a node. This is an expensive computation, especially considering it needs to be determined for each possible split candidate in each dimension. However, a more efficient implementation of this algorithm than the one created for this evaluation could decrease the runtime drastically.

Considering the trade-off between privacy and utility, many of the results show that the lowest k-values do not always result in the best classification performance. In other words, increasing k by 1 or 5 does not immediately result in a utility decrease. Table 6.1 show that in ten out of fifteen data sets, k-values of 10 and higher resulted in the highest classification accuracy. In addition, choosing higher k-values usually result in only a slight decrease in performance. In general however, there is a noticeable decline in performance when increasing k. This is caused by the fact that less bins are created when k is increased. Knowing the amount of bins that can be created for a given value of k can be used to estimate a reasonable range for this parameter. For example, the results for the *iris* set and its variations, and the results for synthetic set 8 show that the performance drops when less bins are created than there are distinct class labels. Since the *maximum* amount of bins that can be created for a given amount of k is equal to $\frac{n}{k}$, where n is the amount of training instances, this can be used to at least choose a k that can result in as many bins as there are class labels. For example, the *maximum* value for k that could result in three bins would be equal to $\frac{n}{h}$, where h is the number of bins. In the case of *iris*, this would be $\frac{135}{3} = 45$, as nine folds are used as training folds for every iteration. The results for *iris* and the variations show that the performance starts to drop from this k-value. Choosing any higher k-values would therefore fail to preserve the utility. Using the amount of distinct class labels naturally only work with classification tasks, but determining the maximum amount of bins that can be constructed based on k and the training set size should give a general idea of the range of k-values to choose from, even in unsupervised situations. In the context of anonymizing data for supervised learning, an approach similar to the evaluation used in this project can be used to partition the data with respect to the classification performance measures. This would involve determining a minimal acceptable k-value, and estimating the maximum k based on the data and the number of distinct class labels. DETs can then be created over the range of these k-values, and based on some performance measure, one could choose from the best performing one, and use the corresponding k-value for the actual anonymization.

The classification accuracy heatmaps show that having more training instances for the DET does not always result in the highest accuracy. However, a pattern can be identified from these heatmaps. In general, the higher accuracy values are found in the top right part, with lower k-values and more training instances, while the lower accuracy results are found in the bottom left. The heatmaps show that even with less training instances, the utility can be mostly preserved. However, the training set proportion does influence the maximum k-value at which the utility is still preserved. In the case of *iris*, 135 instances and $k = 40$ can result in three bins. With only 75 training instances, $k = 40$ results in $\frac{75}{40} \approx 1.9$ bins, which is not enough to preserve its utility. This is especially the case with smaller data sets like *iris*, but it is noticeable with larger data sets as well, for example synthetic data sets 3 and 4, discussed in Sections 6.2.6 and 6.2.7.

Chapter 7

Conclusions

7.1 Main conclusions

The motivation for this research project included two aspects that were key elements throughout this report: the the benefits of data analysis, and the need to protect the privacy of individuals in the data. The effort to unite these two interests was formulated in the main research question: *"How can data be accurately summarized by as few instances as possible to support data analysis, while preserving the privacy of individuals?"* This research question is addressed throughout the parts of this thesis report, structured by the subquestions as formulated in Section 2.1. In Chapter 3, various models to achieve data privacy were discussed, along with their advantages and disadvantages, thereby addressing **SQ1**. Differential privacy and k-anonymity are among the most well-known privacy models. In addition, **SQ2** was addressed in the same chapter by discussing different discretization techniques for data partitioning. Popular techniques include equal-width and equal-frequency binning, MDLP, 1R, Chi2, and discretization through tree-based density estimation.

SQ3 is concerned with ways of expressing the levels of privacy and utility. Since there is no universally agreed upon measure of data privacy, measuring the level of privacy depends on the privacy model that is used to anonymize data. In fact, the prominent privacy models and their variations in literature have at least one input parameter to express the amount of privacy, within the context of that model. For most models, literature does not prescribe what levels of these privacy expressions are sufficient. On the other hand, expressing the utility is more complicated. A common approach to determine the goodness of fit of discretization models is to train a classifier and evaluate its performance on the discretized data.

The proposed treatment was presented in Chapter 4 to address **SQ4**. This treatment involves privacy preserving, horizontal discretization through density estimation trees, by using k as a stopping rule to achieve k-anonymity. DETs allow for the creation of an accurate, multidimensional model regarding the true density of the data. At the same time, using k as a minleaf constraint, and the leaf nodes as bins, the continuous data can be horizontally partitioned. This results in a set of equivalence classes where each equivalence class contains at least k records.

To address **SQ5**, the treatment was validated by evaluating the classification performance on real-world, semi-synthetic and full synthetic data sets, and compared with other discretization techniques in Chapter 6. In five out of fifteen data sets, the classification accuracy on the anonymized data is the same or higher than using the original, continuous attributes. In nine cases, the difference in accuracy between the original and the anonymized data is within a range of 0.02. In only two out of fifteen cases, the accuracy on the original data is at least 0.1 higher than on the anonymized data. In ten cases, the difference in accuracy between DET discretization and the best performing discretization technique is within 0.05, of which seven are within 0.02. In three out of fifteen cases, the accuracy obtained by the best performing discretizer is at least 0.1 higher than with the proposed treatment. In addition, in ten cases, the highest accuracy for DET discretization was obtained with k-values greater than or equal to 10. Even when higher k-values are desirable, the classification performance is only slightly worse. Significance testing shows that in three out of fifteen cases, the accuracies of the anonymized data and the continuous data are significantly different. It can therefore be concluded that anonymization through DET discretization is able to preserve data utility in most cases, while incorporating privacy guarantees by achieving k-anonymity, for relatively high values of k.

SQ6 involves the relationship between privacy and utility. It is clear that not every single increase in k results in a decline in data utility. There is also no general 'optimal' k-value that balances privacy and utility for all data sets. However, in general, there is a noticeable decrease in utility for higher ranges of k compared to lower ranges. The amount of bins that can be created for a given value of k indicate the maximum k-values. In addition, in supervised cases, the number of bins should at least be equal to the number of distinct class labels to allow for the preservation of utility. The minimum k-value depends on the context. With k equal to 10 or 20, the utility can be preserved in most cases, especially with larger data sets. These values can be considered as a safe minimum for k, especially since current literature states much lower k-values.

Finally **SQ7** is concerned with the minimum amount of instances needed to create an accurate summary of the data. Constructing the DET with less training instances still provides a preservation of utility. However, the utility is affected when the same k-value is used as an input parameter, since less bins can be created with less training instances. In other words, in terms of DET discretization, there is no minimum amount of instances, other than a sensible amount of instances to construct any model on. In addition, when less training instances are used, lower k-values should be used to ensure similar data utility.

This research project can be concluded by answering the main research question. Discretization through density estimation trees can be used to accurately summarize data, while supporting data analysis. Incorporating k as a stopping rule, and horizontally partitioning the data achieves k-anonymity with k as an input parameter. Only a limited amount of training instances is needed to create an accurate discretization model that generalizes over new instances.

7.2 Limitations

The proposed approach towards anonymizing data sets incorporates achieving k-anonymity. Doing so, there is an assumption that k-anonymity is a valid privacy model that provides guarantees about privacy. This is an assumption that some agree on, and some do not, as with most privacy models. Therefore, satisfying privacy concerns by using the proposed anonymization approach in this research project, requires agreeing upon the validity and guarantees of k-anonymity.

Although the proposed anonymization approach obtained good classification performance results for twelve of the fifteen data sets compared to the continuous data, there were three cases where there was a significant difference compared to the continuous attributes. This means that this approach to anonymization does not achieve good results in all cases.

Another limitation is the runtime for constructing the DET, which is substantial, especially with larger data sets. This currently limits the applicability of the anonymization to smaller data sets, for example the ones used during the experimental evaluation.

7.3 Future work

Since this thesis report introduces a new approach towards achieving data privacy, more evaluation can be performed to determine its validity and applicability on a larger range of data sets with various different characteristics.

Future work based on this research project would also include the exploration of incorporating more privacy mechanisms in addition to k-anonymization within the context of the described technique. An additional interest would be to include categorical and ordinal attributes in the anonymization approach, instead of only addressing continuous, numerical attributes. In addition, instead of DETs, other multidimensional, horizontal discretization methods that could incorporate k-anonymity can be studied. From a wider perspective, discretization methods as means to achieve data privacy would make for an interesting field of study as well.

References

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer. doi: 10.1007/978-3-319-14142-8

Aggarwal, C. C., & Philip, S. Y. (2008). A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining* (pp. 11–52). Springer. doi: 10.1007/978-0-387-70992-5_2

Anderlini, L. (2016). Density Estimation Trees as fast non-parametric modelling tools. In *Journal of Physics: Conference Series* (Vol. 762, p. 12042).

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. In *ACM Sigmod record* (Vol. 28, pp. 49–60). doi: 10.1145/304182.304187

Apple. (2017). Learning with Privacy at Scale. *Machine Learning Journal*, *1*(8), 1–25.

Bambauer, J., & Muralidhar, K. (2016, May 17). A Response to the Criticisms of Fool's Gold: An Illustrated Critique of Differential Privacy. *INFO/LAW*. Retrieved from http://blogs.harvard.edu/infolaw/2016/05/17/diffensive-privacy/. (online; accessed 2018-04-17)

Bambauer, J., Muralidhar, K., & Sarathy, R. (2013). Fool's gold: an illustrated critique of differential privacy. *Vand. J. Ent. & Tech. L.*, *16*, 701.

Bay, S. D. (2000). Multivariate discretization of continuous variables for set mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 315–319). doi: 10.1145/347090.347159

Bay, S. D., & Pazzani, M. J. (1999). Detecting change in categorical data: Mining contrast sets. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 302–306). doi: 10.1145/312129.312263

Bertino, E., Lin, D., & Jiang, W. (2008). A survey of quantification of privacy preserving data mining algorithms. In *Privacy-preserving data mining* (pp. 183–205). Springer. doi: 10.1007/978-0-387-70992-5_8

Bezzi, M. (2010). An information theoretic approach for privacy metrics. *Trans. Data Privacy*, *3*(3), 199–215.

Breiman, L. (2001, Oct 01). Random Forests. *Machine Learning*, *45*(1), 5–32. doi: 10.1023/A:1010933404324

De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific* *reports*, *3*, 1376. doi: 10.1038/srep01376

Dheeru, D., & Karra Taniskidou, E. (2017). *UCI machine learning repository.* Retrieved from `http://archive.ics.uci.edu/ml`

Domingo-Ferrer, J., & Torra, V. (2008). A critique of k-anonymity and some of its enhancements. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on* (pp. 990–993). doi: 10.1109/ARES.2008.97

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995* (pp. 194–202). Elsevier. doi: 10.1016/B978-1-55860-377-6.50032-3

Du, W., & Zhan, Z. (2003). Using randomized response techniques for privacy-preserving data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 505–510). ACM. doi: 10.1145/956750.956810

Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (pp. 1–19). doi: 10.1007/978-3-540-79228-4_1

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2017). Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, *7*(3), 2. doi: 10.29012/jpc.v7i3.405

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, *9*(3–4), 211–407. doi: 10.1561/0400000042

Ebadi, H., Antignac, T., & Sands, D. (2016). Sampling and partitioning for differential privacy. In *Privacy, Security and Trust (PST), 2016 14th Annual Conference on* (pp. 664–673). doi: 10.1109/PST.2016.7906954

El Emam, K., & Dankar, F. K. (2008). Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, *15*(5), 627–637. doi: 10.1197/jamia.M2716

Erlingsson, U., Pihur, V., & Korolova, A. (2014). RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1054–1067). ACM. doi: 10.1145/2660267.2660348

EUGDPR.org. (2016). GDPR Key Changes. *EUGDPR.org*. Retrieved from `https://www.eugdpr.org/key-changes.html`. (online; accessed 2018-04-10)

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining.* Springer. doi: 10.1007/978-3-319-10247-4

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V., & Herrera, F. (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 734–750. doi: 10.1109/TKDE.2012.35

Golle, P. (2006). Revisiting the uniqueness of simple demographics in the US population. In *Proceedings of the 5th ACM workshop on Privacy in* *electronic society* (pp. 77–80). doi: 10.1145/1179601.1179615

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, *11*(1), 63–90. doi: 10.1023/A:1022631118932

Hooker, J. N. (1994). Needed: An empirical science of algorithms. *Operations Research*, *42*(2), 201–212. doi: 10.1287/opre.42.2.201

Ingram, D. (2018, April 4). Facebook says data leak hits 87 million users, widening privacy scandal. *Reuters*. Retrieved from `https://www.reuters.com/article/us-facebook-privacy/facebook-says-data-leak-hits-87-million-users-widening-privacy-scandal-idUSKCN1HB2CM`. (online; accessed 2018-04-10)

Joy, J., & Gerla, M. (2017). Differential Privacy By Sampling. *arXiv preprint arXiv:1708.01884*.

Kaplan, D. T. (1999). Resampling Stats in MATLAB. *Arlington: Resampling Stats*.

Katal, A., Wazid, M., & Goudar, R. (2013). Big data: issues, challenges, tools and good practices. In *Contemporary Computing (IC3), 2013 Sixth International Conference on* (pp. 404–409). doi: 10.1109/IC3.2013.6612229

Kerber, R. (1992). Chimerge: Discretization of numeric attributes. In *Proceedings of the tenth national conference on Artificial intelligence* (pp. 123–128).

Kohavi, R. (1995). A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (pp. 1137–1143). Morgan Kaufmann Publishers Inc.

Kotsiantis, S., & Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, *32*(1), 47–58.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science*, *1*(2), 111–117.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms.* John Wiley & Sons.

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (pp. 106–115). doi: 10.1109/ICDE.2007.367856

Li, N., Qardaji, W., & Su, D. (2012). On Sampling, Anonymization, and Differential Privacy or, K-anonymization Meets Differential Privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security* (pp. 32–33). ACM. doi: 10.1145/2414456.2414474

Lin, B.-R., Wang, Y., & Rane, S. (2013). On the benefits of sampling in privacy preserving statistical analysis on distributed databases. *arXiv preprint arXiv:1304.4613*.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data mining and knowledge discovery*, *6*(4), 393–423.

doi: 10.1023/A:1016304305535

Liu, H., & Setiono, R. (1997). Feature selection via discretization. *IEEE Transactions on knowledge and Data Engineering*, *9*(4), 642–645. doi: 10.1109/69.617056

Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006). l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on* (p. 24). doi: 10.1109/ICDE.2006.1

Malin, B., & Sweeney, L. (2004). How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of biomedical informatics*, *37*(3), 179–192. doi: 10.1016/j.jbi.2004.04.005

McSherry, F. (2016a, February 3). Differential privacy for dummies. *Github*. Retrieved from `https://github.com/frankmcsherry/blog/blob/master/posts/2016-02-03.md`. (online; accessed 2018-04-17)

McSherry, F. (2016b, May 19). Differential privacy for dummies, redux. *Github*. Retrieved from `https://github.com/frankmcsherry/blog/blob/master/posts/2016-05-19.md`. (online; accessed 2018-04-17)

Mehta, S., Parthasarathy, S., & Yang, H. (2005). Toward unsupervised correlation preserving discretization. *IEEE Transactions on Knowledge and Data Engineering*, *17*(9), 1174–1185. doi: 10.1109/TKDE.2005.153

Mendes, R., & Vilela, J. P. (2017). Privacy-Preserving Data Mining: Methods, Metrics, and Applications. *IEEE Access*, *5*, 10562–10582. doi: 10.1109/ACCESS.2017.2706947

Moisen, G. G. (2008). Classification and regression trees. *In: Jørgensen, Sven Erik; Fath, Brian D.(Editor-in-Chief). Encyclopedia of Ecology, volume 1. Oxford, UK: Elsevier. p. 582-588.*, 582–588.

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on* (pp. 111–125). doi: 10.1109/SP.2008.33

Neumann, D. L., Hood, M., & Neumann, M. M. (2013). Using real-life data when teaching statistics: student perceptions of this strategy in an introductory statistics course. *Statistics Education Research Journal*, *12*(2).

Nguyen, H.-V., Müller, E., Vreeken, J., & Böhm, K. (2014). Unsupervised interaction-preserving discretization of multivariate data. *Data Mining and Knowledge Discovery*, *28*(5-6), 1366–1397. doi: 10.1007/s10618-014-0350-5

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, *34*(2), 133–143. doi: 10.1007/s10462-010-9165-y

Prasser, F., & Kohlmayer, F. (2015). Putting Statistical Disclosure Control into Practice: The ARX Data Anonymization Tool. In A. Gkoulalas-Divanis & G. Loukides (Eds.), *Medical Data Privacy Handbook* (pp. 111–148). Cham: Springer International Publishing. doi: 10.1007/978-3-319-23633-9_6

Ram, P., & Gray, A. G. (2011). Density estimation trees. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery* *and data mining* (pp. 627–635).

Ramírez-Gallego, S., García, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Alonso-Betanzos, A., … Herrera, F. (2016). Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(1), 5–21. doi: 10.1002/widm.1173

Rebollo-Monedero, D., Parra-Arnau, J., Diaz, C., & Forné, J. (2013, apr). On the measurement of privacy as an attacker's estimation error. *International Journal of Information Security*, *12*(2), 129–149. Retrieved from `https://doi.org/10.1007/s10207-012-0182-5` doi: 10.1007/s10207-012-0182-5

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). (2016, May 4). *Official Journal of the European Union*, *L119*, 1–88.

Reuters. (2018, May 2). Cambridge Analytica and British parent shut down after Facebook scandal. *Reuters*. Retrieved from `https://www.reuters.com/article/us-faceboook-privacy/cambridge-analytica-shutting-down-wsj-idUSKBN1I32L7`. (online; accessed 2018-05-07)

Ribeiro, M. X., Ferreira, M. R. P., Traina Jr., C., & Traina, A. J. M. (2008). Data Pre-processing: A New Algorithm for Feature Selection and Data Discretization. In *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology* (pp. 252–257). New York, NY, USA: ACM. doi: 10.1145/1456223.1456277

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465–471. doi: 10.1016/0005-1098(78)90005-5

Samarati, P. (2001). Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, *13*(6), 1010–1027. doi: 10.1109/69.971193

Sarathy, R., & Muralidhar, K. (2010). Some additional insights on applying differential privacy for numeric data. In *International Conference on Privacy in Statistical Databases* (pp. 210–219). doi: 10.1007/978-3-642-15838-4_19

Schmidberger, G., & Frank, E. (2005). Unsupervised discretization using tree-based density estimation. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 240–251). doi: 10.1007/11564126_26

Schoeman, F. (1984). Privacy: philosophical dimensions. *American Philosophical Quarterly*, *21*(3), 199–213.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570. doi: 10.1142/S0218488502001648

Tang, J., Korolova, A., Bai, X., Wang, X., & Wang, X. (2017). Privacy Loss in Apple's Implementation of Differential Privacy on macOS 10.12. *arXiv* *preprint arXiv:1709.02753*.

Tichy, W. F., Lukowicz, P., Prechelt, L., & Heinz, E. A. (1995). Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*, *28*(1), 9–18. doi: 10.1016/0164-1212(94)00111-Y

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, *60*(309), 63–69. doi: 10.1080/01621459.1965.10480775

Wei, H. (2009). A novel multivariate discretization method for mining association rules. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on* (Vol. 1, pp. 378–381). doi: 10.1109/APCIP.2009.102

Wes, M. (2017, April 25). Looking to comply with GDPR? Here's a primer on anonymization and pseudonymization. *International Association of Privacy Professionals*. Retrieved from `https://iapp.org/news/a/looking-to-comply-with-gdpr-heres-a-primer-on-anonymization-and-pseudonymization/`. (online; accessed 2018-05-08)

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering.* Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-662-43839-8_2

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

Wixom, B., Ariyachandra, T., Goul, M., Gray, P., Kulkarni, U. R., & Phillips-Wren, G. E. (2011). The current state of business intelligence in academia. *CAIS*, *29*, 16.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering.* Springer Science & Business Media.

Wu, X., Zhu, X., Wu, G.-Q., & Ding, W. (2014). Data mining with big data. *IEEE transactions on knowledge and data engineering*, *26*(1), 97–107. doi: 10.1109/TKDE.2013.109

Appendices

Appendix A

Experiment results

A.1 Breast

Table A.1: Classification performance breast

k	Accuracy	F1	Recall	Precision	AUC	Bins	Time (s)
5	0.95	0.94	0.94	0.94	0.94	72	0.42
10	0.94	0.93	0.93	0.94	0.93	40	0.33
15	**0.95**	0.94	0.94	0.95	0.94	27	0.28
20	0.94	0.94	0.93	0.94	0.93	21	0.25
25	0.94	0.93	0.93	0.94	0.93	17	0.25
30	0.94	0.93	0.92	0.94	0.92	15	0.22
35	0.94	0.93	0.92	0.94	0.92	13	0.21
40	0.94	0.93	0.93	0.94	0.93	10	0.18
45	0.94	0.93	0.93	0.93	0.93	9	0.15
50	0.94	0.93	0.94	0.93	0.94	8	0.14
55	0.94	0.93	0.94	0.93	0.94	7	0.15
60	0.94	0.93	0.94	0.92	0.94	7	0.14
65	0.94	0.93	0.94	0.93	0.94	7	0.14
70	0.93	0.93	0.93	0.93	0.93	7	0.14
75	0.94	0.93	0.93	0.94	0.93	6	0.13
80	0.94	0.94	0.94	0.94	0.94	6	0.13
85	0.94	0.93	0.93	0.94	0.93	6	0.13
90	0.93	0.92	0.92	0.93	0.92	6	0.13
95	0.93	0.92	0.92	0.92	0.92	5	0.12
100	0.85	0.84	0.86	0.86	0.86	4	0.11

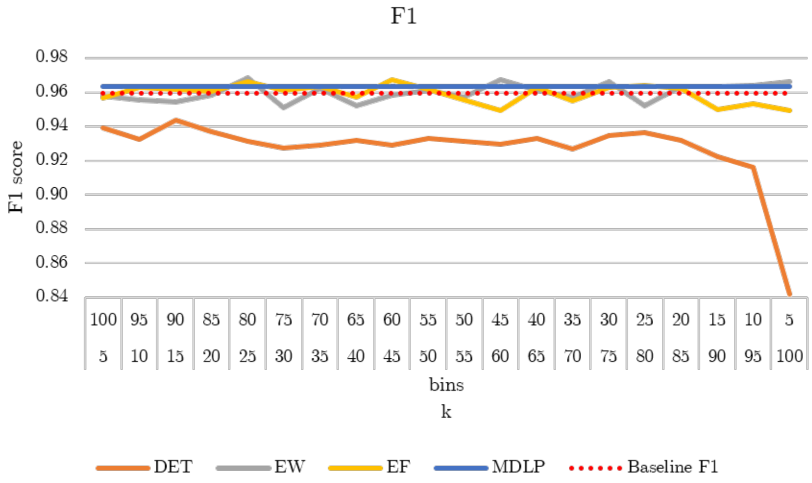


Figure A.1: F1 comparison breast

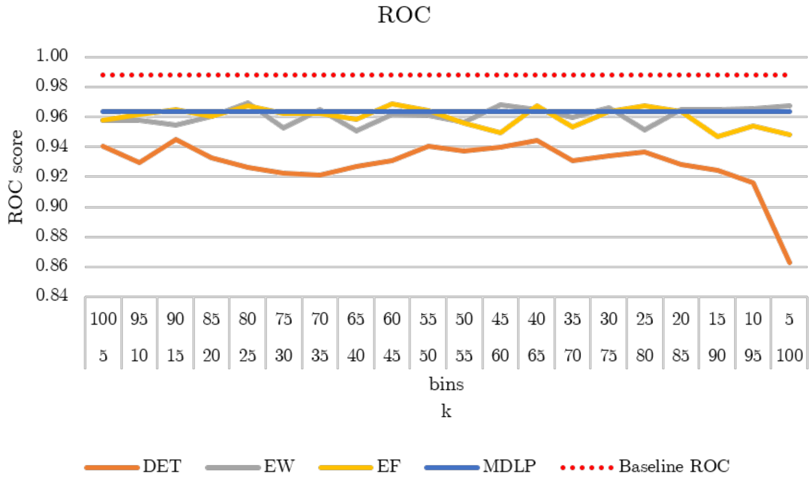


Figure A.2: ROC comparison breast

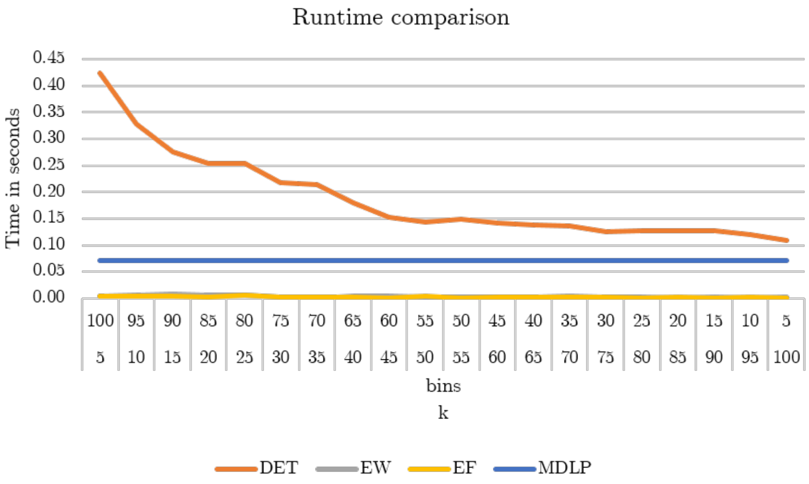


Figure A.3: Runtime comparison breast

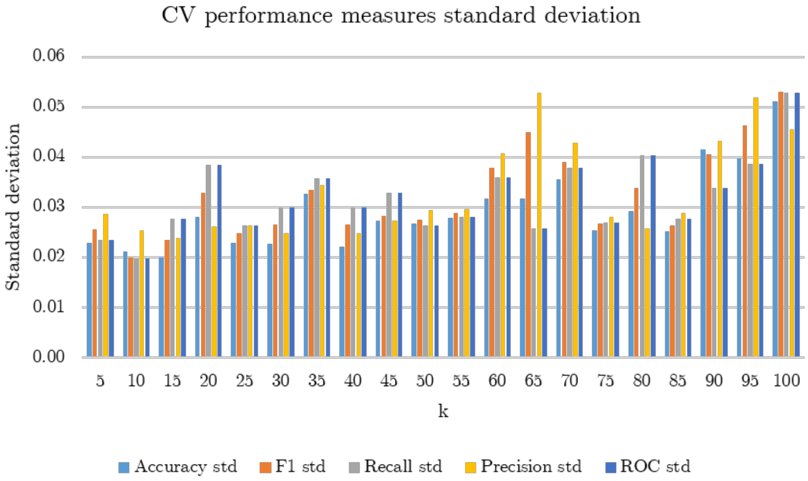


Figure A.4: Standard deviations breast

A.2 Glass

Table A.2: Classification performance glass

k	**Accuracy**	**F1**	**Recall**	**Precision**	**Bins**	**Time (s)**
5	**0.58**	0.47	0.47	0.52	33	1.34
10	0.55	0.46	0.49	0.49	16	0.84
15	0.45	0.26	0.30	0.25	10	0.70
20	0.45	0.27	0.32	0.27	8	0.60
25	0.49	0.35	0.43	0.34	7	0.56
30	0.42	0.27	0.33	0.26	5	0.47
35	0.42	0.25	0.34	0.24	4	0.45
40	0.44	0.29	0.37	0.27	4	0.44
45	0.40	0.26	0.36	0.23	4	0.43
50	0.40	0.21	0.34	0.17	3	0.38
55	0.41	0.24	0.35	0.20	3	0.36
60	0.42	0.20	0.27	0.17	2	0.31
65	0.39	0.19	0.24	0.16	2	0.31
70	0.39	0.17	0.23	0.15	2	0.30
75	0.41	0.19	0.27	0.16	2	0.30
80	0.32	0.16	0.27	0.13	2	0.31
85	0.36	0.16	0.25	0.13	2	0.29
90	0.34	0.17	0.29	0.13	2	0.30
95	0.38	0.19	0.26	0.16	2	0.29
100	0.36	0.10	0.19	0.07	1	0.17

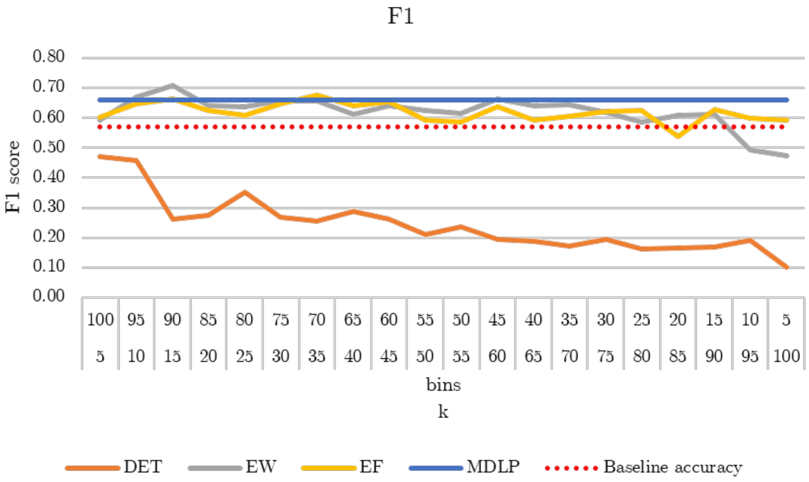


Figure A.5: F1 comparison glass

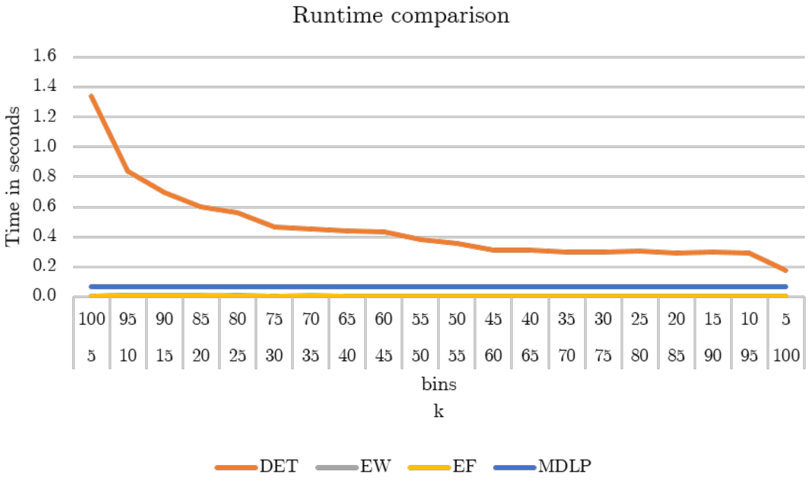


Figure A.6: Runtime comparison glass

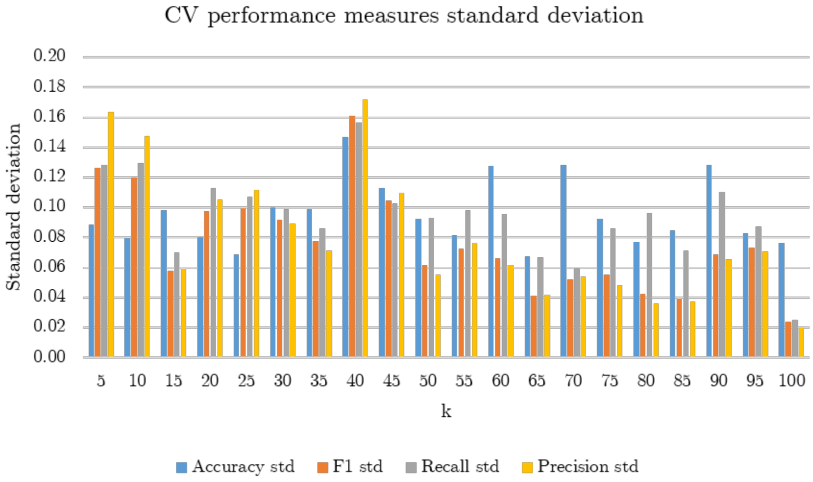


Figure A.7: Standard deviations glass

A.3 Iris

Table A.3: Classification performance iris

k	Accuracy	F1	Recall	Precision	Bins	Time (s)
5	**0.95**	0.94	0.95	0.95	22	0.07
10	0.93	0.93	0.92	0.95	10	0.05
15	0.91	0.89	0.90	0.91	7	0.04
20	0.87	0.86	0.87	0.88	5	0.03
25	0.85	0.81	0.82	0.84	4	0.03
30	0.83	0.81	0.83	0.86	4	0.03
35	0.91	0.90	0.90	0.93	3	0.02
40	0.93	0.93	0.93	0.95	3	0.02
45	0.67	0.62	0.72	0.59	2	0.02
50	0.67	0.54	0.67	0.47	2	0.02
55	0.67	0.54	0.67	0.46	2	0.02
60	0.67	0.52	0.67	0.44	2	0.02
65	0.67	0.53	0.67	0.45	2	0.02
70	0.25	0.13	0.33	0.08	1	0.01
75	0.23	0.12	0.33	0.08	1	0.01
80	0.24	0.13	0.33	0.08	1	0.01
85	0.26	0.14	0.33	0.09	1	0.01
90	0.21	0.11	0.33	0.07	1	0.01
95	0.18	0.10	0.33	0.06	1	0.01
100	0.25	0.13	0.33	0.08	1	0.01

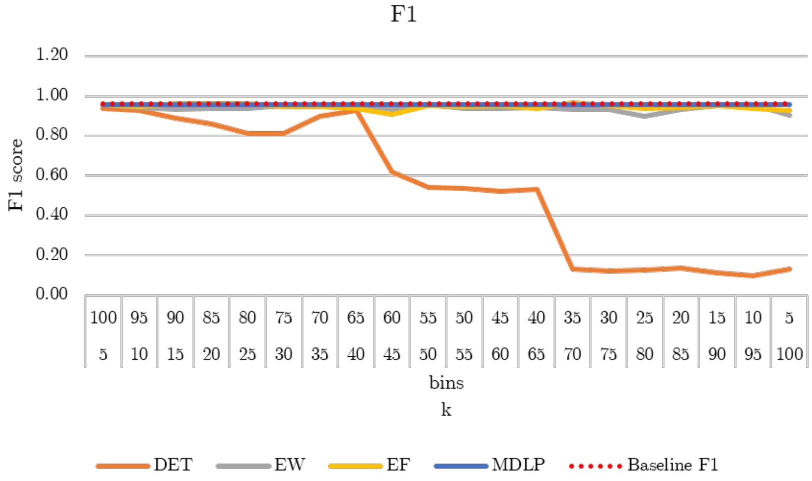


Figure A.8: F1 comparison iris

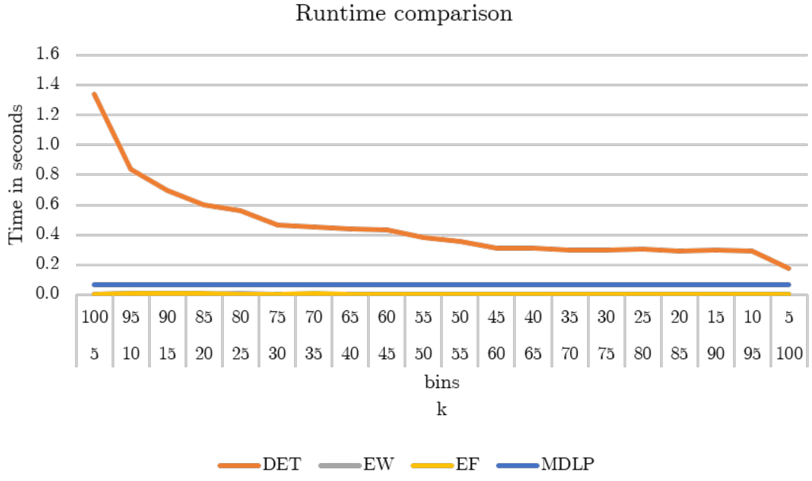


Figure A.9: Runtime comparison iris

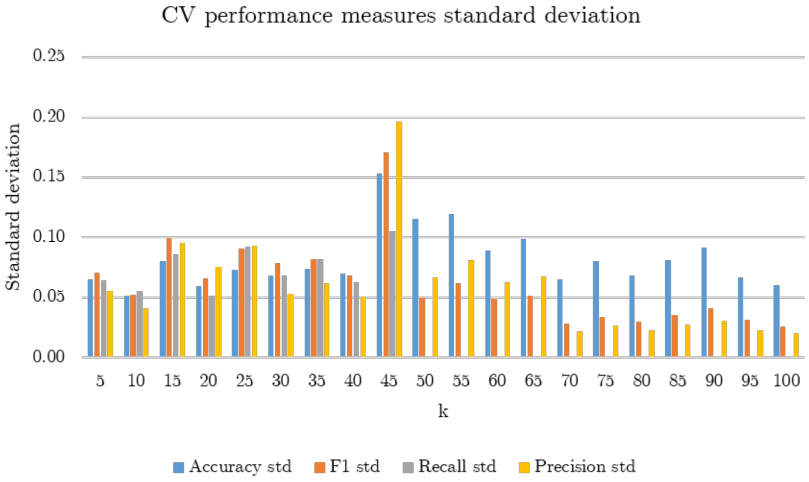


Figure A.10: Standard deviations iris

A.4 Wine

Table A.4: Classification performance wine

k	Accuracy	F1	Recall	Precision	Bins	Time (s)
5	**0.90**	0.89	0.90	0.90	31	4.33
10	0.85	0.84	0.86	0.87	15	2.21
15	0.78	0.77	0.78	0.80	10	1.56
20	0.84	0.85	0.85	0.88	7	1.21
25	0.78	0.78	0.80	0.79	6	1.04
30	0.80	0.80	0.79	0.86	5	0.87
35	0.79	0.76	0.75	0.83	4	0.79
40	0.74	0.72	0.73	0.75	3	0.69
45	0.77	0.76	0.76	0.78	3	0.67
50	0.78	0.77	0.79	0.77	3	0.66
55	0.55	0.45	0.59	0.38	2	0.53
60	0.59	0.47	0.63	0.38	2	0.52
65	0.57	0.47	0.63	0.38	2	0.51
70	0.56	0.48	0.62	0.40	2	0.51
75	0.57	0.47	0.64	0.38	2	0.51
80	0.59	0.49	0.66	0.40	2	0.51
85	0.40	0.19	0.33	0.13	1	0.30
90	0.40	0.19	0.33	0.13	1	0.29
95	0.40	0.19	0.33	0.13	1	0.29
100	0.40	0.19	0.33	0.13	1	0.29

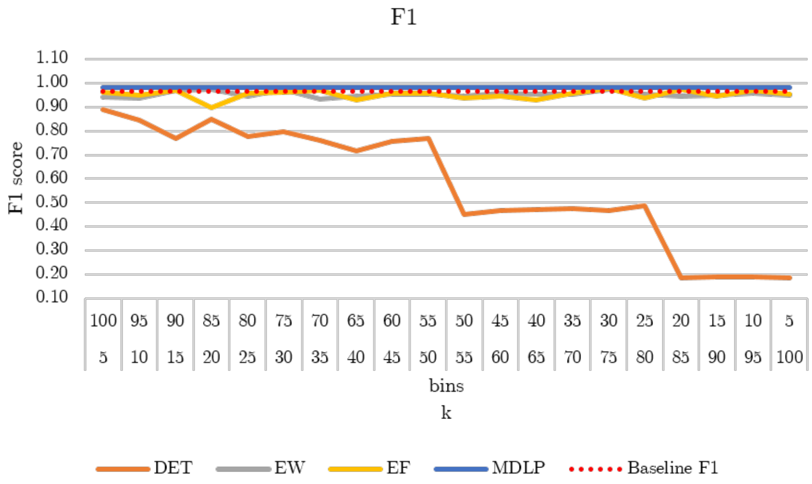


Figure A.11: F1 comparison wine

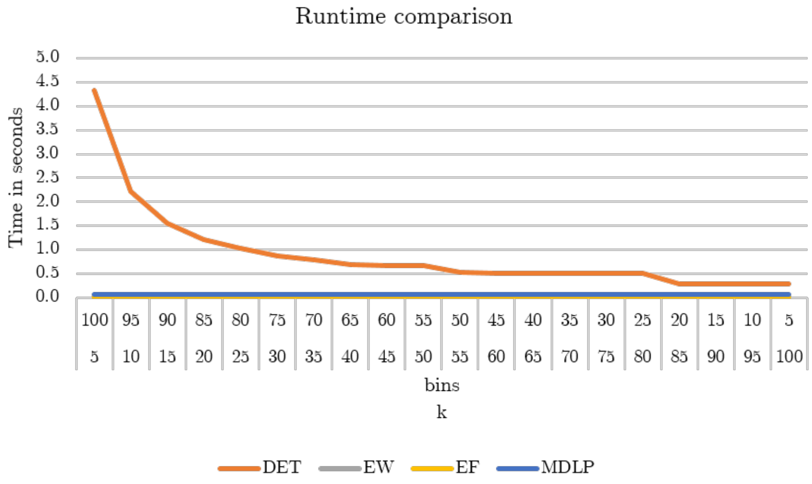


Figure A.12: Runtime comparison wine

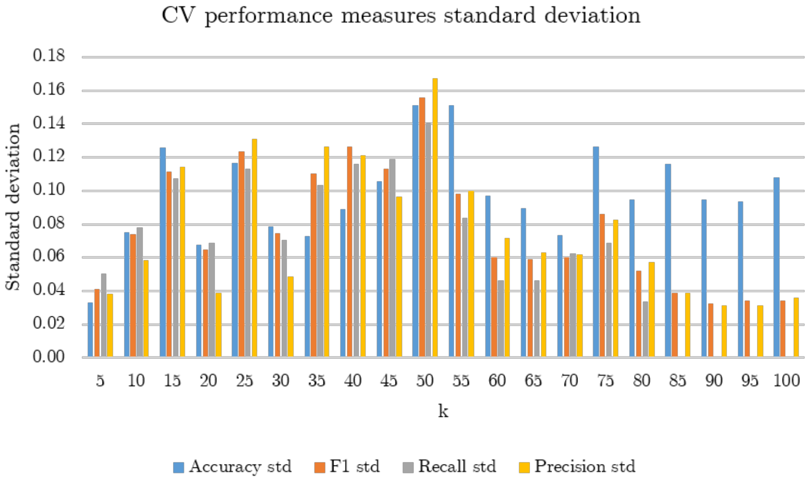


Figure A.13: Standard deviations wine

A.5 Iris + 1

Table A.5: Classification performance iris + 1

k	**Accuracy**	**F1**	**Recall**	**Precision**	**Bins**	**Time (s)**
5	0.90	0.89	0.90	0.90	22	0.17
10	0.93	0.91	0.92	0.95	10	0.10
15	0.93	0.93	0.94	0.94	7	0.08
20	0.91	0.89	0.89	0.91	5	0.07
25	0.83	0.81	0.83	0.85	4	0.07
30	0.85	0.81	0.83	0.85	4	0.06
35	0.90	0.88	0.89	0.92	3	0.06
40	**0.94**	0.94	0.94	0.96	3	0.05
45	0.81	0.79	0.82	0.80	3	0.05
50	0.67	0.57	0.70	0.51	2	0.04
55	0.67	0.53	0.67	0.46	2	0.04
60	0.67	0.52	0.67	0.44	2	0.04
65	0.66	0.51	0.66	0.45	2	0.04
70	0.30	0.15	0.33	0.10	1	0.02
75	0.21	0.12	0.33	0.07	1	0.02
80	0.25	0.13	0.33	0.08	1	0.02
85	0.23	0.12	0.33	0.08	1	0.02
90	0.24	0.13	0.33	0.08	1	0.02
95	0.26	0.14	0.33	0.09	1	0.02
100	0.24	0.13	0.33	0.08	1	0.02

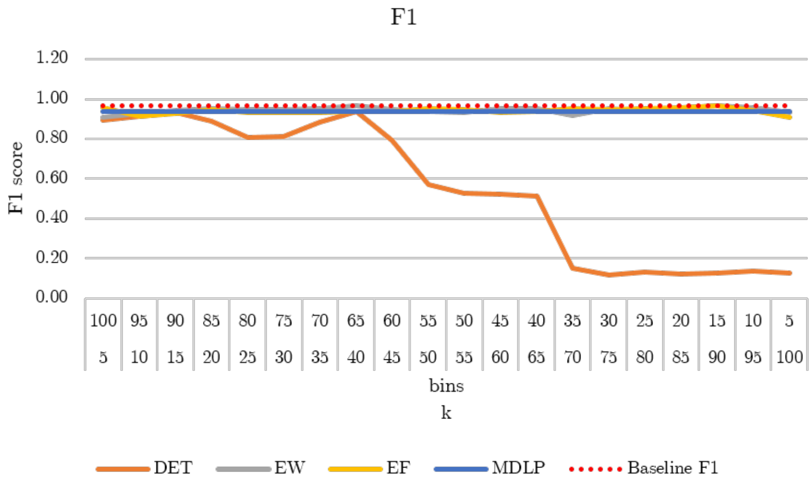


Figure A.14: F1 comparison iris + 1

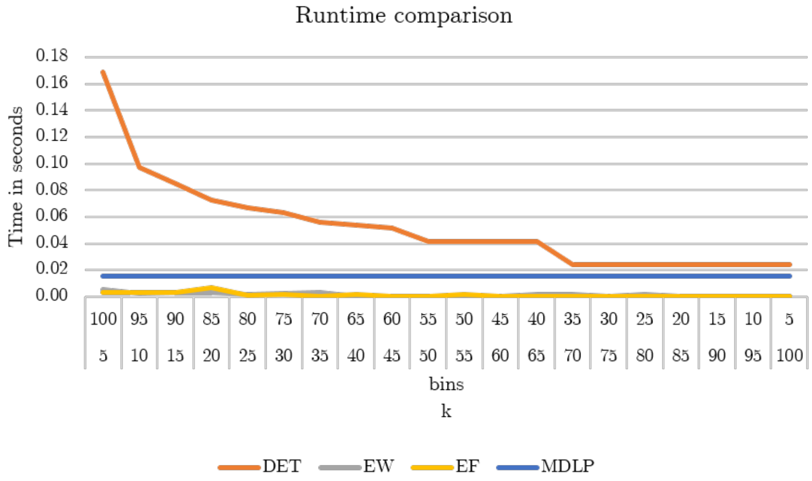


Figure A.15: Runtime comparison iris + 1

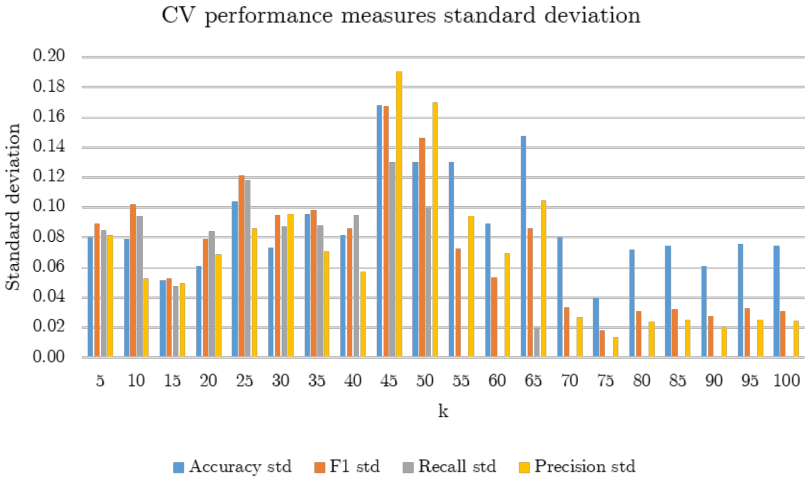


Figure A.16: Standard deviations iris + 1

A.6 Iris + 5

Table A.6: Classification performance iris + 5

k	Accuracy	F1	Recall	Precision	Bins	Time (s)
5	0.83	0.81	0.81	0.82	24	1.19
10	0.86	0.85	0.86	0.88	12	0.59
15	0.84	0.84	0.86	0.88	7	0.41
20	0.81	0.77	0.82	0.81	5	0.35
25	0.75	0.74	0.77	0.80	5	0.33
30	0.81	0.78	0.79	0.84	4	0.31
35	0.89	0.88	0.88	0.92	3	0.27
40	**0.92**	0.91	0.91	0.93	3	0.27
45	0.71	0.70	0.72	0.70	3	0.25
50	0.67	0.54	0.67	0.49	2	0.21
55	0.67	0.53	0.67	0.46	2	0.20
60	0.67	0.53	0.67	0.44	2	0.21
65	0.65	0.49	0.62	0.41	2	0.20
70	0.22	0.12	0.33	0.07	1	0.12
75	0.30	0.15	0.33	0.10	1	0.12
80	0.22	0.12	0.30	0.07	1	0.12
85	0.21	0.11	0.33	0.07	1	0.12
90	0.24	0.13	0.33	0.08	1	0.12
95	0.25	0.13	0.33	0.08	1	0.12
100	0.25	0.13	0.33	0.08	1	0.12

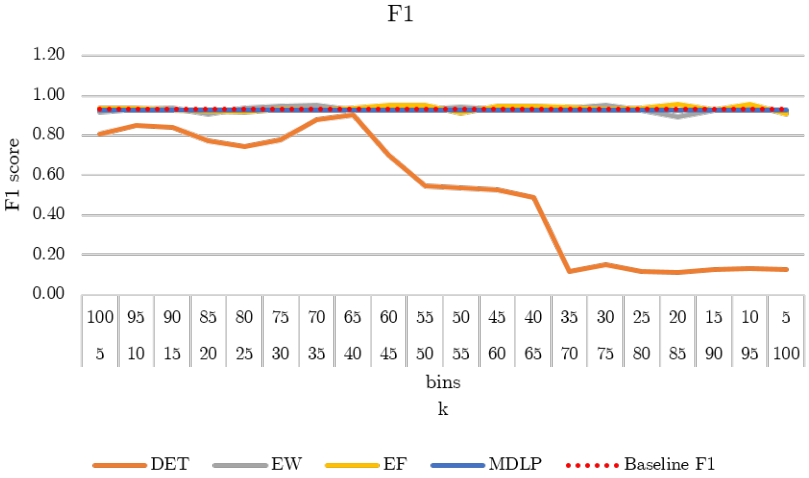


Figure A.17: F1 comparison iris + 5

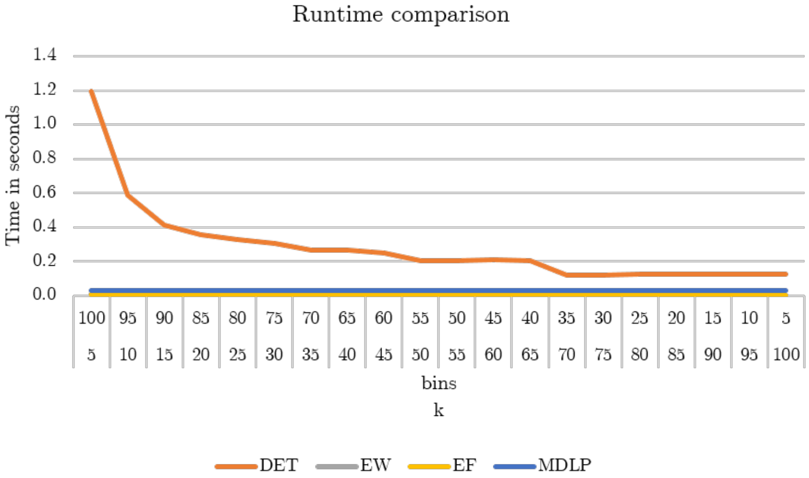


Figure A.18: Runtime comparison iris + 5

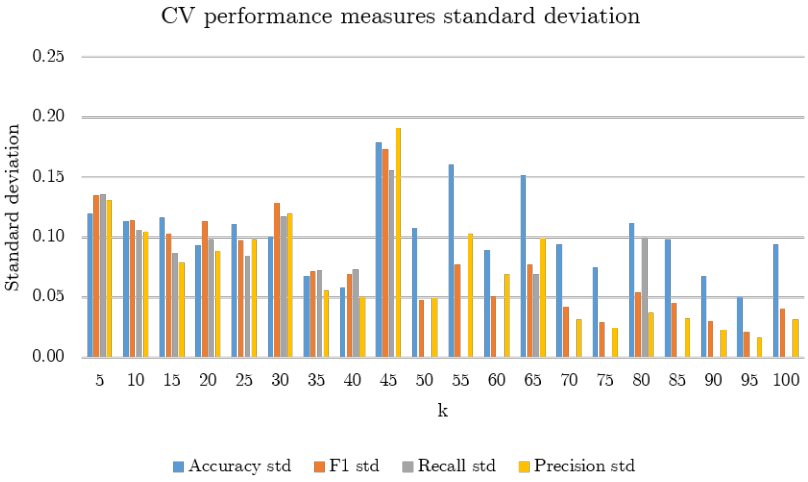


Figure A.19: Standard deviations iris + 5

A.7 Iris + 10

Table A.7: Classification performance iris + 10

k	**Accuracy**	**F1**	**Recall**	**Precision**	**Bins**	**Time (s)**
5	0.82	0.80	0.82	0.85	25	3.71
10	0.79	0.76	0.78	0.79	12	1.92
15	0.81	0.80	0.82	0.82	8	1.31
20	0.81	0.78	0.79	0.82	5	1.08
25	0.71	0.69	0.73	0.77	5	0.96
30	0.67	0.67	0.70	0.76	4	0.85
35	0.75	0.71	0.73	0.75	3	0.72
40	**0.94**	0.93	0.95	0.93	3	0.69
45	0.68	0.62	0.70	0.58	3	0.63
50	0.64	0.52	0.64	0.46	2	0.57
55	0.67	0.54	0.67	0.47	2	0.55
60	0.67	0.56	0.70	0.48	2	0.55
65	0.67	0.52	0.67	0.44	2	0.56
70	0.23	0.12	0.33	0.08	1	0.34
75	0.19	0.10	0.33	0.06	1	0.34
80	0.18	0.10	0.33	0.06	1	0.34
85	0.23	0.12	0.33	0.08	1	0.34
90	0.24	0.12	0.30	0.08	1	0.35
95	0.27	0.14	0.33	0.09	1	0.34
100	0.27	0.14	0.33	0.09	1	0.33

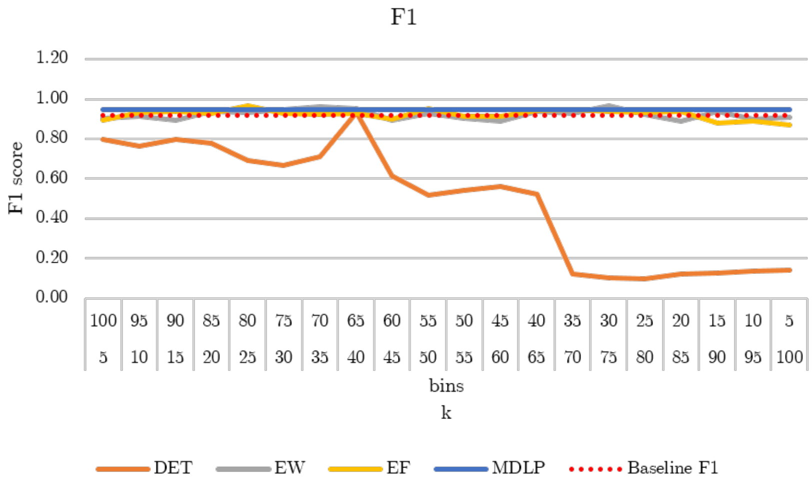


Figure A.20: F1 comparison iris + 10

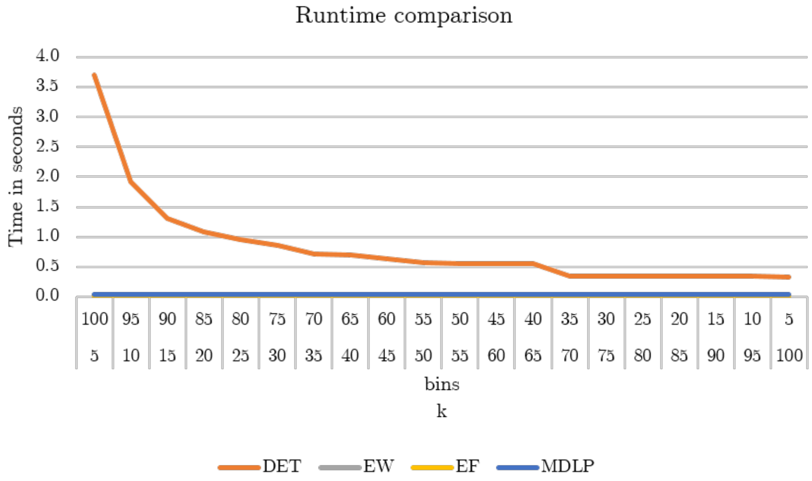


Figure A.21: Runtime comparison iris + 10

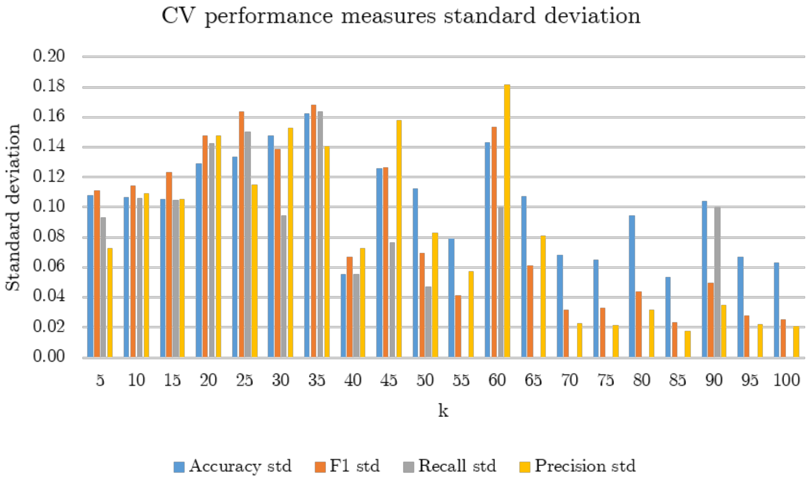


Figure A.22: Standard deviations iris + 10

A.8 Synthetic 1

Table A.8: Classification performance synthetic 1

k	**Accuracy**	**F1**	**Recall**	**Precision**	**AUC**	**Bins**	**Time (s)**
5	0.74	0.74	0.74	0.74	0.74	121	6.80
10	0.72	0.72	0.72	0.72	0.72	58	4.28
15	0.76	0.76	0.76	0.76	0.76	37	3.87
20	0.76	0.76	0.76	0.76	0.76	29	3.59
25	0.69	0.69	0.69	0.69	0.69	22	3.48
30	**0.77**	0.77	0.77	0.77	0.77	19	3.41
35	0.76	0.76	0.76	0.76	0.76	15	3.32
40	**0.77**	0.77	0.77	0.77	0.77	14	3.21
45	0.76	0.76	0.76	0.76	0.76	12	3.14
50	0.76	0.76	0.76	0.76	0.76	11	3.09
55	0.66	0.66	0.66	0.66	0.66	11	3.00
60	0.64	0.64	0.64	0.64	0.64	8	2.85
65	0.66	0.65	0.67	0.65	0.67	7	2.77
70	0.66	0.65	0.67	0.65	0.67	7	2.80
75	0.66	0.65	0.67	0.65	0.67	7	2.77
80	0.55	0.55	0.55	0.55	0.55	7	2.75
85	0.61	0.61	0.61	0.61	0.61	7	2.78
90	0.61	0.61	0.61	0.61	0.61	7	2.75
95	0.61	0.61	0.61	0.61	0.61	6	2.73
100	0.56	0.55	0.58	0.57	0.58	5	2.58

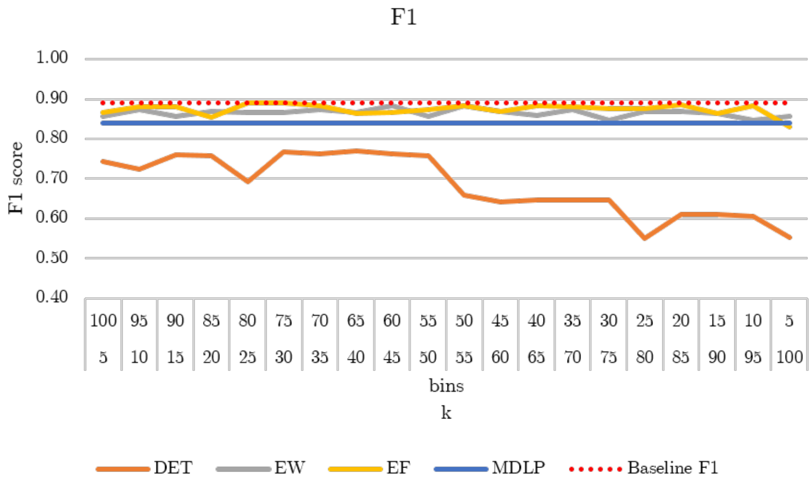


Figure A.23: F1 comparison synthetic 1

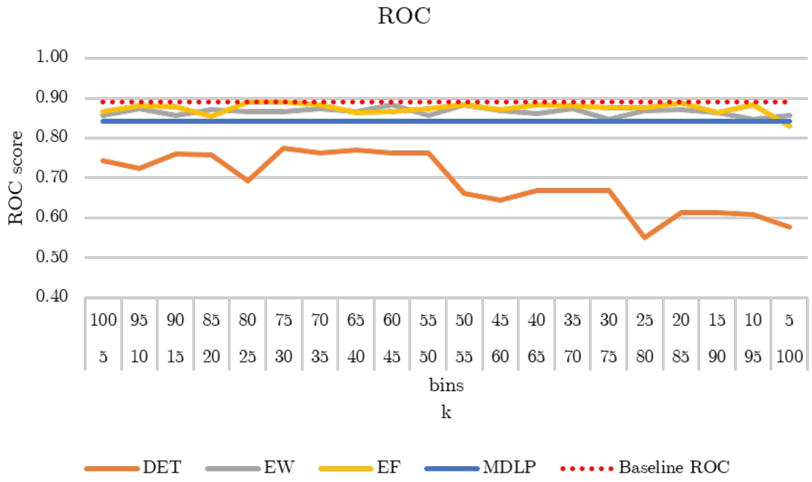


Figure A.24: ROC comparison synthetic 1

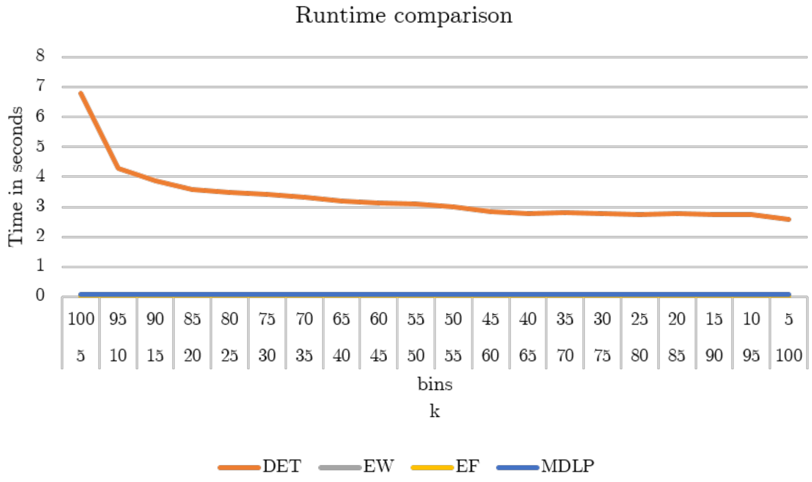


Figure A.25: Runtime comparison synthetic 1

A.9 Synthetic 2

Table A.9: Classification performance synthetic 2

k	Accuracy	F1	Recall	Precision	AUC	Bins	Time (s)
5	0.54	0.54	0.55	0.55	0.55	136	225.89
10	**0.67**	0.67	0.69	0.68	0.69	65	105.16
15	0.57	0.57	0.59	0.58	0.59	41	34.51
20	0.50	0.50	0.50	0.50	0.50	29	24.61
25	0.56	0.56	0.56	0.56	0.56	25	22.17
30	0.51	0.51	0.52	0.52	0.52	20	17.95
35	0.47	0.46	0.48	0.48	0.48	16	13.53
40	0.48	0.47	0.48	0.48	0.48	15	13.13
45	0.50	0.47	0.52	0.52	0.52	13	12.68
50	0.50	0.46	0.53	0.52	0.53	12	11.57
55	0.44	0.43	0.44	0.45	0.44	11	11.15
60	0.51	0.50	0.52	0.52	0.52	9	10.84
65	0.51	0.50	0.53	0.52	0.53	9	10.62
70	0.48	0.48	0.48	0.48	0.48	7	10.10
75	0.48	0.48	0.48	0.48	0.48	7	10.05
80	0.48	0.48	0.48	0.48	0.48	7	10.01
85	0.47	0.44	0.48	0.49	0.48	7	9.86
90	0.48	0.42	0.50	0.50	0.50	6	9.52
95	0.48	0.42	0.50	0.50	0.50	6	9.52
100	0.48	0.42	0.50	0.50	0.50	6	9.48

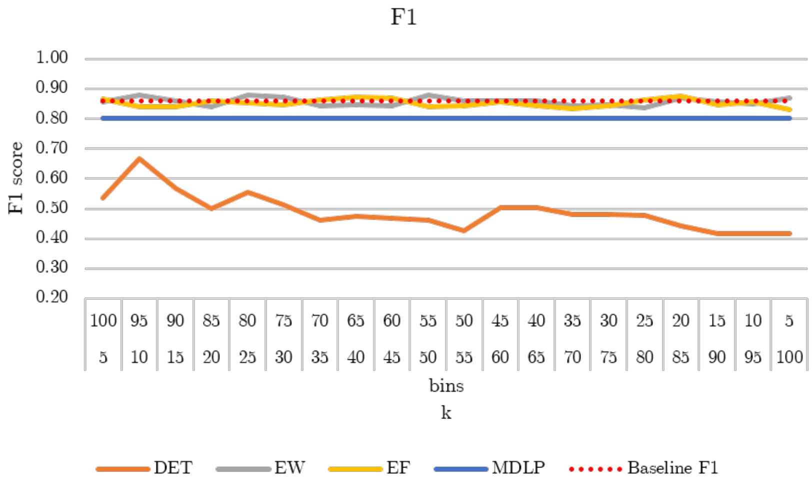


Figure A.26: F1 comparison synthetic 2

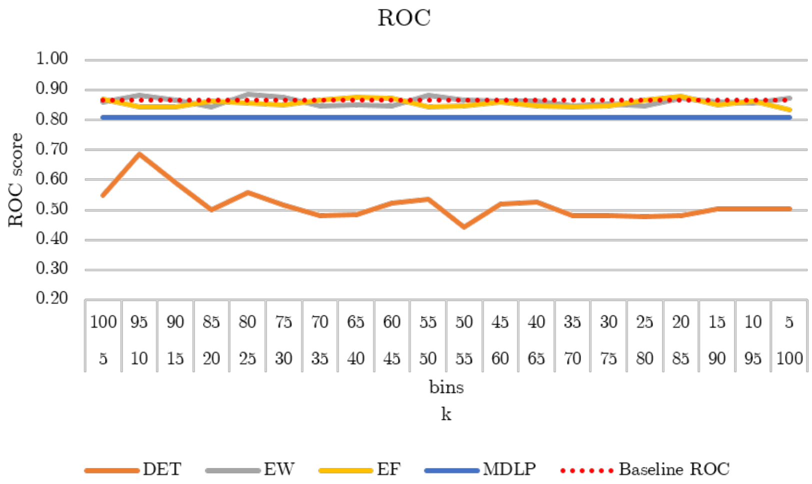


Figure A.27: ROC comparison synthetic 2

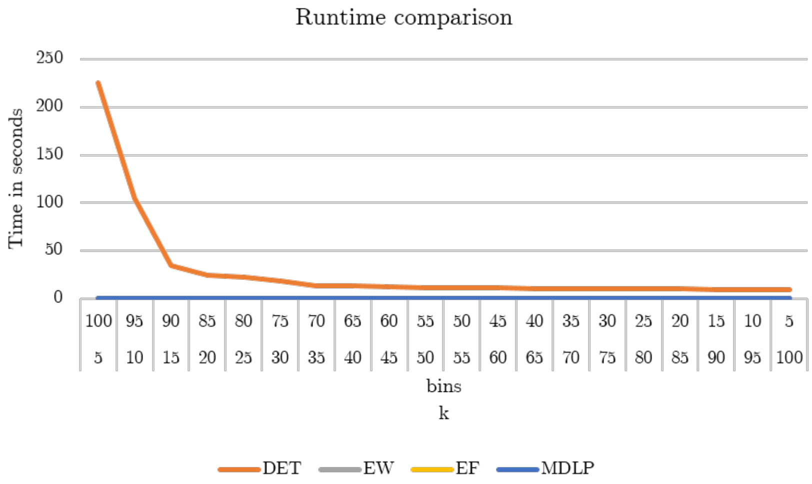


Figure A.28: Runtime comparison synthetic 2

A.10 Synthetic 3

Table A.10: Classification performance synthetic 3

k	Accuracy	F1	Recall	Precision	AUC	Bins	Time (s)
5	**0.94**	0.94	0.94	0.94	0.94	1157	105.98
10	0.93	0.93	0.93	0.93	0.93	547	100.86
15	0.93	0.92	0.92	0.93	0.92	362	97.66
20	0.93	0.93	0.93	0.93	0.93	275	93.97
25	0.93	0.93	0.93	0.93	0.93	219	98.53
30	0.93	0.93	0.93	0.93	0.93	181	93.17
35	0.92	0.92	0.92	0.92	0.92	155	92.99
40	0.93	0.93	0.93	0.93	0.93	134	92.01
45	0.93	0.93	0.93	0.93	0.93	125	92.28
50	0.92	0.92	0.92	0.92	0.92	103	91.75
55	0.92	0.92	0.92	0.92	0.92	100	91.40
60	0.92	0.92	0.92	0.92	0.92	91	90.44
65	0.91	0.91	0.91	0.91	0.91	84	89.72
70	0.90	0.90	0.90	0.90	0.90	77	91.10
75	0.90	0.90	0.90	0.90	0.90	73	89.35
80	0.90	0.90	0.90	0.90	0.90	67	88.46
85	0.90	0.90	0.90	0.90	0.90	67	87.77
90	0.90	0.90	0.90	0.90	0.90	64	88.42
95	0.90	0.90	0.90	0.90	0.90	58	89.42
100	0.89	0.89	0.90	0.89	0.90	53	87.08

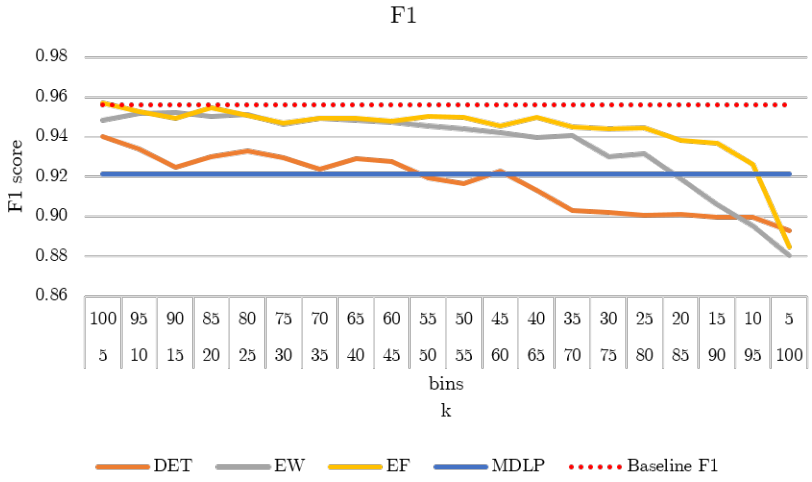


Figure A.29: F1 comparison synthetic 3

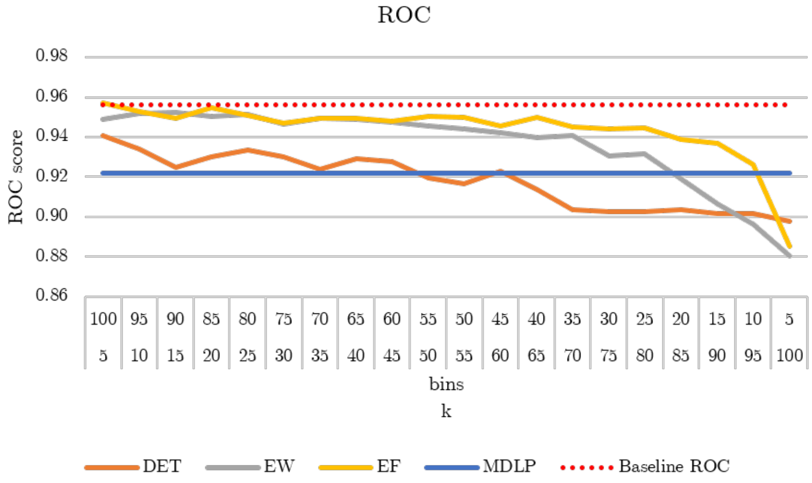


Figure A.30: ROC comparison synthetic 3

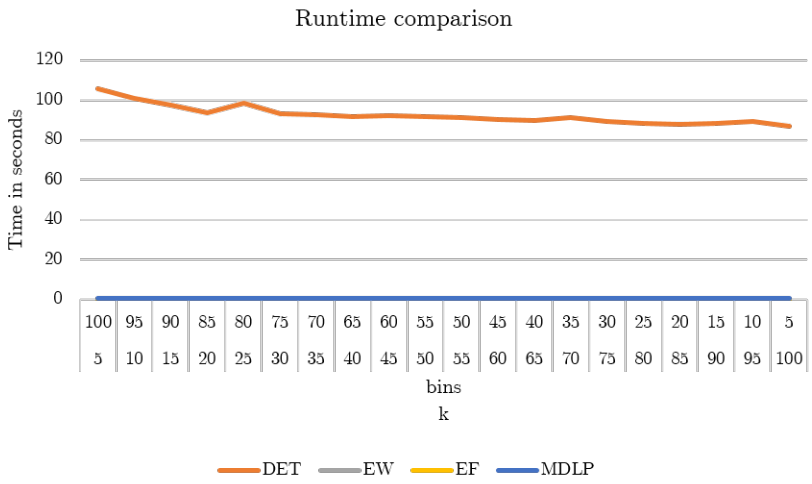


Figure A.31: Runtime comparison synthetic 3

A.11 Synthetic 4

Table A.11: Classification performance synthetic 4

k	**Accuracy**	**F1**	**Recall**	**Precision**	**AUC**	**Bins**	**Time (s)**
5	0.94	0.94	0.94	0.94	0.94	2299	345.22
10	0.94	0.94	0.94	0.94	0.94	1098	339.30
15	**0.94**	0.94	0.94	0.94	0.94	725	339.65
20	0.94	0.94	0.94	0.94	0.94	543	342.55
25	0.94	0.94	0.94	0.94	0.94	434	354.39
30	0.94	0.94	0.94	0.94	0.94	355	355.46
35	0.93	0.93	0.93	0.93	0.93	306	346.29
40	0.93	0.93	0.93	0.93	0.93	268	336.57
45	0.93	0.93	0.93	0.93	0.93	238	344.46
50	0.93	0.93	0.93	0.93	0.93	218	331.58
55	0.93	0.93	0.93	0.93	0.93	198	331.02
60	0.93	0.93	0.93	0.93	0.93	179	332.94
65	0.93	0.93	0.93	0.93	0.93	166	334.45
70	0.93	0.93	0.93	0.93	0.93	151	329.84
75	0.93	0.93	0.93	0.93	0.93	140	346.09
80	0.93	0.93	0.93	0.93	0.93	130	352.41
85	0.93	0.93	0.93	0.93	0.93	123	353.18
90	0.93	0.93	0.93	0.93	0.93	117	334.83
95	0.92	0.92	0.93	0.92	0.93	113	336.63
100	0.93	0.93	0.93	0.93	0.93	111	344.15

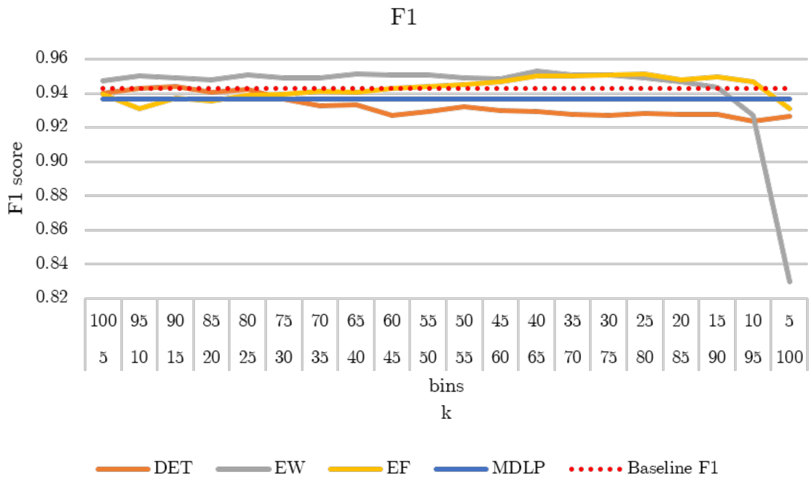


Figure A.32: F1 comparison synthetic 4

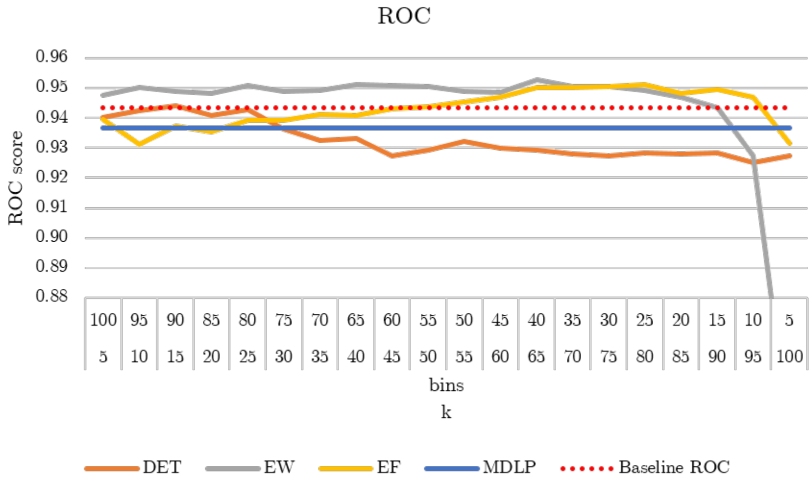


Figure A.33: ROC comparison synthetic 4

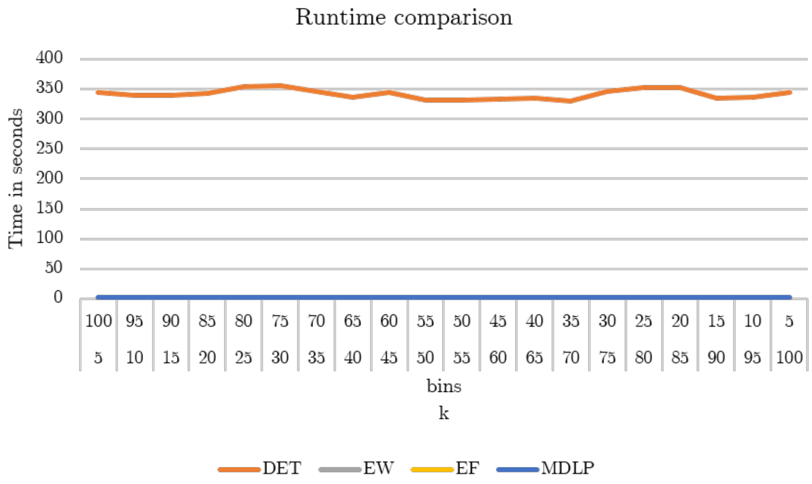


Figure A.34: Runtime comparison synthetic 4

A.12 Synthetic 5

Table A.12: Classification performance synthetic 5

k	**Accuracy**	**F1**	**Recall**	**Precision**	**AUC**	**Bins**	**Time (s)**
5	0.98	0.98	0.98	0.98	0.98	123	5.98
10	0.99	0.99	0.99	0.99	0.99	59	3.22
15	0.95	0.95	0.95	0.95	0.95	40	2.39
20	0.95	0.95	0.95	0.96	0.95	30	1.94
25	0.98	0.98	0.98	0.98	0.98	25	1.78
30	0.98	0.98	0.98	0.98	0.98	20	1.59
35	0.91	0.91	0.91	0.91	0.91	16	1.32
40	0.94	0.94	0.94	0.94	0.94	15	1.28
45	0.92	0.92	0.92	0.92	0.92	14	1.17
50	0.98	0.98	0.98	0.98	0.98	12	1.15
55	0.99	0.99	0.99	0.99	0.99	12	1.07
60	**0.99**	0.99	0.99	0.99	0.99	10	0.99
65	**0.99**	0.99	0.99	0.99	0.99	10	0.85
70	**0.99**	0.99	0.99	0.99	0.99	9	0.83
75	**0.99**	0.99	0.99	0.99	0.99	8	0.80
80	**0.99**	0.99	0.99	0.99	0.99	8	0.79
85	**0.99**	0.99	0.99	0.99	0.99	8	0.77
90	0.86	0.86	0.87	0.88	0.87	7	0.74
95	0.94	0.94	0.93	0.94	0.93	6	0.67
100	0.95	0.95	0.95	0.96	0.95	6	0.65

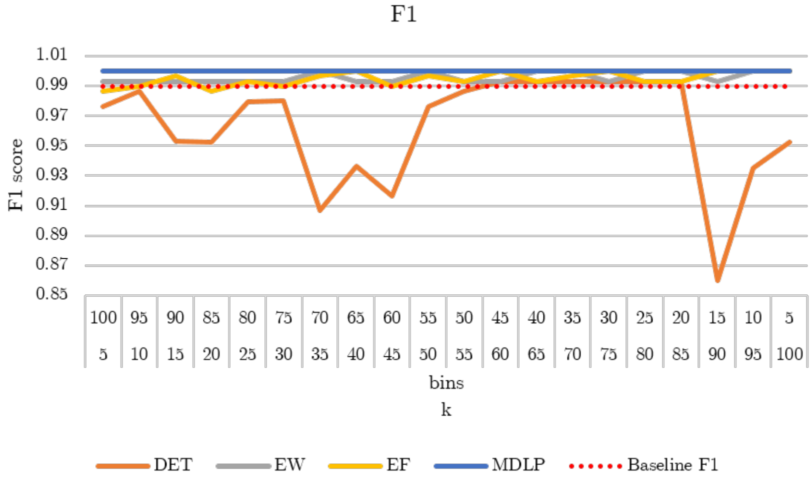


Figure A.35: F1 comparison synthetic 5

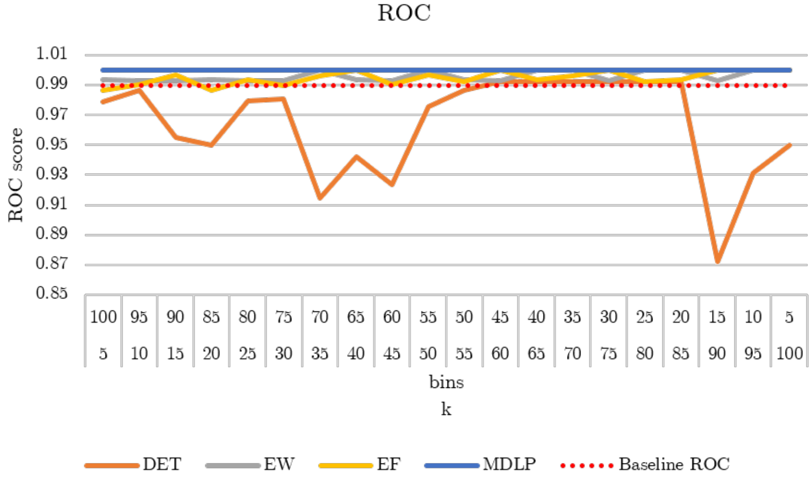


Figure A.36: ROC comparison synthetic 5

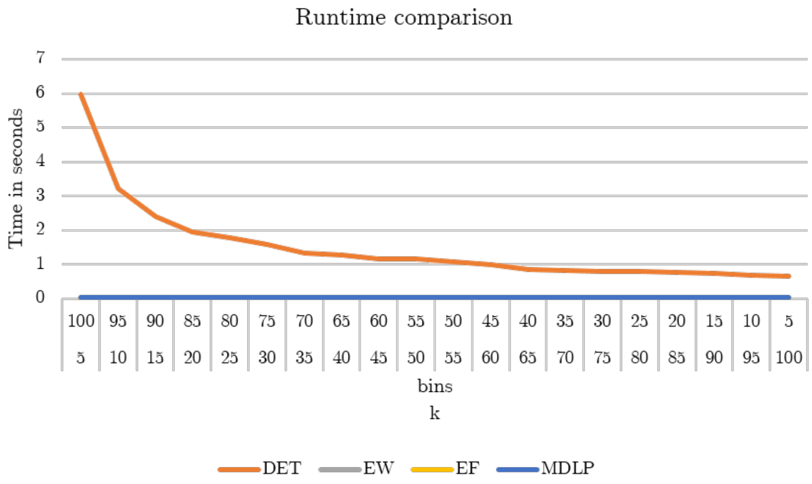


Figure A.37: Runtime comparison synthetic 5

A.13 Synthetic 6

Table A.13: Classification performance synthetic 6

k	**Accuracy**	**F1**	**Recall**	**Precision**	**AUC**	**Bins**	**Time (s)**
5	**0.93**	0.93	0.93	0.93	0.93	118	2.82
10	0.93	0.93	0.93	0.92	0.93	56	2.05
15	0.90	0.90	0.91	0.90	0.91	37	1.62
20	0.89	0.89	0.90	0.90	0.90	28	1.28
25	0.88	0.88	0.89	0.89	0.89	23	1.23
30	0.79	0.79	0.80	0.80	0.80	20	1.13
35	0.82	0.82	0.84	0.85	0.84	17	1.08
40	0.78	0.78	0.81	0.83	0.81	15	0.94
45	0.77	0.77	0.79	0.80	0.79	13	0.93
50	0.87	0.87	0.87	0.87	0.87	11	0.69
55	0.89	0.89	0.90	0.90	0.90	10	0.55
60	0.89	0.89	0.90	0.90	0.90	10	0.56
65	0.87	0.87	0.87	0.87	0.87	9	0.54
70	0.84	0.83	0.83	0.85	0.83	8	0.53
75	0.84	0.83	0.83	0.86	0.83	7	0.51
80	0.73	0.68	0.69	0.84	0.69	5	0.49
85	0.73	0.68	0.69	0.84	0.69	5	0.51
90	0.73	0.68	0.69	0.84	0.69	5	0.51
95	0.73	0.68	0.69	0.84	0.69	5	0.52
100	0.73	0.68	0.69	0.84	0.69	5	0.48

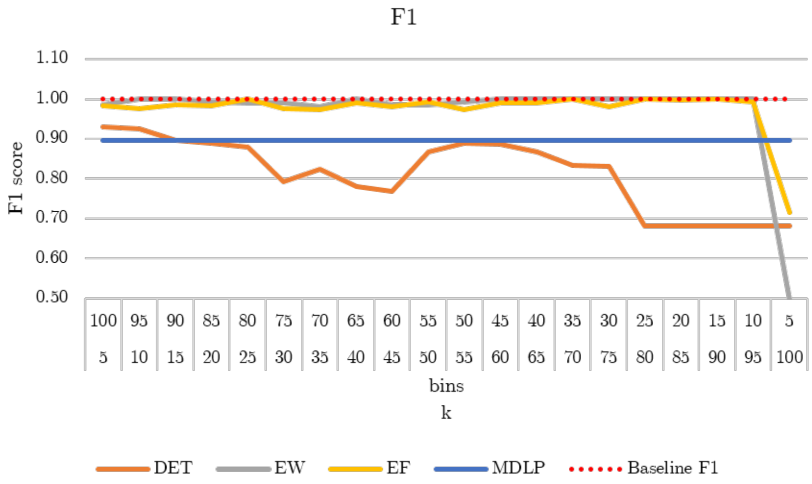


Figure A.38: F1 comparison synthetic 6

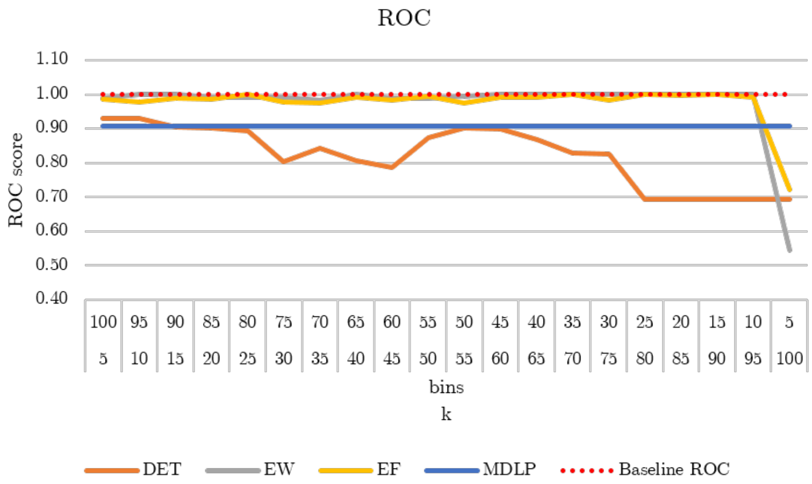


Figure A.39: ROC comparison synthetic 6

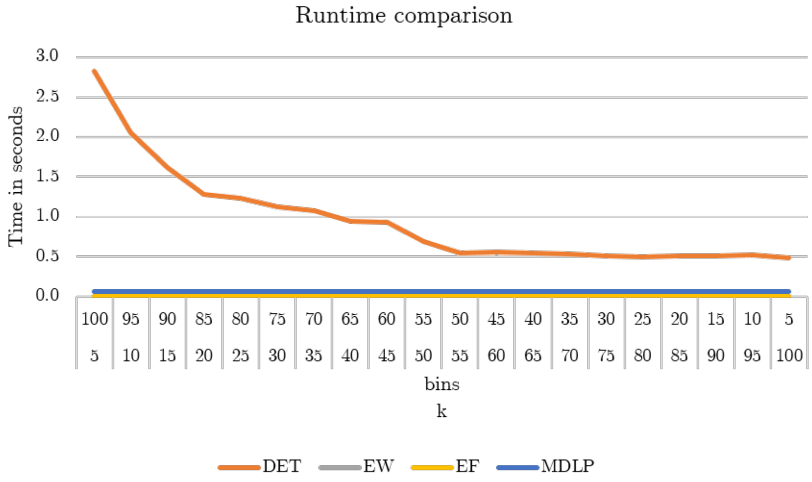


Figure A.40: Runtime comparison synthetic 6

A.14 Synthetic 7

Table A.14: Classification performance synthetic 7

k	**Accuracy**	**F1**	**Recall**	**Precision**	**AUC**	**Bins**	**Time (s)**
5	**1**	1	1	1	1	119	1.00
10	**1**	1	1	1	1	57	0.88
15	**1**	1	1	1	1	37	0.83
20	**1**	1	1	1	1	26	0.84
25	**1**	1	1	1	1	20	0.78
30	**1**	1	1	1	1	19	0.76
35	**1**	1	1	1	1	16	0.72
40	**1**	1	1	1	1	15	0.71
45	**1**	1	1	1	1	13	0.66
50	**1**	1	1	1	1	11	0.62
55	**1**	1	1	1	1	11	0.61
60	**1**	1	1	1	1	10	0.59
65	**1**	1	1	1	1	10	0.59
70	**1**	1	1	1	1	9	0.57
75	**1**	1	1	1	1	8	0.54
80	**1**	1	1	1	1	8	0.54
85	**1**	1	1	1	1	8	0.53
90	**1**	1	1	1	1	6	0.48
95	**1**	1	1	1	1	6	0.49
100	**1**	1	1	1	1	6	0.48

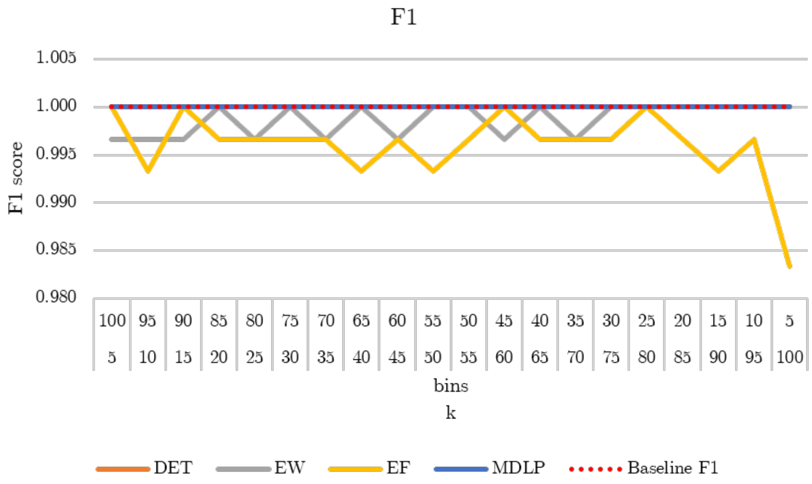


Figure A.41: F1 comparison synthetic 7

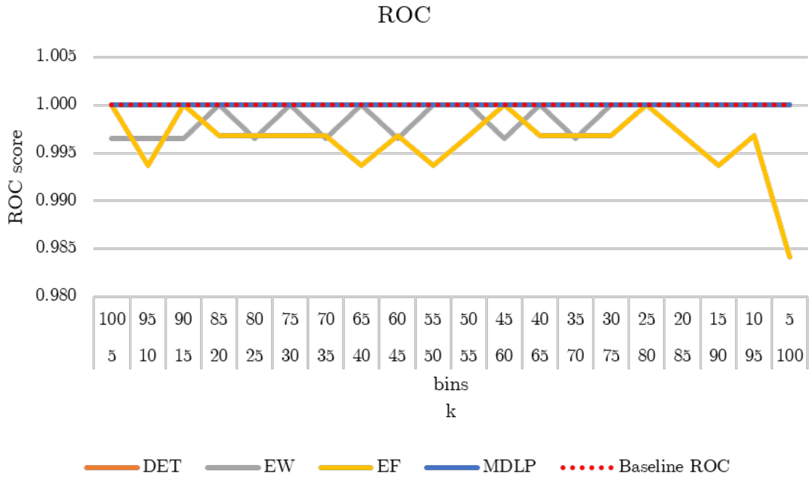


Figure A.42: ROC comparison synthetic 7

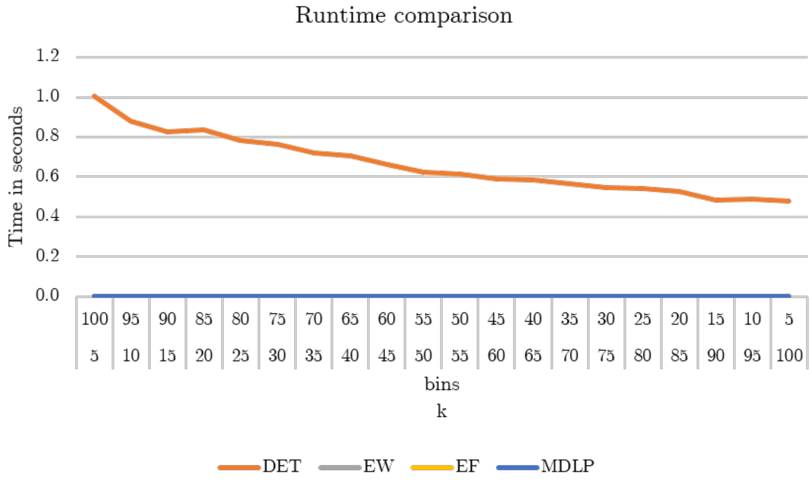


Figure A.43: Runtime comparison synthetic 7

A.15 Synthetic 8

Table A.15: Classification performance synthetic 8

k	**Accuracy**	**F1**	**Recall**	**Precision**	**Bins**	**Time (s)**
5	0.88	0.87	0.88	0.87	113	0.79
10	0.86	0.86	0.86	0.86	57	0.74
15	0.86	0.86	0.86	0.85	39	0.69
20	0.88	0.87	0.88	0.87	26	0.68
25	**0.89**	0.88	0.89	0.88	22	0.65
30	**0.89**	0.88	0.89	0.88	18	0.62
35	**0.89**	0.88	0.89	0.88	16	0.61
40	**0.89**	0.88	0.89	0.88	13	0.62
45	**0.89**	0.88	0.89	0.88	12	0.60
50	0.87	0.86	0.87	0.87	11	0.57
55	0.87	0.86	0.87	0.87	11	0.59
60	0.87	0.86	0.87	0.87	8	0.57
65	0.87	0.87	0.87	0.87	7	0.57
70	0.87	0.87	0.87	0.88	7	0.56
75	0.87	0.87	0.87	0.88	7	0.55
80	0.88	0.87	0.88	0.88	6	0.54
85	0.88	0.87	0.88	0.88	6	0.53
90	**0.89**	0.88	0.89	0.88	6	0.53
95	**0.89**	0.88	0.89	0.88	6	0.53
100	**0.89**	0.88	0.89	0.88	6	0.53
105	0.88	0.88	0.88	0.88	6	0.55
110	**0.89**	0.88	0.89	0.88	6	0.55
115	0.78	0.71	0.78	0.66	5	0.51
120	0.61	0.49	0.62	0.42	4	0.49
125	0.61	0.49	0.62	0.42	4	0.49
130	0.61	0.49	0.62	0.42	4	0.50
135	0.61	0.49	0.62	0.42	4	0.50
140	0.56	0.45	0.57	0.38	4	0.49
145	0.56	0.45	0.57	0.38	4	0.49
150	0.55	0.44	0.56	0.37	4	0.52

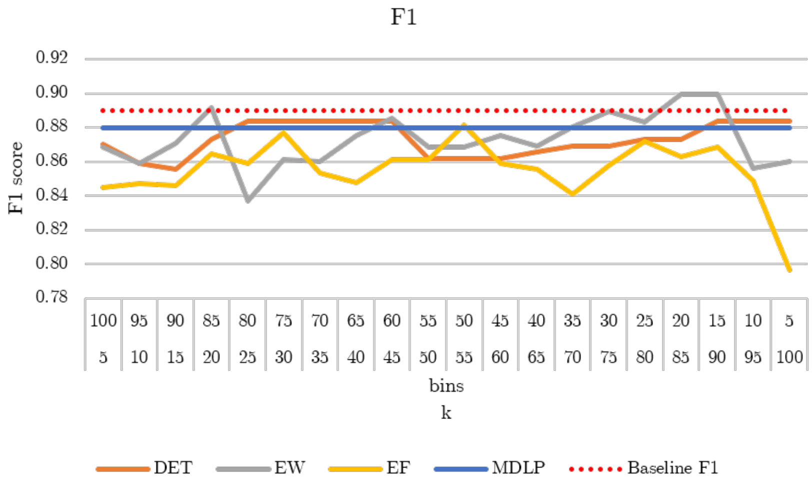


Figure A.44: F1 comparison synthetic 8

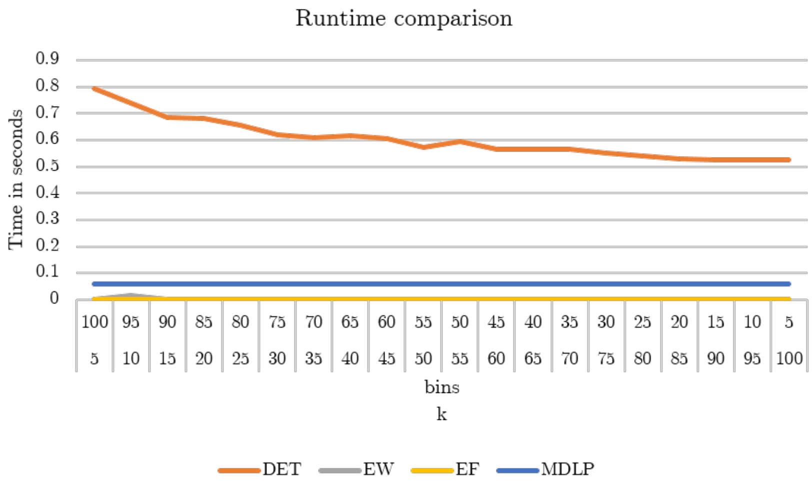


Figure A.45: Runtime comparison synthetic 8