UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science master thesis

Entity Matching of Unlabelled Dataset

First examiner:
Ramon Rico Cuevas

Candidate:
Filips Veips

Second examiner:
Dr. Ioana R. Karnstedt-Hulpus

In cooperation with:
Unknown Group: Akela

August 4, 2024

Abstract

Entity matching is an essential field of study in terms of working with data. Effective coinciding of the entities with each other can significantly increase the effective output out of the data. The main problems with entity matching comes from 2 sources: the flaws of the data and the the matching effectiveness. This research is dedicated to proceeding through these problems in order to state an effective entity resolution algorithm capable of dealing with real-world data. We have constructed four different entity matching models: probabilistic model SPLINK, machine learning models logistic regression, support vector machines and BERT-based transformer. All the models were applied to the same data which was preprocessed accordingly. SPLINK model showed the best result and can be used in similar tasks in the future. However, it is worth mentioning the performance of other models is also quite optimistic and their usage can be viable.

Contents

1. Introduction

1.1 Motivation & Context

Nowadays, multiple players in the IT industry face the huge increase in the the number and in the sheer size of various databases. Corporations as well as individuals collect, evaluate and analyse data on a massive scale. Today there are several solutions towards data management. For example, there are relational database management systems such as MS SQL, data warehousing solutions as Google BigQuery or data lakes as Amazon S3. However, with the increasing amount of data, there is a rising attention towards the data integration [1].

The main question of the process of data integration is how to unite multiple sources for more comprehensive data analysis. One of the key factors influencing the effectiveness of the process of data integration is the presence of the same entities across the multiple data sources. Consequently, issue of linking the records from the different datasets related to the same object becomes evident. This issue is a primary concern of the process called entity matching [2].

The process of entity matching is the process of locating, linking, and combining records that relate to the same entity across many datasets. Entity matching is also known in various sources as record matching, entity resolution, record linkage, or deduplication.

The impact of entity matching is crucial to data analysis development as this technique can be used across various domains. The key role of the entity resolution lies in the spheres where accurate data processing is essential such as education, healthcare and financial institutions. In such cases record linkage is required to clearly define the object of education, medication or financial applications respectively to make the treatment or actions towards the object targeted and efficient by combining information from different

attribute alignment → blocking → entity resolution → canonicalization

Figure 1.1: Entity resolution pipeline

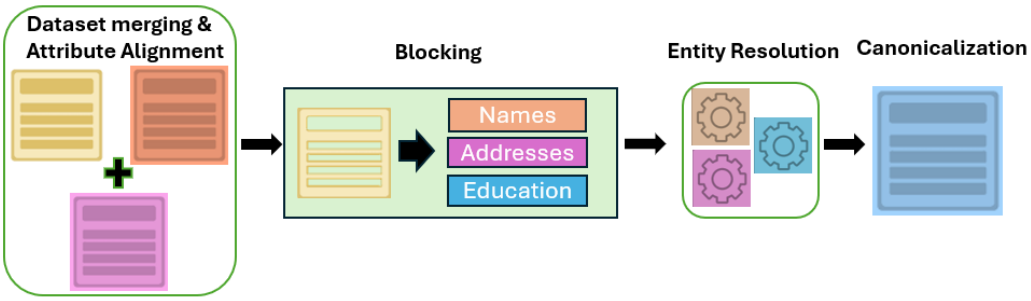

Figure 1.2: The process of Entity Resolution

sources.

The process of entity resolution usually consists of several steps as it can be seen in the figure 1.1 [3]. This figure shows a typical entity resolution solution in the form of the pipeline. The first stage, tribute alignment, refers to identifying common features among the several datasets. In the second step, called blocking, entities which have similar features are combined to the blocks. In such blocks, records are iteratively compared with each other, the records that are not in the same block are considered to be non-matches. If researched structures have one or more traits, they are assigned either to more than one blocks or to blocks within blocks depending on method. In either case, if the method is specified accordingly, there is no data loss. In the entity resolution stage, related objects i.e. the objects having exactly the same features are identified. This process is conducted within the blocks constructed in the previous step. In the fourth stage, additional actions towards the identified entities are applied. The process known as canonicalization refers to merging records, as such the resulting dataset lacks duplicates. Therefore it can be also referred as deduplication.

The process of the pipeline is extensively illustrated in the figure 1.2. Multiple datasets are united, so the common features of raw dataset entities

can be found. Then, in the blocking stage, entities are split into blocks; entities having the same names into one block, having the same addresses into the other. In the step of entity resolution, the objects are processed within the blocks. In the last step, identified matches are returned in the form of pairs or groups of same structures or as deduplicated dataset depending on the purpose of the entity resolution.

Though the pipeline itself is a commonly appreciated solution [2] [4] for the entity resolution, various problems may occur in each step of the pipeline. The most common problems in the entity matching are: [5]

Poor data quality: Data from the real world is rarely perfectly unspoiled, well-organized, and consistent. Raw data which can be both manually and automatically entered may be incorrect, have different spellings, do not follow the schema or simply be absent. Moreover, we cannot always depend on various data sources to adhere to the same format, grammatical norms or schema.
Too large or too small number of matches: If the number of entities is too large, this leads towards $O(N^2)$ matches. In such a case, it is virtually impossible to explicitly compare every pair that could possibly exist. On the other side, if the number of matches is too small, this is the result of the significant imbalance in the data. This can lead to challenges in positive example identification in the abundance of negative instances if a learning-based strategy is employed.
Dependency on human intervention: The possibility of the first two issues compared with often high number of instances leading to necessity of human correction. Different methods of data storage, various features and their impact are making it either unrealistic or impossible to perform entity matching as automated, human-free process. The lack of standardisation is both the result and the cause of such situation.

Though the problems themselves may seem unsolvable, there are mul-

tiple approaches aimed towards the minimization of their negative effect on the record matching[6] and further advancements in the field of entity matching [7].

1.2 Related Work

Entity resolution is constantly growing and evolving, though there are already significant implications by many researchers to the topic. There are several variation of entity matching: deterministic, probabilistic, machine learning and hybrid algorithms.

Deterministic algorithms [8] match records according to precise attribute values and predetermined rules. Although these techniques are simple, they may not be able to handle variations and data entry problems. The usual examples of such techniques are key-based blocking, phonetic encoding (Soundex) or exact matching.

Probabilistic algorithms [9] determine the probability that two records are identical using a combination of attribute comparisons and statistical models. These techniques use comparison functions such as Jaro-Winkler or Levenshtein distances to assess the similarity between entities within the same block, assign attributes weights and determine a composite score for every pair of records. Based on a threshold score, the records are then categorized as matches, non-matches, or possible matches.

Machine learning algorithms make use of both supervised and unsupervised learning approaches in order to increase the precision of entity resolution [5] [10]. However such techniques require labeled training data in order to identify patterns and apply them to the data. The possible examples are neural networks, decision trees, random forests and even transformers.

Hybrid algorithms combine aspects of deterministic, probabilistic, and machine learning techniques. Their essential sense of combining the advantages of the other approaches while omitting their shortcomings. In order to limit the number of candidate pairs, the blocking can be used. For the final resolution probabilistic matching or machine learning models may be applicable. Complicated combination of these methods can be viewed as

brand new models capable of producing cutting-edge results [11].

In terms of dealing with the limitations of the entity resolution, it is worth saying that real world datasets are often plundered with the data which can be considered to be of insufficient quality. This issue is constantly in the vision of the scientific community and is the point of applications of the dedicated advanced methods [12]. Ranbaduge et. al. propose a novel technique called signatures that can accurately link records even when candidate values contain errors or variations, or are missing. Signature is the probability of the compared entity traits, for example names, to be the same feature. Signatures are calculated using mutual feature comparison and if the probability of traits to be the same feature is above the certain specified threshold, such feature can be used in the process of blocking.

As the process of record linkage is constantly evolving, it is of particular interest in using the cutting-edge techniques.The family of the probabilistic algorithms is represented in our research by the SPLINK algorithm. SPLINK is the latest realization of the fastLink algorithm [9]. The idea for the algorithm itself was firstly mathematically introduced by I.P. Fellegi and A.B. Sunter in 1969 [13]. However, fastLink is based on the improved algorithm variations [14] [15]. In latest studies [16] SPLINK showed quite promising results towards the entity resolution. Implementation of the Fellegi-Sunter algorithm is providing additional opportunities [17] in terms of the record matching as for example additional features in terms of blocking and data standardization were implemented there as well. The particular Fellegi model is discussed in the section 3.1.

On the other hand, the advancements in the machine learning algorithms also require to be paid attention on. A number of recent reports [18] advise in favor of using Machine learning algorithms in order to match records. This includes using the most novel neural networks [5]. Hou et al. [10] propose innovative techniques pushing forward the usage of machine

learning algorithms in entity resolution. Two different types of features are proclaimed: attribute value similarity or the tokens occurring in both records or in one and only one record. The first type of feature measures a pair's value similarity at each record attribute. Such attributes require different similarity metrics such as Jaro-Winkler similarity [19] similarity or Levenshtein distance [20]. The second type checks if the feature occurs in both entities or in one entity. In such a case each feature occurrence can show implications on feature match/non-match.

In the cases of the high presence of the textualised records in the data, it becomes viable to use Large Language Models (LLMs) in order to link records. The usage of such models can provide a better machine learning evaluation of the text data and therefore better linkage accuracy [21]. Though the datasets applicable for LLM-based algorithms such as BERT [22] require labeled data without inconsistencies in features or missing data, their usage provide the most advance design towards text-based machine learning.

1.3 Research Objective

The primary focus of our research is the implementation of the contemporary entity matching algorithms. There are several possible applications towards the dealing with the issues of record linkage. However, it is essential to determine the most suitable algorithms towards the limitations of the real-world data. By comparing several methods, the best one in terms of metrics may be used as the universal algorithm or as the most suitable towards given type of the dataset. The method realizations should follow the entity resolution pipeline 1.1 in order to be comparable with each other and with other possible applications. In order to proceed in this task, at first, it is needed to state several key steps of the research.

1. To construct and apply the entity matching model on the input real-world dataset in order to determine the effectiveness of the model on real world data.

2. To construct and evaluate another model or models and compare them in order to research different approaches towards entity matching.

The process of model construction contains the steps of the attribute alignment,blocking and the entity resolution itself. Moreover, due to the nature of the data models should be capable of possible data difficulties mitigation. Therefore, the models which are compared are the probabilistic record linkage model SPLINK, Machine learning models of logistic regression [23] and Support Vector machines [24] for record matching and BERT-based transformer.

The choice of such models is determined by both their novelty and wide selection. [2] SPLINK is the most contemporary iteration of probabilistic record linkage algorithms. Both the logistic regression and support vector machine classifier are stable algorithms which can be used to the entity matching task [25]. While the usage of LLMs in a record linkage is also a contemporary idea. All the models represent different families of the methods, so the selection is wide enough to compare the families as well. The goal of the comparison is to determine the possible direction of the record linkage algorithmic improvements based on the real-world data. Using the resulting paradigm may provide a useful entity resolution solution for many data examples.

2. Data

2.1 Data Description

The data on which manipulations are conducted upon has been provided by Akela of the Unknown Company. This enterprise is focused on providing B2B and 2BC solutions aimed at increased efficiency of the data management and data integration. The research in cooperation with them is also focused on these topics which can influence positively on the company's profile.
The data provided by the Akela is in the form of the single dataset. This dataset is a combination of the 3 raw data sources. The raw data sources has different features and the merging process has been conducted as inner join. However, the presence of the common traits doesn't essentially lead to the presence of the data inside these features leading to multiple missing values.
The dataset itself contains about 1 million rows and 16 columns. It can be described as a compilation of entities from various sources united by the common features which can be found in the columns. Each row is a different entity (the business company) in the dataset. Based on the research object mentioned above the idea is to find row which refer to the same entity. In order for better optimization of the research a dataset of 100,000 rows is used.
The columns of which the dataset consists off are:

1. **id**. The id refers to unique identificator of each entity. Ids are unique for all the rows(entities) and have no duplicates. Ids are presented in the form of strings consisting of both letters and numbers.
2. **name**. The name refers to the name of the company in the source dataset, and is a primary mnemonic feature for manual validation and

result output. The main issue with the names is the fact that different sources may have different names for one entity while in the same time there can be occurrences of different companies which have similar or close phonetically names.

3. **legal_name**. The legal name refers to official legal name of the company based on governmental sites and systems. Presence of the legal name definitely increases chances for successful data match as legal names are unique, however, there are rising concerns of the data operation problems due to data missingness caused by inability to import the unique legal name of the company. This may occur due to various reasons such as access limitations or low development of government data.

4. **alias[1-3]**. The columns alias1, alias2 and alias3 refer to unique codes representing company name reduction or asset market ticket in the data. Such unique codes may be helpful in determining duplicates, as the same companies are be referred to the same aliases across the data. The number in the end refer to the source dataset from which the information has been exported, in our case we can speak of 3 data sources combined in our dataset. However, such information may be limited and it may result in errors in the process of the data analysis.

5. **phone**. The phone of the company. It is presented as string.

6. **crunchbase_url**. The URL of the Crunch base - the major source of the information of various businesses. Crunch base URL are unique fro all the entities there, therefore the presence of such URL increases the efficiency of data matching.

7. **homepage_url**. The URL of the companies site in the World Wide Web. The URLs in WWW follow HTTP protocol which ensures the uniqueness of each Internet domain. Unique URLs can be used for data matching if duplicates.

8. **linkedin_url**. In the LinkedIn as well as in the other social networks the architecture of the networks requires users to have unique iden-

tification numbers. Therefore, entities which have the same id in the social network may be the same value. In many cases such id can be viewed in the unique social network URL-link to the company's page in the network. Therefore, here and below the social network URLs can be used as flags for match.

9. **twitter_url**. The URL for the company's page at Twitter/X.
10. **facebook_url**. The URL for the company's page at Facebook.
11. **instagram_handle**. The URL for the company's page at Instagram.
12. **completeness_score**. Such score shows how much features for the entity are presented in the data and if there is any missing values for the entity in the dataset.
13. **updated_at**. Such feature refer to time when the data was updated the last time. Usually this time remains unchanged from the time when the source dataset has been exported into the working dataset. While such feature can theoretically be used in order to determine duplicates there are also possibilities of misspecification regarding this feature. There can be matches of the data entities which are uncorrelated but have been inputted into the dataset in the same time.
14. **SOURCE**. This feature which is represented by string "source_N" where N is the number of the source dataset from which the information has been exported into the united dataset. In our case there are 3 source datasets.

As it can be understood from above the data contains multiple features which can be used in the process of entity resolution. However, the data exploration is incomplete without the missing value exploration. It can be seen in the figure 2.1

As it can be observed, there are a lot of missing values in the data. Some features are almost entirely consist of missing values. Therefore, we can state that the data should be prepared accordingly.

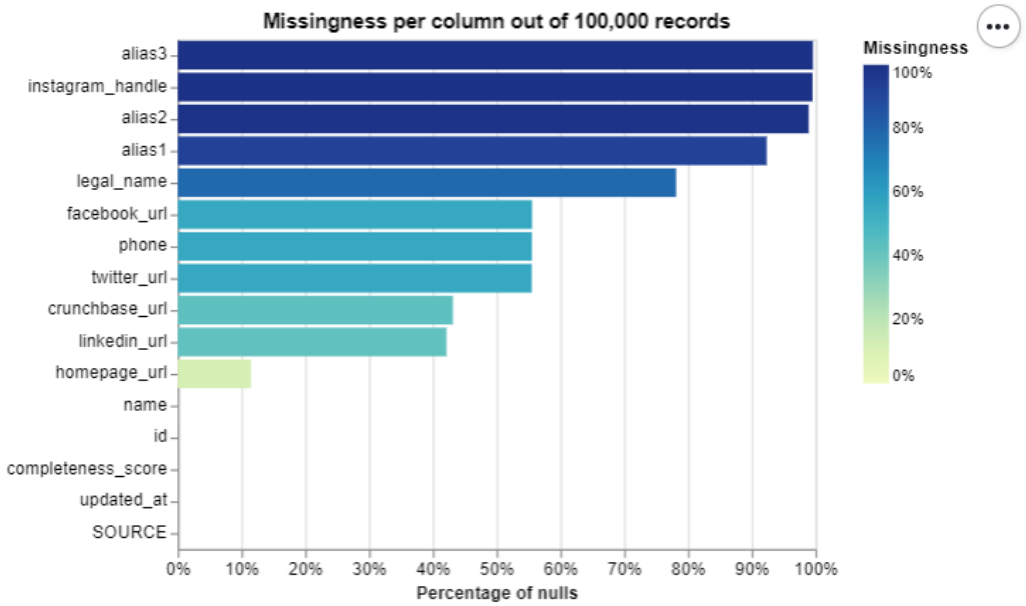

Figure 2.1: Missingness Chart for a sample of 100,000 records

2.2 Data Preparation

2.2.1 Data Cleaning

In order to make dataset suitable for the task it should be structured and prepared. Cleaning is the first and crucial step in the process of data preparation. This method aims at detecting and correcting (or removing) inaccurate, incomplete, or irrelevant records from a dataset. The cleaned dataset is required to be accurate, consistent, complete and efficient in order for algorithms to work correctly.
It is conducted in several steps:

1. Removed columns with excessive number of missing values. In our case such columns are "alias[1-3]".
2. Filled the remaining missing values with empty strings or zeros.
3. Standardized data. All the strings have been lower-cased.
4. Deduplication has been performed. Based on the ground truth dataset discussed in the section 2.2.2 and provided by Akela, duplicates within each source datasets have been removed. This process has took a small fraction of data and has not violated the objective as the task is to find

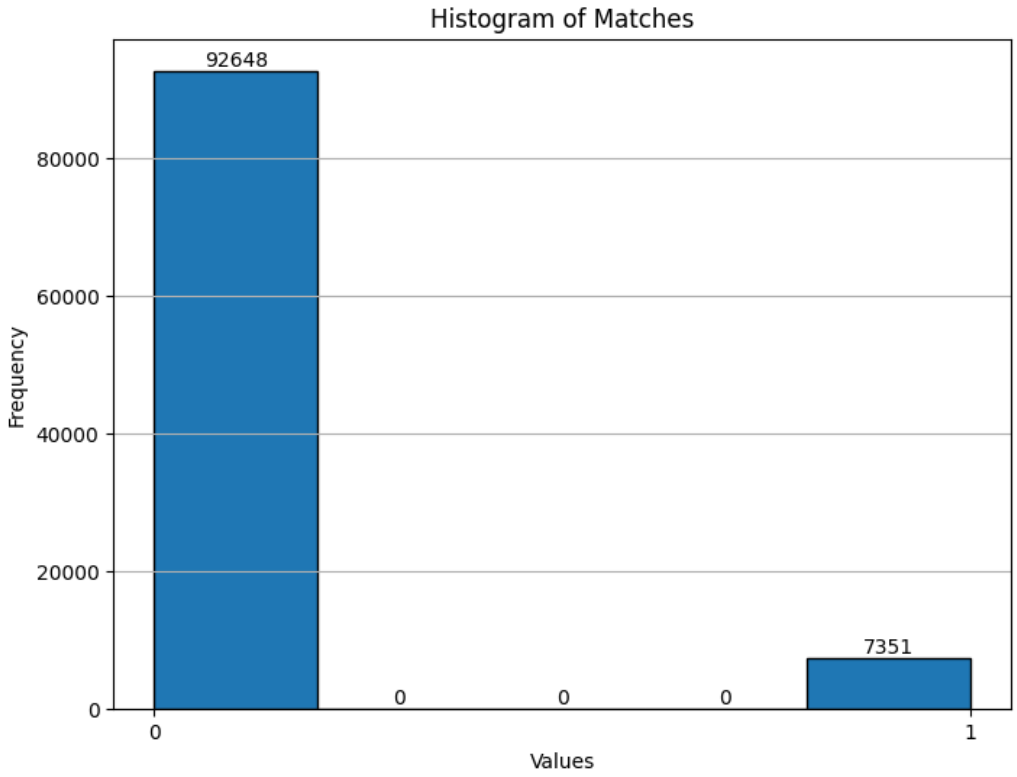

Figure 2.2: Distribution of matches/non-matches in dataset

duplicates from different source datasets.

The data proceeded the process of cleaning can be considered suitable for the data analysis. However, the later used machine learning approaches such as Logistic regression classifier, Support Vector Machines classifier and BERT-based transformer require the ground truth labeled data for their correct application. These concerns are addressed in section 2.2.2.

2.2.2 Data Labeling

Labeling is a method which key part is assigning a label or score to pairs of records to indicate the likelihood that they refer to the same entity. This process is essential for the supervised machine learning methods as these algorithms require ground truth matches for successful match pattern detection.

The process of data labeling has been conducted using the ground truth labels provided by Akela. The distribution of the matches and non-matches

can be seen in the figure 2.2. This figure depicts the clear imbalance in the dataset, that can possibly influence predictions. As there is a significant domination of non-matches in the dataset, it is evident that using a fraction of the dataset is justified, so the blocking technique can be applied [26].
The process of blocking should be applied on the dataset, so the entity resolution shall consider only the matching candidates. As being the candidate means only a chance to be a match, possible candidate dataset is representative of the matches and non-matches with possible balance towards the matches. The process of blocking is be discussed explicitly in the section 4.1.2.

2.3 Legal & Ethical remark

Another part worth mentioning is ethical and legal consideration of the data. During the research process conducted along the Akela the non-disclosure agreement(NDA) was signed between the parties. The NDA explicitly prohibits data distribution and publication. However, the analysis performed can be distributed freely as well as the results and the conclusions of the research.
Regarding the origin of the data, the presented entities have been collected with the attention towards ethical and legal question by Akela employees. From the part of the research, it is possible to state the absence of any legal and ethical concerns during the process of data analysis.

3. Methods

3.1 Probabilistic Record Linkage

Probabilistic Record Linkage is a method used to identify, match, and merge records that refer to the same entity across different data sources. This process, successfully deals with various challenges connected to representations of entities in multiple datasets.
In the context of entity resolution, probabilistic record linkage offers a statistical approach to estimate the likelihood that two records refer to the same entity. Unlike deterministic methods that rely on exact matches of predefined attributes, probabilistic record linkage calculates a probability score for each pair of records, indicating the likelihood of a match [8]. This allows for more flexibility and accuracy, particularly in handling incomplete or inconsistent data [16].
The probabilistic record linkage approach we are using in our research is SPLINK realization of the Fellegi-Sunter algorithm.

3.1.1 Fellegi-Sunter model

The Fellegi-Sunter model consists of the following probabilities:

$$P(\text{Records match}) = \lambda \tag{3.1}$$

$$P(\text{Observation} \mid \text{Records match}) = m \tag{3.2}$$

$$P(\text{Observation} \mid \text{Records do not match}) = u \tag{3.3}$$

Where:
λ: probability that any two records match.
m: probability to be linked based upon the researched feature given the

records are a match.
u: probability to be linked based upon the researched feature given the records are **not** a match.
m-probability in this model is a measure of data quality. The higher the chance of the disturbance in the data, the less the m-probability.
u-probability is a measure of coincidence or cardinality. The less cardinality the higher probability to match.
Then the probability to match entities given the feature (observation) is calculated as follows:

$$
\begin{aligned}
P(\mathsf{Match} \mid \mathsf{Observation}) &= \frac{2^{\log_2\left(\frac{\lambda}{1-\lambda}\right)+\log_2\left(\prod_i^{\mathsf{features}} \frac{m_i}{u_i}\right)}}{1+2^{\log_2\left(\frac{\lambda}{1-\lambda}\right)+\log_2\left(\prod_i^{\mathsf{features}} \frac{m_i}{u_i}\right)}} \\
&= \frac{\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\mathsf{features}} \frac{m_i}{u_i}}{1+\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\mathsf{features}} \frac{m_i}{u_i}} \\
&= 1-\left[1+\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\mathsf{features}} \frac{m_i}{u_i}\right]^{-1}
\end{aligned}
\tag{3.4}
$$

The entity resolution model such as SPLINK using Fellegi-Sunter algorithm receive the probabilities to match for each entity based on each feature as a result. Taking into account such probabilities it is possible to find matches.

3.1.2 SPLINK

SPLINK (Scalable Probabilistic linkage) is a sophisticated record linkage model designed to handle large-scale data linkage tasks efficiently. SPLINK is distributed in the form of an open-source software package which is mainly used for probabilistic data linking and entity resolution. It leverages probabilistic techniques to estimate the likelihood that records across different datasets refer to the same entity, thereby enabling accurate and scalable link-

age [9]. The key feature of the SPLINK model is its exceptional performance in scenarios where more conventional deterministic matching techniques are insufficient, as in cases where the input data is inaccurate or missing as in our case.
SPLINK package possesses a big variety of instruments which are useful in probabilistic linkage. For example, data preprocessing techniques such as data cleaning, blocking and pairwise comparison can be done within the package. Weighted Scoring is used in the SPLINK model in order to determine weights of features in candidate pairs. The evaluation in its turn is conducted through the usage of the Expectation-Maximization algorithm. This algorithm calculates and then maximizes log-likelihood function for pairs of candidates to be matches or not matches.

3.2 Machine Learning Algorithms

Machine learning algorithms is one more important family of the entity matching algorithms, which can be used in order to estimate pairs in the given data. Generally, they are divided into two parts: supervised and unsupervised algorithms. Supervised machine learning algorithms require a number of already matched or labeled data in order to estimate patterns which allow the data to refer to the same entity[25].
The machine learning algorithms require cleaned and labeled data. Blocking is also an essential part as it highly increase the efficiency of the model. All the the data preparation algorithms have been conducted using Python Record Linkage toolkit. This is a powerful element capable of the data linkage and of the application of the Machine learning algorithms. The algorithms we used are discussed below. The whole structure of the entity resolution can be seen in the figure 3.1. The machine learning algorithms are applied in the final processes of entity matching and clustering. Their particular structure depending from algorithm to algorithm are be discussed in the according sections 3.2.1, 3.2.2 and 3.2.3.

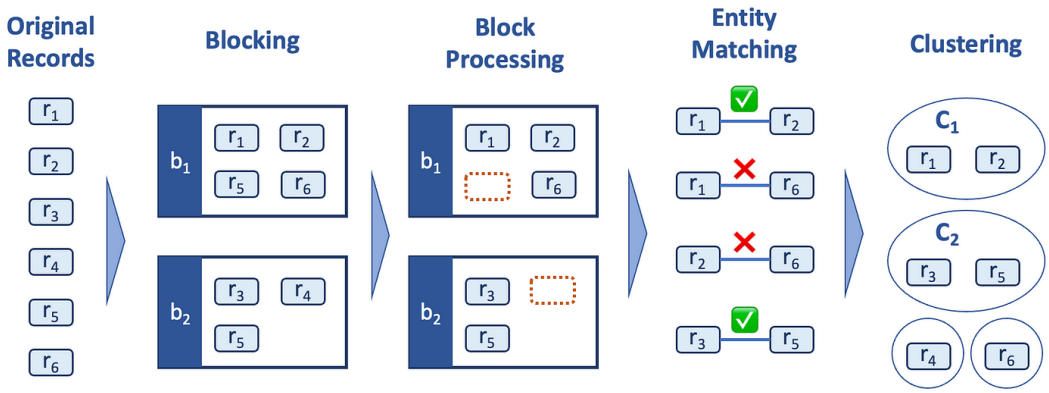

Figure 3.1: Entity resolution using the ML model

3.2.1 Logistic Regression

Logistic regression is a supervised learning algorithm used for binary classification problems. It models the probability that a given input belongs to a particular class. This particular feature can be used in entity matching as logistic regression may be used to predict whether a pair of records is a match or a non-match [23].

Basically, the algorithms uses the features belonging to entities as the patterns for classification. The algorithms requires a labeling dataset with values of 0 and 1, which depict non-match and match respectively. Then the sigmoid function is applied to the linear combination of input features in order to find the probability of a match. Use the binary cross-entropy loss to measure the difference between the predicted probabilities and the actual labels. The binary cross-entropy loss function is used to measure the difference between the predicted probabilities and the actual labels. Its minimization is done via the optimization algorithm, often the gradient descent. The logistic regression is an obvious choice of the machine learning algorithm which is capable of performing entity matching while is not very complex, especially computationally.

3.2.2 Support Vector Machines

Support Vector Machines (SVMs) is a family of the supervised learning algorithms used for classification and regression tasks [24]. They work by finding the hyperplane that best separates the classes in the feature space.

Regarding the process of the entity matching, SVMs can classify pairs of records as matches or non-matches based on similarity features. The feature space is constructed for all the entities taking into account vectorised feature similarity through kernel function application. The support vectors identified after solving the optimization problem on the feature space. The choice of SVM model is derives by its structure. This is the more complex model compared to logistic regression and in the same time facilitates the machine learning algorithm. Therefore, such a model is useful in fulfilling the research objective [3].

3.2.3 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a highly efficient deep learning model based on the transformer architecture [22]. BERT can capture complex patterns from the text, making it highly effective for natural language processing tasks. This ability makes it highly effective for entity matching, where the goal is to determine if different records refer to the same entity. The key advantage of BERT model is its ability to effectively deal with text data, however its performance fades compared to other models if there is plenty of numerical training data [21].

BERT is based on the transformer architecture, which, at first, embeds the data in to encoded vectors with preserved order of the tokens. Feed-forward neural network applies non-linear transformations to the encoded input leading to pattern estimation which helps to determine matches. The input data has to undergo labeling, tokenization and feature extraction in the beginning of the process. This and the complex model algorithm determine the most evident disadvantage of the model: its computational inefficiency. However, it is worth noting that the complex structure of the model provides the reason of testing the model on the data in order to understand the way it works with our dataset.

The whole idea of the BERT and other transformers can be viewed in the figure 3.2. The actions stated above here can be seen step-by-step. Here we can see how 2 entities, 2 rows from the dataset in our case, are loaded into the transformer. The entity features are embed and vectorized and then added

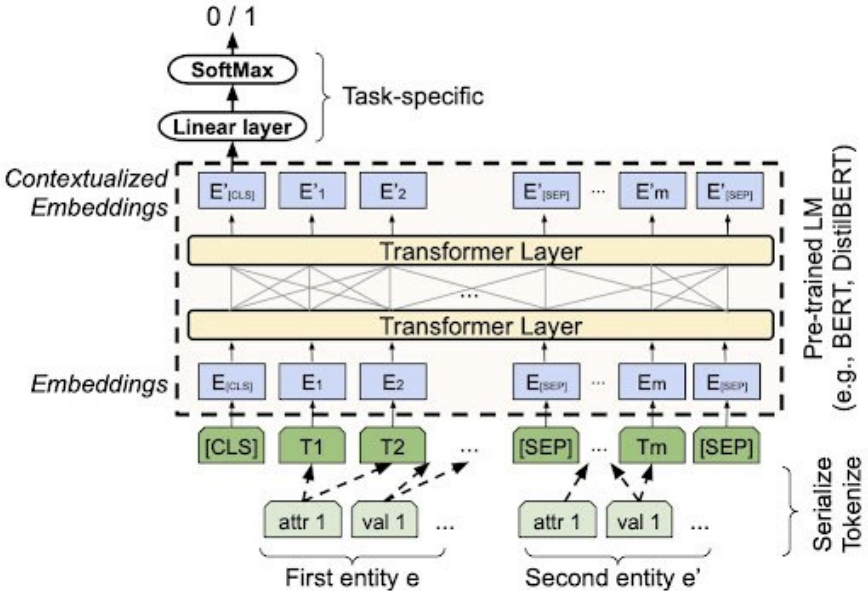

Figure 3.2: Entity matching using transformers

to the transformer layer of the neural network. The output of the network is a layer of contextualized embeddings. Such are further transformed and inputted into the SoftMax function which determines the matches or non-matches. The whole structure is rather complex, though it may give decent results.

4. Results

4.1 Metrics & Blocking

4.1.1 Metrics

Throughout the research, several key evaluation metrics have been used. Their usage was determined by the research objective. To speak precisely, the key factor influencing the effectiveness of the model is the ability to locate and produce the matched entities across the entity pool. Therefore, it is possible to consider the matching as a classification task with matched entities being the correctly classified objects. In such a case, the particular attention should be paid towards the true positive results, while the impact of true negative results should be neglected [27]. As such, it is more important to pay attention towards precision and recall and, subsequently, to calculate F1 score than using accuracy [28].
Precision measures the proportion of the matches that are actually correct. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{4.1}$$

Recall which is also known as Sensitivity or True Positive Rate, measures the proportion of matches that are correctly identified by the model. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{4.2}$$

Where:

True Positives (TP): The correct matches successfully identified by the model. This is the predicted values which are also attributed to ground true matches.

False Positives (FP): The non-matches incorrectly classified as matches. The matches are considered to be false positives, when they are predicted as matches while not being the labeled as true matches.

False Negatives (FN): The matches incorrectly classified as different entities. False negatives are the successfully blocked matches, but that have been predicted as non-matches.

The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both the mentioned evaluative criteria. F1 score is essentially important in the cases of the dataset imbalance, such as in our case. The F1 score is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

The model are evaluated using the stated metrics. The decisive metric in the research is F1 score as a measure combining both precision and recall.

4.1.2 Blocking

Blocking is a method which the main purpose is to reduce the number of comparisons between records while performing record linkage [29]. This effect can be achieved by dividing the data into smaller subsets, or blocks, based on certain criteria instead of comparing every record with all other records. The comparison of each record with all other records results in the complexity of $O(N^2)$. On the contrary, record comparisons while using blocking are performed only within the blocks which significantly reduces computational complexity and time.

The blocking was performed on the following features: **name, legal_name, crunchbase_url, homepage_url, facebook_url, twitter_url, linkedib_url, instagram_handle, phone**. In such a case all the database features defining the comparison to be possible are accounted in the blocking. **crunchbase_-**

url, homepage_url, legal_name are blocked on the exact matching, as such URL features are a representation of unique id and the location of the semantically close instances is harmful to the accuracy of the model. "Legal_name" is also a unique feature when presented, so it is also blocked using the exact match rule.
facebook_url, twitter_url, linkedib_url, instagram_handle, phone are blocked using the Jaro-Winkler similarity [19]. The threshold for successful blocking is 0.9. This is done due to the fact that the entities (businesses) may use semantically close, but not identical names for different social networks. For example, trade company "Trade Company" may use Facebook "...trade_company..." URL while using "...tradecompany1..." URL in Twitter. The threshold is put high in order not to match different businesses named closely to each other. **name** is blocked sing the Jaro-Winkler similarity and Levenshtein distance [20]. The threshold for successful blocking is 0.75. The loose threshold for Jaro-Winkler is needed to allow for more comparison results. The accuracy of the model is preserved by adding additional requirement of Levenshtein distance.
The process of blocking provides us with the list of candidates. The blocking package construct pairs of the entities **within** the blocks. The proportion of correctly labeled matches among the candidate pairs is present in the figure 4.1. As we can see several potential matches are not among the candidate pairs, though the matches are making almost 25% of candidate dataset. Therefore we may consider the proportion of matches/non-matches to be sufficient.

4.2 Entity Resolution

4.2.1 SPLINK

As was stated in the previous sections data was prepared for the model. The data underwent cleaning, blocking, labeling and was handled into the SPLINK model. The model calculates weights and estimates probabilities to match using log-likelihood function[14]. The weights for each compar-

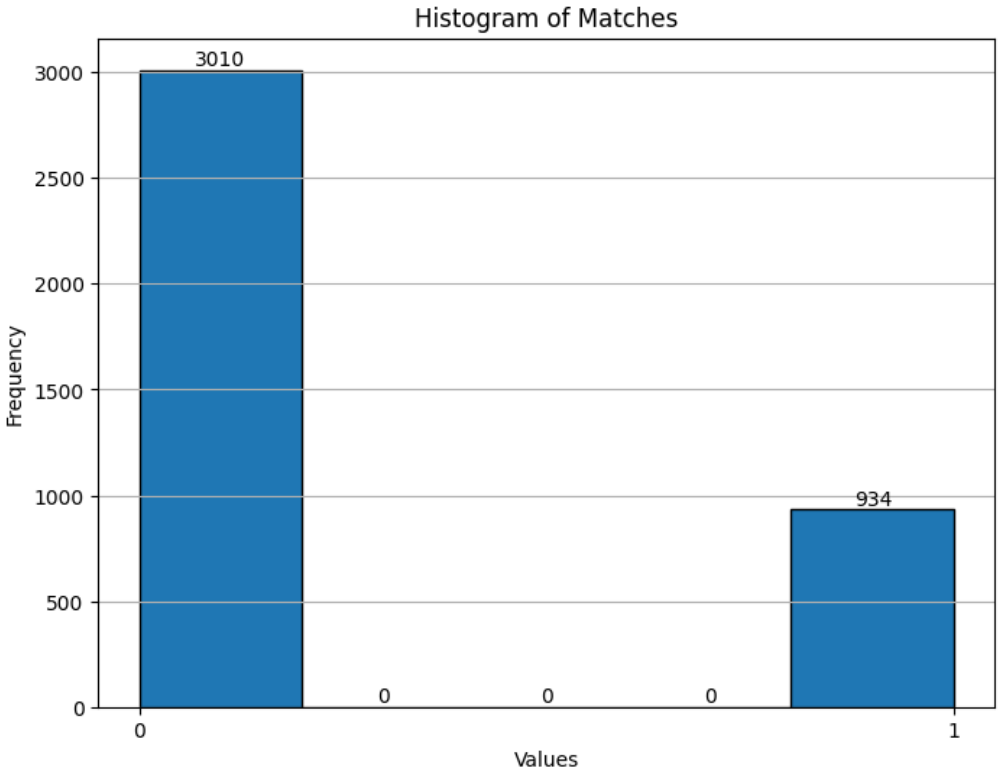

Figure 4.1: Proportion of matches and non-matches after blocking was applied

ison feature can be seen in the figure 4.2. On this figure we can observe the SPLINK model emphasis on the exact match of the URLs, while Jaro-Winkler similarity doesn't influence the probability to be matched significantly. The possible interpretation of this fact, is that the social network URLs tend not to differ across the domains. On the contrary, Jaro-Winkler similarity and Levenshtein distance provided a significant number of possible matches for names and legal names. This stands for the divergence of entity namings across the domains.

The performance of the SPLINK model are observed on the figure 4.3. The AUC area on the graph equals to **0.88**. The results of the metrics for the SPLINK model are presented in the table 4.1. Additional insights on the SPLINK model can be observable in appendix figure A.1.

4.2.2 Machine Learning Algorithms

According to discussed architecture of the Machine learning models, both logistic regression and support vector machine classifiers as well as BERT-

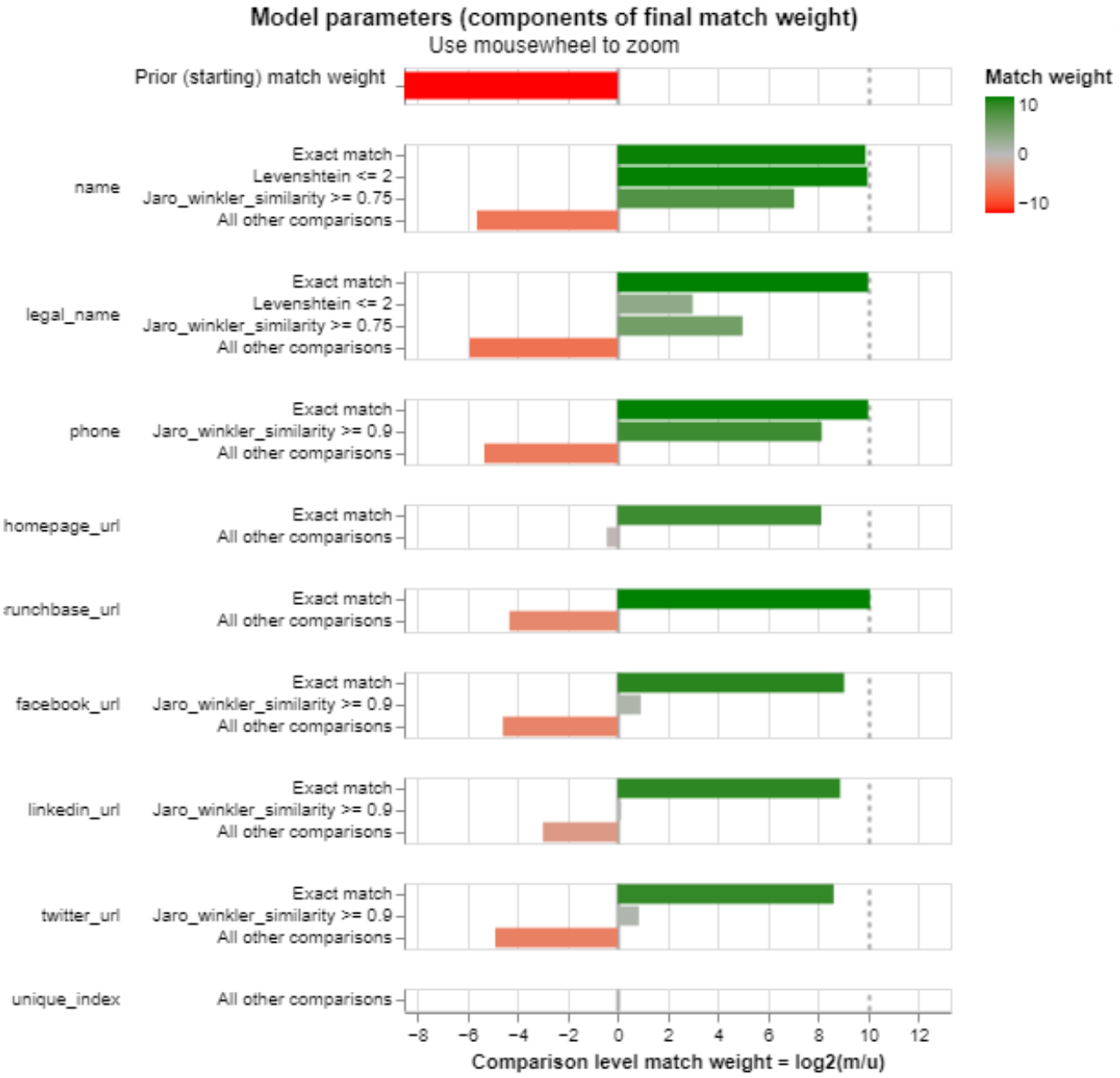

Figure 4.2: SPLINK feature weights

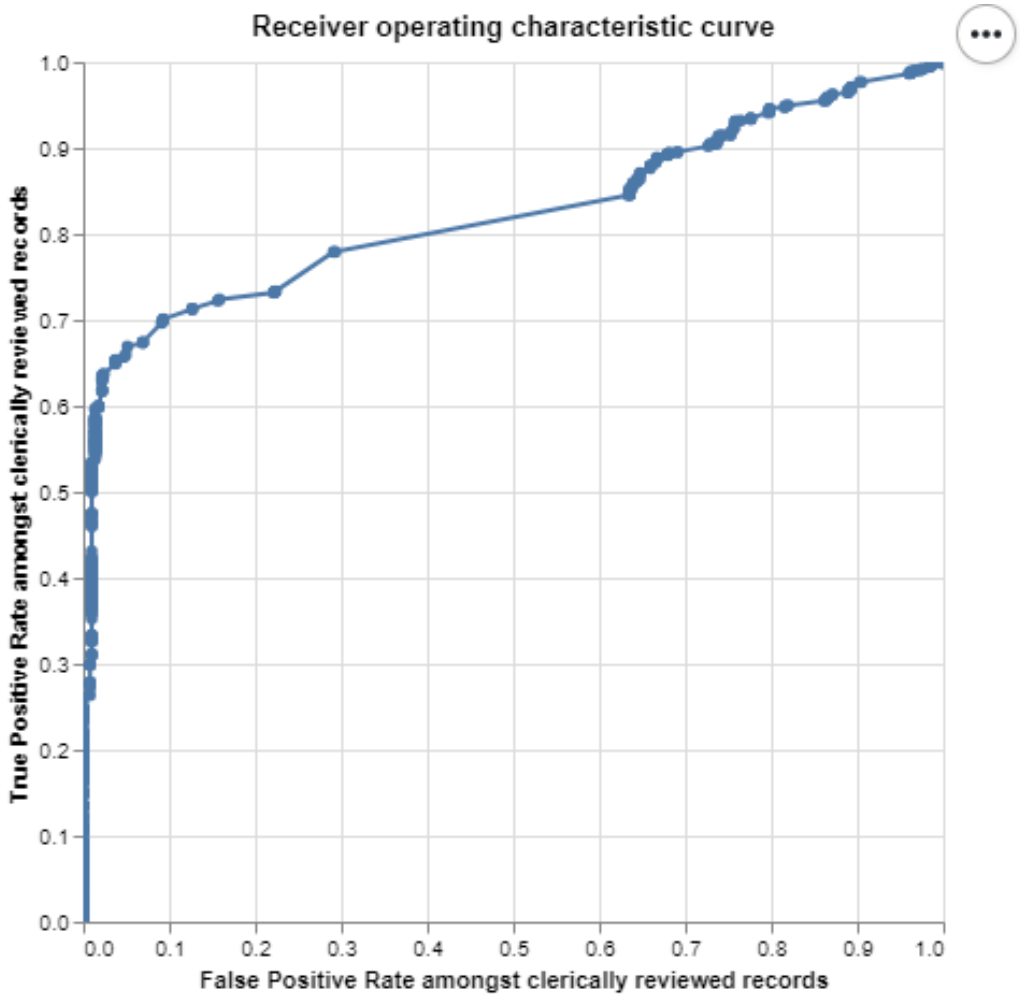

Figure 4.3: SPLINK ROC curve

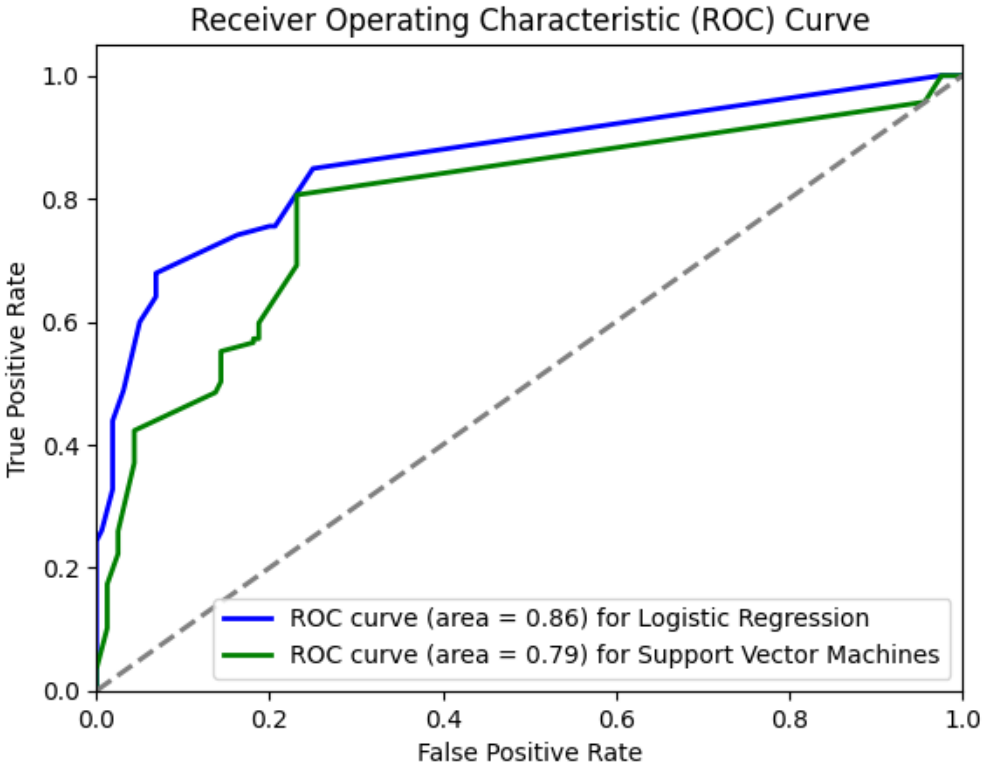

Figure 4.4: Log. Reg. & SVM ROC-AUC

based transformer have been constructed and applied on data.

The figure 4.4 shows the resulting ROC curves for the Logistic regression and support vector machines. The figure 4.5 shows the confusion matrix as a result of the BERT-based transformer.

The results of the models according to evaluation metrics are presented in the table 4.1. Here we can see the comparison of the models using metrics of precision, recall and F1 score.
The highest F1 score as the decisive measurement is shown by the SPLINK probabilistic linkage model. Both SPLINK and BERT models show the dominance of precision over recall. Taking into the account BERT confusion matrix in the figure 4.5, it is possible to state that both models tend to be conservative in predictions. These model overestimate the precision while neglecting many possible matching candidates. However, as the recall Of the SPLINK model is higher leading to higher F1 score, we can derive a more suitability of SPLINK model as its wider assumption of matches helps to link more entities.

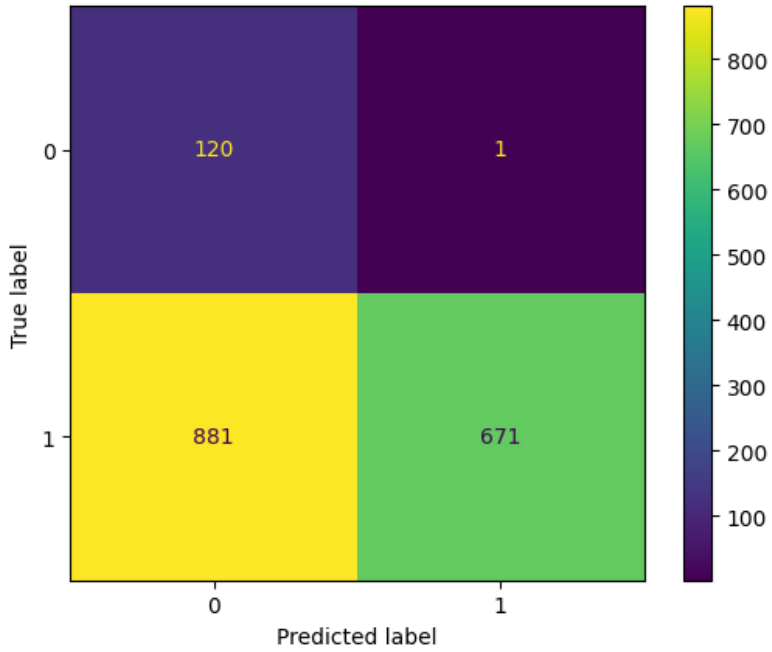

Figure 4.5: Confusion Matrix of BERT algorithm

SVM and logistic regression models show similar and not very high results. This fact may be explained by the difficulties that may occur during the usage of classification algorithms in the process of entity matching. Taking also into account AUC values from the figure 4.4, it is possible to derive several conclusions. Firstly, the higher recall than precision in both cases speaks of wider range of candidates considered to be matching, leading to rise in false positive instances and lower precision. Secondly, AUC is high in both cases, this means the models are capable of successfully identifying the matches in 86 % case for Logistic regression and 79 % cases for SVM. However, their range of identification is too wide, leading to much false positive instances.

In addition it is worth saying that BERT's behaviour was highly computationally expensive which led to some simplifications of the model such as lower amount of epochs. This might lead to less accurate results than theoretically possible.

	Model	Precision	Recall	F1
0	SPLINK	0.997	0.562	0.655
1	Log Reg	0.305	0.542	0.390
2	SVM	0.275	0.390	0.322
3	BERT	0.999	0.432	0.603

Table 4.1: Table with model results and comparison metrics

5. Conclusion & Discussions

To sum up, several scientifically advanced models were implemented. The model constructed show wide range of the methods capable of dealing with entity resolution problems. The real-world data presented several difficulties such as data missingness and overall imbalance. In such a case SPLINK model representing probabilistic record linkage model family showed the best result. Logistic regression classifier, support vector machine classifier and BERT transformer showed decent results, but unmatched to SPLINK model.

While SPLINK showed the best results, BERT also showed substantial result though we can observe difficulties, especially computational, related to its complexity. The logistic regression and support vector machines both showed rather mediocre results, but they also showed a potential for improvement in the future.

All the models may be theoretically improved and show better results by adjusting internal thresholds as well as threshold used for blocking.

A. Appendix

A. Appendix

A.1 Appendix visualization

In this section additional materials are presented in order to increase the thesis clearness.

In the figure A.1 the weights for the entity 0 are presented. Here we may see the example of how the model is working. The model calculates the weights for each feature of each candidate entity influencing probability to be a match or non-match.

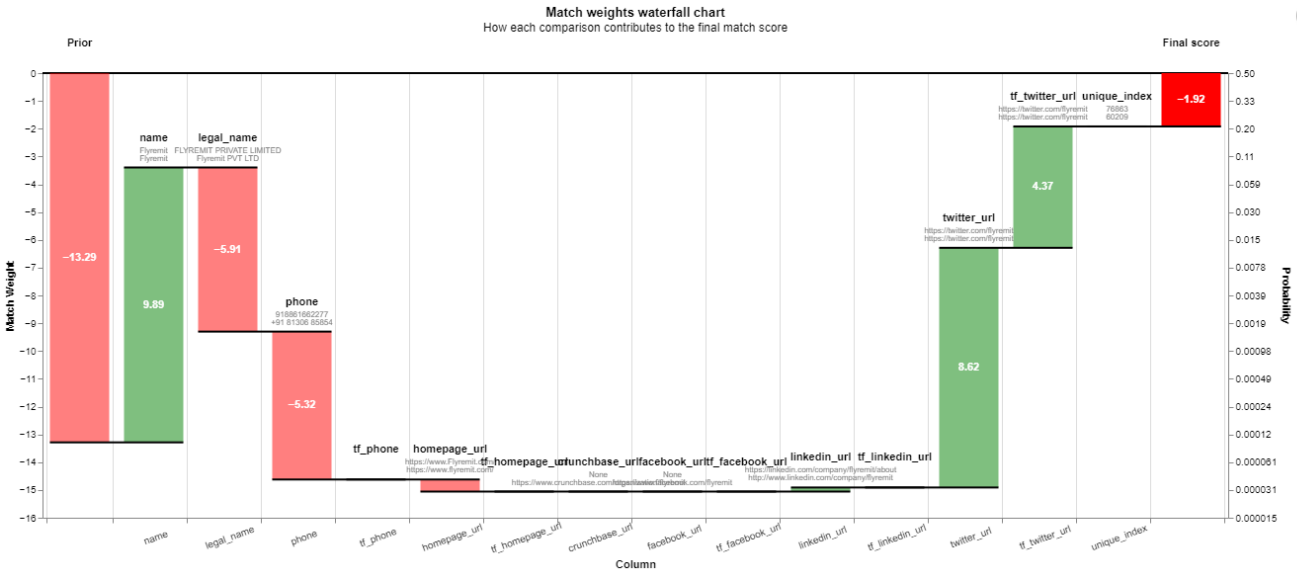

Figure A.1: SPLINK weights for entity 0

Bibliography

[1] M. Stonebraker and I. F. Ilyas, "Data integration: The current status and the way forward," *IEEE Data Eng. Bull.*, vol. 41, no. 2, pp. 3–9, 2018. [Online]. Available: http://sites.computer.org/debull/A18june/p3.pdf.

[2] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis, *End-to-end entity resolution for big data: A survey*, 2020. arXiv: 1905.06397 [cs.DB]. [Online]. Available: https://arxiv.org/abs/1905.06397.

[3] O. Binette and R. C. Steorts, "(almost) all of entity resolution," *Science Advances, 8(12)*, Mar. 2022. DOI: https://doi.org/10.1126/sciadv.abi8021.

[4] L. Barbosa, V. Crescenzi, X. L. Dong, *et al.*, "Big data integration for product specifications.," *IEEE Data Eng. Bull.*, vol. 41, no. 2, pp. 71–81, 2018.

[5] N. Barlaug and J. A. Gulla, "Neural networks for entity matching: A survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 3, pp. 1–37, Apr. 2021, ISSN: 1556-472X. DOI: 10.1145/3442200. [Online]. Available: http://dx.doi.org/10.1145/3442200.

[6] A. Sefid, J. Wu, A. C. Ge, *et al.*, "Cleaning noisy and heterogeneous metadata for record linking across scholarly big datasets," Jun. 2019. DOI: https://doi.org/10.48550/arXiv.1906.08470.

[7] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007. DOI: 10.1109/TKDE.2007.250581.

[8] S. B. Dusetzina, S. Tyree, A.-M. Meyer, A. Meyer, L. Green, and W. R. Carpenter, *Linking data for health services research: A framework and instructional guide [internet]*, 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK253312/.

[9] T. Enamorado, B. Fifield, and K. Imai, "Using a probabilistic model to assist merging of large-scale administrative records," *American Political Science Review*, vol. 113, no. 2, pp. 353–371, 2019. DOI: https://doi.org/10.1017/S0003055418000783.

[10] B. Hou, Q. Chen, Y. Wang, Y. Nafa, and Z. Li, "Gradual machine learning for entity resolution," Jun. 2019. DOI: https://doi.org/10.48550/arXiv.1810.12125.

[11] A. Narayan, I. Chami, L. Orr, S. Arora, and C. Ré, *Can foundation models wrangle your data?* 2022. arXiv: 2205.09911 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2205.09911.

[12] T. Ranbaduge, P. Christen, and R. Schnell, "Large scale record linkage in the presence of missing data," Apr. 2021. DOI: https://doi.org/10.48550/arXiv.2104.09677.

[13] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, pp. 1183–1210, 1969. [Online]. Available: https://api.semanticscholar.org/CorpusID:17349112.

[14] X. L. Huiping Xu and S. Grannis, "A simple two-step procedure using the fellegi–sunter model for frequency-based record linkage," *Journal of Applied Statistics*, vol. 49, no. 11, pp. 2789–2804, 2022, PMID: 35909667. DOI: https://doi.org/10.1080/02664763.2021.1922615.

[15] T. H. Vo, G. Chauvet, A. Happe, E. Oger, S. Paquelet, and V. Garès, "Extending the fellegi-sunter record linkage model for mixed-type data with application to the french national health data system," *Computational Statistics & Data Analysis*, vol. 179, p. 107 656, 2023, ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2022.107656. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947322002365.

[16] R. Linacre, *Deduplicating and linking large datasets using splink*. Nov. 2023. [Online]. Available: https://realworlddatascience.net/case-studies/posts/2023/11/22/splink.html.

[17] R. Linacre, S. Lindsay, T. Manassis, Z. Slade, and T. Hepworth, "Splink: Free software for probabilistic record linkage at scale," *International Journal of Population Data Science, 7(3)*, Aug. 2022. DOI: https://doi.org/10.23889/ijpds.v7i3.1794.

[18] S. Tahamont, Z. Jelveh, M. McNeill, S. Yan, A. Chalfin, and B. Hansen, "No ground truth? no problem: Improving administrative data linking using active learning and a little bit of guile," National Bureau of Economic Research, Working Paper 31100, Apr. 2023. DOI: 10.3386/w31100. [Online]. Available: http://www.nber.org/papers/w31100.

[19] M. A. Jaro, "Probabilistic linkage of large public health data files," *Statistics in Medicine*, vol. 14, no. 5-7, pp. 491–498, 1995. DOI: https://doi.org/10.1002/sim.4780140510. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.4780140510. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780140510.

[20] P. Black, "Dictionary of algorithms and data structures," Jan. 2023. DOI: 10.18434/T4/1422485.

[21] R. Peeters and C. Bizer, "Entity matching using large language models," Jun. 2024. DOI: https://doi.org/10.48550/arXiv.2310.11244.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv:

1810.04805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.04805.

[23] D. Dey, "Entity matching in heterogeneous databases: A logistic regression approach," *Decision Support Systems*, vol. 44, no. 3, pp. 740–747, 2008, ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2007.10.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923607001741.

[24] M. Lee and T. Jo, "Support vector machine for string vectors," in *Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006*, D.-S. Huang, K. Li, and G. W. Irwin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1056–1067, ISBN: 978-3-540-37256-1. DOI: 10.1007/978-3-540-37256-1_136. [Online]. Available: https://doi.org/10.1007/978-3-540-37256-1_136.

[25] J. R. Talburt, "3 - entity resolution models," in *Entity Resolution and Information Quality*, J. R. Talburt, Ed., Boston: Morgan Kaufmann, 2011, pp. 63–101, ISBN: 978-0-12-381972-7. DOI: https://doi.org/10.1016/B978-0-12-381972-7.00003-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123819727000038.

[26] M. Allen and D. Cervo, "Chapter 8 - data integration," in *Multi-Domain Master Data Management*, M. Allen and D. Cervo, Eds., Boston: Morgan Kaufmann, 2015, pp. 109–130, ISBN: 978-0-12-800835-5. DOI: https://doi.org/10.1016/B978-0-12-800835-5.00008-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128008355000087.

[27] O. Binette, S. A. York, E. Hickerson, Y. Baek, S. Madhavan, and C. Jones, "Estimating the performance of entity resolution algorithms: Lessons learned through patentsview.org," *The American Statistician*, vol. 77, no. 4, pp. 370–380, Apr. 2023, ISSN: 1537-2731. DOI: 10.1080/00031305.2023.2191664. [Online]. Available: http://dx.doi.org/10.1080/00031305.2023.2191664.

[28] M. Barnes, *A practioner's guide to evaluating entity resolution results*, 2015. arXiv: 1509.04238 [cs.DB]. [Online]. Available: https://arxiv.org/abs/1509.04238.

[29] L. Gagliardelli, G. Papadakis, G. Simonini, S. Bergamaschi, and T. Palpanas, "Gsm: A generalized approach to supervised meta-blocking for scalable entity resolution," *Information Systems*, vol. 120, p. 102 307, 2024, ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2023.102307. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437923001436.