



DEPARTEMENT BEDRYFS- EN SISTEEMINGENIEURSWESE  
DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

VOORBLAD VIR INDIVIDUELE WERKOPDRAGTE - 2016  
FRONT PAGE FOR INDIVIDUAL ASSIGNMENTS - 2016

Persoonlike besonderhede / Personal details	
Studentenommer Student number	12157237
Voorletters en van Initials and surname	KA HAMERSMA
Titel Title	Mr.
Selnommer Cell number	0825524746
Werkopdrag / Assignment	
Modulekode Module Code	RPS420
Werkopdragnommer Assignment number	FINAL PROJECT REPORT
Onderwerp Subject	IMPROVING PROBABILISTIC RECORD LINKAGE WITH A SINGLE-LAYER NEURAL NETWORK
Dosent Lecturer	PROF JW JOUBERT
Datum Date	28/09/2017
Verklaring / Declaration	
<p>1. Ek begryp wat plagiaat is en is bewus van Universiteitsbeleid in hierdie verband</p> <p>2. Ek verklaar dat hierdie my eie oorspronklike werk is</p> <p>3. Waar iemand anders se werk gebruik is (hetsy uit 'n gedrukte bron, die internet of enige ander bron), is dit behoorlik erken en die verwysings ooreenkomstig departementele vereistes gedoen</p> <p>4. Ek het nie 'n ander student se vorige werk gebruik en as my eie ingedien nie</p> <p>5. Ek het niemand toegelaat en sal niemand toelaat om my werk te kopieer met die doel om dit as sy of haar eie werk voor te hou nie</p>	
<p>1. I understand what plagiarism is and I am aware of the University's policy in this regard.</p> <p>2. I declare that this is my own original work</p> <p>3. Where other people's work has been used (either from a printed source, internet or any other source) this has been carefully acknowledged and referenced in accordance with departmental requirements</p> <p>4. I have not used another student's past work to hand in as my own</p> <p>5. I have not allowed and will not allow anyone to copy my work with the intention of handing it in as his/her own work</p>	
Handtekening Signature	
Datum van inhandiging Date of submission	28/09/2017
Kantoorgebruik / For office use:	
Dosent Lecturer	Kommentaar / Comments:
Uitslag Result	
Datum Date	

# Improving probabilistic record linkage with a single-layer neural network

Hammersma, Kris A.

September 28 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Problem awareness . . . . .	5
1.2	Document layout . . . . .	7
1.2.1	Literature review . . . . .	7
1.2.2	Research design . . . . .	7
1.2.3	Model evaluation and comparison . . . . .	8
1.2.4	Conclusion . . . . .	8
<b>2</b>	<b>Literature review</b>	<b>9</b>
2.1	Examples of record linkage . . . . .	9
2.2	Deterministic record linkage . . . . .	10
2.3	Probabilistic record linkage . . . . .	10
2.4	Measuring linkage accuracy . . . . .	12
2.5	Using neural networks for improved accuracy . . . . .	13
2.6	The single-layer perceptron . . . . .	13
<b>3</b>	<b>Record linkage models</b>	<b>15</b>
3.1	Deterministic model . . . . .	15
3.2	Probabilistic model . . . . .	16
3.3	Probabilistic model with single layer perceptron . . . . .	17
<b>4</b>	<b>Model evaluation and comparison</b>	<b>19</b>
4.1	Deterministic model results . . . . .	19
4.2	Probabilistic model results . . . . .	19
4.3	Probabilistic model with refined weight results . . . . .	19
4.4	Discussion . . . . .	20
<b>5</b>	<b>Conclusion</b>	<b>22</b>
	<b>References</b>	<b>23</b>

# List of Tables

1.1	Relative A . . . . .	6
1.2	Relative B . . . . .	6
2.1	A pair of records that need to be matched . . . . .	11
3.1	Records A and B . . . . .	16
3.2	Fellegi-Sunter input weights for records A and B . . . . .	16
3.3	Binary input features and weights for records A and B . . . . .	17

# List of Figures

2.1	Wilson's comparison of precision and recall for each model at different $\theta$ (threshold) values [10]. . . . .	13
2.2	A single layer perceptron [4] . . . . .	14
3.1	The single layer perceptron used to do weight training in this project. . . . .	18
4.1	The traditional and improved probabilistic model precision and recall values for different values of $\theta$ . . . . .	20

# Abstract

Data analysis requires data to be of a high quality. Unfortunately this is not always the case, especially when data is extracted from different data sources. In the case where there is no unique identifier to match data records from multiple data sources alternative methods need to be developed to match the records. Record linkage attempts to do this primarily with deterministic and probabilistic approaches. Deterministic models depend on certain corresponding fields from each record pair to be identical matches to match the record pair together. Probabilistic methods use a set of equations called the Fellegi-Sunter formulae to calculate decision-making weights, which is used to score a record pair on how well they match. If the matching score is above a certain threshold, the record pair is considered to be a match. This project investigates whether the development of a learning algorithm that refines the weights will improve the probabilistic model's matching accuracy. The dataset that was used to train and test the record linkage models was a set of 92650 record pairs, some of which were matches and some of which were non-matches. It was found that a learning algorithm did improve the matching accuracy of the probabilistic model, although it is likely that the increase in the number of input features will improve the matching performance even more.

# Chapter 1

## Introduction

*“Truth will prevail where there is pains taken to bring it to light.”*

— George Washington

### 1.1 Problem awareness

Data analysis is the interpretation of data to answer the question of ‘*what is going on here?*’ [2]. Finding context behind the data is arguably more useful than the data itself. A useful technique for identifying patterns and behaviour in data is the comparison of corresponding data in different *datasets*. A common problem in data analysis is the lack of a *unique identifier* which can link *records* in multiple datasets that refer to a single individual or *entity* [7]. An individual is likely to have the same birth date, gender, name and surname in all the datasets that contain their information and by analysing these *data fields* it is possible to determine whether entries are referring to the same entity. Comparing these entries can lead to the individual’s records being *matched* without the use of a unique ID. This is the essence of all record matching methods.

The means of bringing together two or more separately recorded pieces of information relating to the same individual is known as *record linkage*. The data is obtained from different sources which can have different formats for storing the data, different techniques for capturing the data, and different data quality policies. This leads to data variation when it should theoretically be identical. Record linkage has two key methods for matching data; the *deterministic* approach and the *probabilistic* approach. Deterministic record linkage requires one or more data field to have an exact agreement that is unique to that specific individual, such as an identification number. This method assumes that all data is entered perfectly and error-free, and that unique identifiers are present. Unfortunately such identifiers are not always captured perfectly or are not available due to privacy reasons [8]. Therefore the deterministic matching method will often be unable to match records that refer to the same individual, or instead match records that do refer to different individuals. Probabilistic record linkage is used when a unique identifier is not available or of a poor and inconsistent quality [7]. Instead, to determine whether records are a match, other data fields are compared and links are made based on *how well* the data fields agree [6].

The above-mentioned data inconsistencies often prevent datasets from containing perfect data. This complicates the record linkage process by making it impossible to find exact matches for a single entity in the data. An example of this is where a hospital admission

Table 1.1: Relative A

Name	Surname	City of Birth	Date of Birth	Date of Death	Birth State
Jonathan	Smith	Buffalo	1931-04-21	Unknown	New York

Table 1.2: Relative B

Name	Surname	City of Birth	Date of Birth	Date of Death	Birth State
John	Smith	New York	1931-04-21	2014 - 10 - 26	New York

record for “Mary-Jane Cook” and a vehicle accident registry entry for “Mary Cook” which have a common date are likely referring to the same individual and would prove useful insight if matched. Data matching is further complicated by *different* people having the same name, surname and gender. These records should be determined to be *non-matches*. The challenge then becomes navigating data variation (“Mary” and “Mary-Jane” referring to the same person), formatting differences (“12/03/2016”, “2016-12-03” and “12 March 2016” being the same date), human error (typos) and misleading data similarities (having more than one “John Smith” in the system) to determine clear matches and non-matches.

Take as an example the case of a genealogy system which has the purpose of connecting people by their mutual relatives, without the assistance of an identity number that is unique for each person. Suppose a client of the genealogy system enters information about their family member, Relative A, and that their third cousin, twice removed whom they have never met and have no knowledge about who is also a client of the genealogy system enters information for the same family member, but with different details, as Relative B. The genealogy system now has the task of realising that Relative A and Relative B are in fact the same person, thereby connecting the two clients as family. The difficulty arises when the information entered by the clients differ, either as a result of human error or outdated and incorrect information. Even though it is known that Relative A and Relative B are a match for the same entity, variation in the data prevent the records of being an exact match. In the genealogy example, let us assume that Jonathan Smith was born in the town of Buffalo, New York and is assumed to be alive and well. His cousin whom he met only once many years ago is trying to expand her family tree by using a certain genealogy website and enters all the information she has of her relative in the system (Table 1.1). However, Jonathan later began using the name “John” and lived the remainder of his days in New York, with his wife and children. Years later his wife attempts to document her family tree using the same genealogy website, and enters the information of her now late husband on the site (Table 1.2), hoping to find connections to his relatives. The entries show that although the information entered into the system are similar, the variation in data will prevent a perfect match from being made.

This is an example of one of the pitfalls of the exact or *deterministic* matching approach: matches can only be made if the data is identical. Instead of perfect matches, rules can be made for certain fields to be in a range of values instead of being a specific value. For instance, a rule can be implemented that requires the date of birth to not vary by month or year, but allows it to vary by day. This type of leniency reduces matching accuracy and leads to more incorrect matches being made. Therefore the need arises for a



more intelligent system that will be able to look past the data variation and still be able to identify a strong match, regardless of identical entries. Deterministic record matching is often used when matching accuracy is prioritised, for instance: the tracing of money transfers. In this case, matching incorrectly is possibly worse than not finding a match at all.

A different technique called the *probabilistic* approach can be used instead to calculate a score for a pair of records entries and classify them as a match or a non-match accordingly. This approach however does not weigh each field as equally important. In matching two entities in New York City as the same person, it is extremely unlikely that the fact that both entities reside in New York will contribute to the accuracy of the match, as it is one of the most densely populated cities in the world. Instead, the probabilistic approach will in this case assign a smaller weight to the city field and larger weights to matching values in the name, surname and date of birth fields. These weights are fixed values calculated using the Fellegi-Sunter probabilistic record linkage equations [5]. If the sum of the weights are more than a certain threshold value, it is classified as a match. It is hypothesized that it is possible to improve the probabilistic approach by adding a neural network algorithm which adjusts the weights after each iteration, teaching the system to make better matches [10]. This is an improvement on the traditional probabilistic approach, as it does not solely rely on the static values calculated by the Fellegi-Sunter equations, but rather optimises itself continuously to reduce matching errors. To this end the application of a neural network algorithm to the traditional probabilistic record linkage method has been researched and developed. By developing this method data matches can be made more accurately and improved insight into the inner workings of each dataset can be gained.

The aim of this project is the application of a simple neural network to a probabilistic record linkage model, and the comparison of the performance of this new model with those of the traditional deterministic and probabilistic models. The dataset used for this project is typical cash-in-transit data, where records of cash sent and records of cash received are matched to ensure no money is lost or stolen. This type of data is currently being matched with deterministic techniques because of the risks involved with matching data incorrectly. Any increase of matching performance while maintaining matching integrity should therefore prove useful.

## 1.2 Document layout

### 1.2.1 Literature review

A literature review has been done regarding the process of record linkage. The three main record linkage approaches relevant to this project are described in detail in the following chapter. The measures for matching accuracy are outlined, with examples of their use and how to compare them. The concept of a neural network algorithm follows, with the explanation of its simplest form: the single layer perceptron.

### 1.2.2 Research design

The deterministic record linkage model requires logically relevant rules for classifying record pairs since it only accepts exact values when comparing pairs. These were carefully selected according to the current matching rules used on this data source. The probabilistic record linkage model agreement and disagreement weights are calculated using the traditional probabilistic formulas, as mentioned in Chapter 2.3. A function of these weights is used to determine whether a record pair is a match or not. The third model

requires a function for deciding how much to adjust the agreement and disagreement weights, also known as a *learning rate* and uses the same function to determine if a record pair is a match. A training dataset was used to feed this model. All models were used on a test dataset and had recall and precision values calculated and compared as a measurement of accuracy. It was originally expected that the probabilistic record linkage model with a simple learning algorithm will show considerable improvement on the traditional deterministic and probabilistic approaches.

### 1.2.3 Model evaluation and comparison

The results show that the deterministic model found only a small portion of the matches, but with great accuracy. The probabilistic model found a significantly larger number of matches in the data, without sacrificing much in terms of accuracy. Contrary to initial expectations the learned probabilistic model was only a marginal improvement on the traditional probabilistic model for certain threshold values when applied to this dataset.

### 1.2.4 Conclusion

These results could be caused by the low number of inputs in the dataset, as this causes the matching accuracy to be more dependent on data quality. Because of the few input features, the weights adjusted to heavily favour the date field comparison to determine whether a record pair was a match, while favouring the value comparison less. This implies that to decide whether a given transaction was a match, more importance is given to the dates of the transfer than the value of the transfer. Another possibility is that the training set that was used to train the third model did not properly represent the test set, as the test set contained a much lower proportion of matches than the training set. It is possible that, given a dataset with more input values, the addition of a learning algorithm will have a more significant improvement on the traditional probabilistic approach as shown in this project.

## Chapter 2

# Literature review

To better understand record linkage, a few examples of its use are given. The two traditional record linkage methods as well as a third approach using a learning algorithm are investigated and explained in detail. Matching accuracy measures are described and its application shown with the help of examples. An approach for using a simple neural network to refine probabilistic record linkage is given.

### 2.1 Examples of record linkage

Record linkage is a tool used in many industries where data needs to be matched. Some examples follow here.

1. With the increase in available patient information, medical research often requires that patient data from multiple databases be combined. Due to privacy concerns, only a handful of countries make use of a unique national patient identifier. In this case, a mixture of the deterministic and probabilistic approaches are usually applied. Bell et al. [3] used the traditional probabilistic record linkage approach by calculating the Fellegi-Sunter values for each linking variable when matching data records of very low birthweight infants in California and delivery or birth-related hospital admissions in the 1980's. Regardless of missing data entries and corrupt data files they found that more than 96% of the records could be determined to be either clear matches or non-matches.
2. In genealogy people attempt to discover the identities of their relatives and ancestors. Connections are made by comparing individual names, surnames, places of birth, dates of birth and death and so forth. Sometimes this results in multiple entries in the system that refer to the same person. The goal of genealogical systems is to match these individuals as the same person, thereby linking relatives. The problem in this environment lies in outdated or incorrect data. Relatives can change their name, wrong dates can be remembered, and variation in spelling and human error can hamper exact matches. To circumvent this, Wilson [10] uses a probabilistic approach to match relatives, with a learning algorithm that adjusts the matching weights after each iteration, depending on whether the match was correctly or incorrectly made. His work indicated that a simple neural-network could greatly improve the matching accuracy in the genealogical record linkage environment.
3. To keep track of cash-in-transit between branches, cash centres and ATM machines, banks record the arrival and departure of the assets. The issue and reception records

of the assets need to be exactly matched to trace the cash flow to ensure there is no theft or fraud taking place. As such, very strict deterministic rules are applied, as the danger of making incorrect matches is much worse than the failure of making a correct match. The result is a very complex system that applies certain rules only when certain conditions are met. An example of this would be if the value of cash sent and the delivery point match with the value of cash received and the collection point, then a leniency of several days is given for the delivery and collection dates to match. This is supposedly to factor in the time needed to deliver the cash, but seems fallible as false matches can be made between the records if several cash-in-transit values were similar in the leniency period. An improvement on such a system will clearly benefit any cash-in-transit system.

Data inconsistencies in these examples necessitate the use of systems that use probabilistic methods to classify the data as a match or a non-match and processes the data accordingly. In the case of the third example, it is possible to use both deterministic and probabilistic methods to determine whether records should match. In this case one can apply a very strict precision threshold for the probabilistic method to make a match.

## 2.2 Deterministic record linkage

Deterministic record linkage compares specific fields for exact matches to see whether a pair of records are a match. This requires that data be perfectly captured and that these fields will be sufficient to describe a unique entity [7]. In other words, only one entity can possibly have a unique combination of values, and if another entity has this combination, it must be the same entity, and therefore be a match. Unfortunately the lack of a unique identifier, human-error, data variation, incorrect information and less than ideal data capturing leads to suboptimal matching accuracy. As mentioned, certain conditional rules can be applied to deterministic matching algorithms, but the list of conditions tend to grow exponentially with each added rule and each additional rule need extraordinary reasoning to be included, as each conditional rule also adds the possibility that more pairs will be matched that should not be matched.

## 2.3 Probabilistic record linkage

In contrast with the deterministic approach, probabilistic record linkage does not require that all data fields be exact matches. Instead, to evaluate a pair of records their data fields are compared individually and a score generated for each field that is a match and a negative score generated for each field that is a non-match. These scores are then added to determine whether the pair of records themselves are matches. A set of equations were developed by Fellegi and Sunter to formalise the scoring approach commonly used in probabilistic record linkage [5]. This traditional probabilistic approach estimates two *field agreement probabilities* using the Fellegi-Sunter equations for each pair of fields that needs to be compared, with each pair  $i = 1..n$  [5, 10]:

$$m_i = a_i^m / c_m \tag{2.1}$$

$$u_i = a_i^u / c_u \tag{2.2}$$

where:

Table 2.1: A pair of records that need to be matched

Name	Surname	City of Birth	Date of Birth	Date of Death	Birth State
Jonathan	Smith	Buffalo	1931-04-21	Unknown	New York
John	Smith	New York	1931-04-21	2014-10-26	New York

- $m_i =$  the probability field  $i$  is the same on a matching pair,
- $u_i =$  the probability field  $i$  is the same on a non-matching pair,
- $a_i^m =$  the number of matching pairs that have the same field  $i$ ,
- $a_i^u =$  the number of non-matching pairs that have the same field  $i$ ,
- $c_m =$  the total number of matching pairs, and
- $c_u =$  the total number of non-matching pairs.

To measure a *score* for a given pair of records, one of two weights are added for each field  $i$ . If the two records agree on the field, an *agreement weight* is added. If they disagree, a *disagreement weight* is added. If one or both records have no data for the field, then neither weight is added. The score for the record pair is the sum of these weights. The agreement and disagreement weights for each field  $i$  can be calculated with:

$$w_{\text{agree}}^i = \ln(m_i/u_i) \quad (2.3)$$

$$w_{\text{disagree}}^i = \ln(1 - m_i/1 - u_i) \quad (2.4)$$

Finally, to determine whether the record pair should be classified as a match or a non-match, the score is compared to a *decision threshold* value,  $\theta$ . If the score is above the  $\theta$  value, it is classified as a match, otherwise it is labelled a non-match. Let us revisit the genealogical matching example of Jonathan Smith where we want to compare two sets of records in Table 2.1 with each column being field  $i = 1..6$ , by calculating the agreement and disagreement weights for each field  $i$  that is going to be compared, summing the applicable weights together for all the pairs and comparing it with some value  $\theta$ . As an example, let us calculate the field agreement weights for the Surname data field, where  $i = 2$ . Let us assume that statistical sampling has shown that 1% of entries on the genealogy system bear the surname ‘‘Smith’’, and that the data accuracy of the entries captured in the genealogy system is 95%. This means that 95 matching pairs out of 100 will have the same values in their Surname fields and there exists a probability that 1 out of every 100 people will also be called ‘‘Smith’’. This means we can compute the field agreement and field disagreement probabilities for the Surname data field as follows:

$$m_2 = 95.0/100 = 0.95 \quad (2.5)$$

$$u_2 = 1.0/100 = 0.01 \quad (2.6)$$

And consequentially the field agreement weights:

$$\begin{aligned} w_{\text{agree}}^2 &= \ln(m_2/u_2) \\ w_{\text{agree}}^2 &= \ln(0.95/0.01) \\ w_{\text{agree}}^2 &= 4.554 \end{aligned} \quad (2.7)$$

and

$$\begin{aligned}
 w_{\text{disagree}}^2 &= \ln(1 - m_2/1 - u_2) \\
 w_{\text{disagree}}^2 &= \ln(1 - 0.95/1 - 0.01) \\
 w_{\text{disagree}}^2 &= -2.986
 \end{aligned}
 \tag{2.8}$$

Since the record pair in Table 2.1 have matching Surnames (and therefore are in agreement) the input value for  $i = 2$  will have be  $w_{\text{agree}}^2 = 4.554$ . For this record pair, the disagreement weight is disregarded as these fields match. All the field scores will be calculated and added together. This will be the total score for the record pair, and this score will be compared to  $\theta$ . If the record pair scores above  $\theta$ , the pair is considered a match. If the pair scores below  $\theta$ , it is considered a non-match.

## 2.4 Measuring linkage accuracy

The value of  $\theta$  is important as it essentially determines the strictness of the matching algorithm. A lower  $\theta$  allows weaker matches to be matched incorrectly, while a higher  $\theta$  will increase the accuracy, while excluding some record pairs that are actual matches, but score low due to data inconsistency. The accuracy of a record linkage system can be measured in *precision* and *recall*. Precision is the percent of pairs that scored above the decision threshold  $\theta$  that are correctly matched. Recall is the percent of matching pairs that scored above  $\theta$ . Stated differently, precision is the percent of pairs that were matched correctly and recall is the percent of actual matches that were found to be matches [10]. Take an example with 100 matching pairs and 200 non-matching pairs. If 90 of the matching pairs and 20 of the non-matching pairs scored above  $\theta$ , then the precision can be said to be:

$$\textit{precision} = 90/(90 + 20) = 81.8\%
 \tag{2.9}$$

Similarly, recall would be:

$$\textit{recall} = 90/100 = 90\%
 \tag{2.10}$$

To increase recall a smaller value for  $\theta$  can be chosen. This will also decrease precision as the classification requirements for a match will be less strict. This means that if a larger value for  $\theta$  is used that precision will increase and recall will decrease. As the value of  $\theta$  can be chosen arbitrarily, each record linkage model will yield different precision and recall values for each  $\theta$  value. Wilson [10] compares four record matching models based on their precision and recall values at certain  $\theta$  points. This results in a graph, Figure 2.1, that shows the range of precision and recall values a model can possibly achieve. A technique that can improve both precision and recall is the application of a learning algorithm which may generate and adjust the agreement and disagreement weights to make less classification mistakes, thereby improving the record linkage accuracy. A learning algorithm will generally compare the output of a system with the expected results, and if the output error is large enough, adjust the system's decision variables. One such algorithm that can be used to fine-tune the agreement and disagreement weights of the linkage system is a neural network algorithm, where the agreement and disagreement weights act as the system decision variables. Once these different models are built, it is possible to compare them using the graph in Figure 2.1. As the goal of any record linkage model is to attain

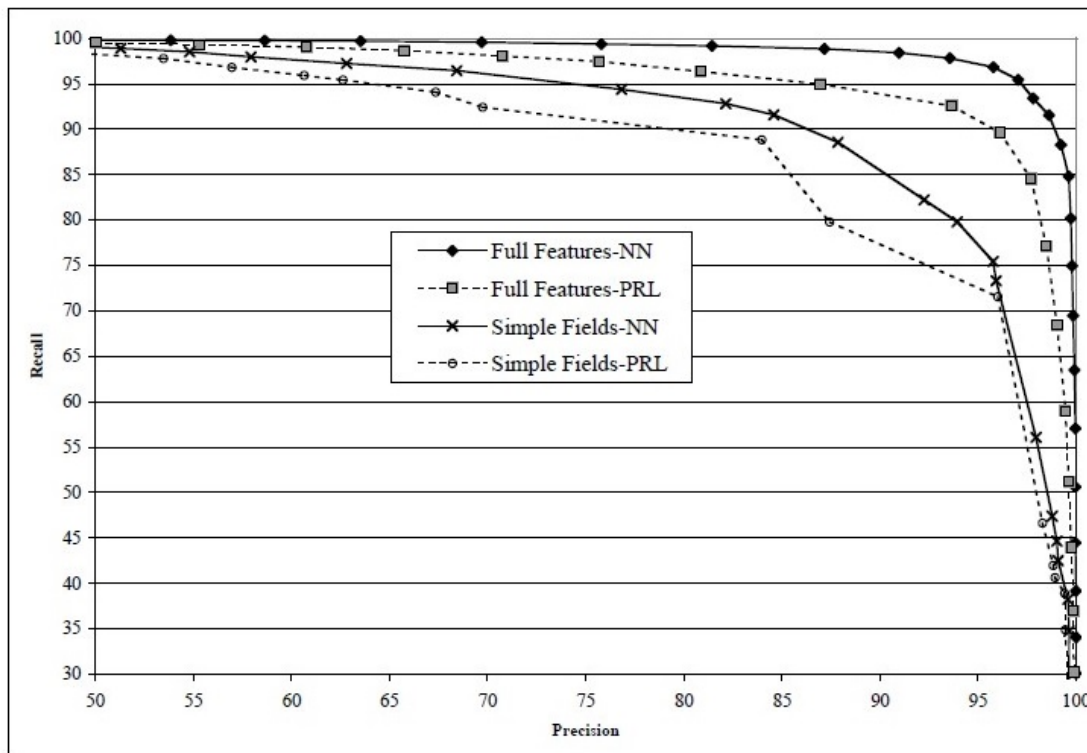


Figure 2.1: Wilson's comparison of precision and recall for each model at different  $\theta$  (threshold) values [10].

the highest possible precision and recall values for any given  $\theta$ , the model that produces the widest curve is the model that yields the best matching results.

## 2.5 Using neural networks for improved accuracy

Neural networks are adaptive models that can learn the *parameters* of a population by processing a significant number of exemplars. These networks are constructed from small units or *neurons* that are linked by a set of weighted connections [1]. In the case of probabilistic record linkage it is possible to use an algorithm that evaluates the matching errors and adjusts the agreement and disagreement weights accordingly, much like a neural network adjusts its parameters. Therefore a simple neural network proves to be an ideal weight improvement algorithm that can yield greater performance [10].

## 2.6 The single-layer perceptron

Usually when a decision of any type needs to be made, several influential factors are taken into consideration, typically with a varying amount of regard for each factor. For instance, when deciding to take an umbrella to work, one might disregard the cloudy skies, but one is unlikely turn a deaf ear to a news report of a coming storm. It is in this sense that a perceptron functions as a decision-making model. It adjusts the input weights after each attempt of classifying the input by comparing its output with the desired output. A perceptron is the simplest form of a neural network that employs a supervised learning

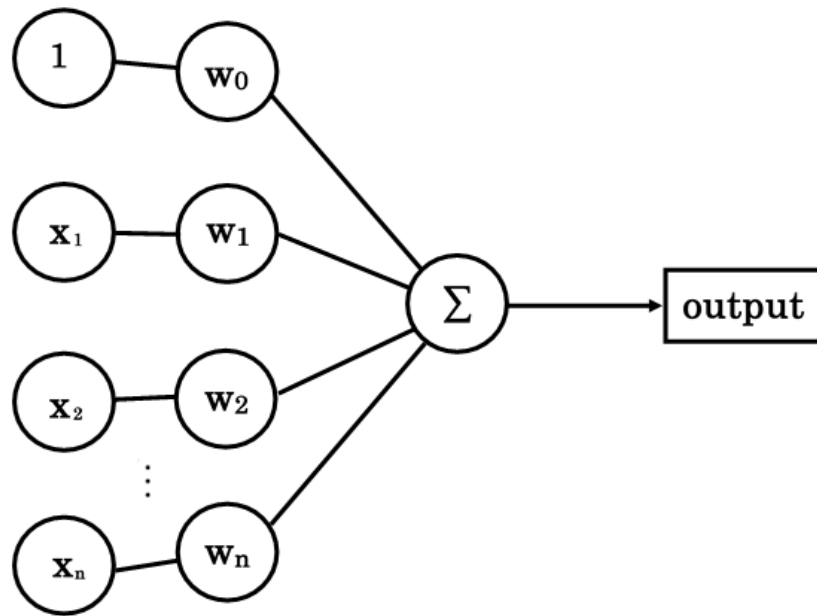


Figure 2.2: A single layer perceptron [4]

rule (Figure 2.2). It is a single-layer neural network that takes several weighted inputs and calculates a single value for each set of inputs. It then checks the calculated value against the correct value and adjusts the input weights based on the error [9] and a learning rate value or function. Suppose that for each record pair the perceptron is given the score from each record pair field and the correct answer (that is to say, whether the record pair is a match or a non-match) and it calculates the record pair score. It then compares the record pair score against the correct value that it was given. Based on how much of an error was made by its guess of the answer it then adjusts each weight by a fixed learning rate value. This process is repeated for thousands of record pairs until the weights become as accurate as possible and show little sign of improving further. These refined weights can then be used in the probabilistic model instead of the Fellegi-Sunter values to calculate a score for each record pair and classify them as matches and non-matches. Essentially this means that by feeding the learning algorithm a large dataset of record pairs with known classifications (whether it is a matching pair or non-matching pair) it will teach itself which inputs to assign more weight to when deciding whether it is a match or a non-match, thereby possibly increasing the record linkage accuracy [10].

The goal of this project was to compare the accuracy of the three record linkage methods that have been mentioned thus far: the deterministic approach, the probabilistic approach and the probabilistic approach with a weight-adjusting perceptron algorithm.



## Chapter 3

# Record linkage models

It is hypothesised that an added weight-adjustment algorithm will produce more accurate record linkage when used with a probabilistic record linkage model. To test whether a learning algorithm such as a perceptron will be able to improve the accuracy of the deterministic and traditional probabilistic approaches, the following three record linkage models were built and tested:

1. A deterministic record linkage model.
2. A traditional probabilistic record linkage model.
3. A probabilistic record linkage model with a single-layer perceptron.

The models were given a large number of labelled record pairs, some of which are matches and some of which are non-matches and were given the task of determining whether the record pair is a match or a non-match. The data that was used to train and test the different methods was cash-in-transit dataset used by a bank to track cash flow. The training data used for each set contained a random sample of 70000 record pairs. The test dataset contained 22650 record pairs. Precision and recall were calculated for the results of each model for various values of  $\theta$ . This gave an indication of the accuracy of the models, which were then compared in a similar manner to the models shown in Figure 2.1. It was expected that there would be a clear improvement on both measurements with the weight-adjustment algorithm.

### 3.1 Deterministic model

The deterministic model makes use of certain fields having exact matches to determine whether a pair of records are a match or a non-match. The training set had a series of times, dates, location names, monetary values, scanning codes and other string values that belong to each of the two records that need to be classified as a match or a non-match. Each record's field is compared with the corresponding field of the record it is being compared to and if a single pair of fields are not a match, it is considered a non-match. If all the fields match, then the model considers the pair of a records a match. As there is no threshold value used in the deterministic model, only a single precision and recall value can be calculated. Using more fields to match records with will more most likely result in a higher precision, but result in lower recall. This means that fewer matching errors will be made, but fewer matches will be found as well as the requirements to make a match is more strict.

Table 3.1: Records A and B

Record	Scan Code	Date	Time	Location	Amount	Area
A	98801472617	2016-06-19	19:01:24	Johannesburg	190350	H12
B	98801472617	2016-06-19	19:01:24	Johannesburg	190340	I49

As an example of how this model works, consider the following records A and B that need to be matched (Table 3.1). The deterministic model will compare each field from record A with each field from record B. If a set of corresponding fields does not match, it regards the record pair as a non-match. In the case of Table 3.1, as the *Amount* and *Area* fields do not match, the model classifies these pairs as a non-match.

## 3.2 Probabilistic model

The probabilistic model uses fixed values calculated with the Fellegi-Sunter equations and a  $\theta$  value to determine whether the pair should be labelled a match or a non-match. As the  $\theta$  value can be chosen depending on precision or recall is prioritised, more than one threshold value needs to be used to compare this model with the alternatives. The training data was used to determine the  $m_i$  and  $u_i$  values for each pair of fields  $i$  that were being compared. These were in turn used to calculate the agreement weights  $w_{\text{agree}}^i$  and  $w_{\text{disagree}}^i$  for each field  $i$ . The model is run a few hundred times, each time with a different threshold value, using the test dataset as input. For each threshold value precision and recall is calculated. Using the example from Table 3.1, let us assume that the following input weights have been calculated for each field using the dataset records A and B originate from and the Fellegi-Sunter equations outlined in Chapter 2.3:

Table 3.2: Fellegi-Sunter input weights for records A and B

Field	Agreement Weight	Disagreement Weight
Scan Code	+4.19	-3.21
Date	+3.78	-2.33
Time	+5.43	-0.67
Location	+1.77	-6.90
Amount	+5.12	-1.38
Area	+1.52	-0.99

Taking the fields from Table 3.1 as input, we can calculate the score for this record pair by summing the applicable weights. As the *Amount* and *Area* fields do not match, their disagreement weights will be used. The rest of the fields match and therefore their agreement weights will be used. The score of this record pair is then:

$$\text{score} = 4.19 + 3.78 + 5.43 + 1.77 - 1.38 - 0.99 = 12.8 \quad (3.1)$$

This means that for any  $\theta$  value below 12.8, the record pairs will be classified as match. For  $\theta$  above 12.8, the record pair will be considered a non-match. Essentially the  $\theta$  value is acting as a passing minimum to be classified as a match. By adjusting the threshold

value, the strictness of the model is being changed, essentially changing the size of the net being cast by the model. By throwing a wider net, more correct matches will be made, but more incorrect and unwanted matches will also be pulled in. This model is used with a wide range of  $\theta$  values, as each  $\theta$  will yield a different precision and recall value. Each  $\theta$  value is used to classify all the records in the dataset.

### 3.3 Probabilistic model with single layer perceptron

The probabilistic model with a single layer perceptron uses the Fellegi-Sunter equations to determine the initial weight agreement values, but then adapts those values by checking its result with the labelled classification each time and changing the weights based on the error and the input that was given. To train this model, the inputs were transformed into a set of binary features. For a set of  $n$  fields that need to be matched,  $2n$  binary input features are created. The first feature tests whether a pair of fields are a match, and the second feature tests whether that same pair of fields is a non-match. Essentially this means that when fields  $i$  are a match:  $f_i = 1$  and  $f_{i+1} = 0$ . Adding to this the bias input required by a neural network, let  $f_{2n+1} = 1$ . Each weight that was calculated previously using the Fellegi-Sunter equations is now used with each corresponding feature to determine the record-pair score. The bias weight  $w_{2n+1}$  is assigned a random starting value. Returning to the example from Table 3.1, we know that the first four fields were a match and the last two were not. Seeing as there are six fields in total, we will have twelve binary inputs and one bias input set to 1.

Table 3.3: Binary input features and weights for records A and B

Input	Value	Weight ( $w_i$ )
$f_1$	1	+4.19
$f_2$	0	-3.21
$f_3$	1	+3.78
$f_4$	0	-2.33
$f_5$	1	+5.43
$f_6$	0	-0.67
$f_7$	1	+1.77
$f_8$	0	-6.90
$f_9$	0	+5.12
$f_{10}$	1	-1.38
$f_{11}$	0	+1.52
$f_{12}$	1	-0.99
$f_{13}$ (Bias)	1	+0.1

The weights are changed after each record-matching attempt by taking the model's input and magnitude of error into consideration. If the input feature was 0, then its weight is not changed. A fixed learning rate value is used to determine how sensitive the weight adjustment is. To determine the error made by each iteration, the target is defined as being either 0 or 1, that is, a match or a non-match. A logistic function converts the score to a value between 0 and 1, with the error being the difference between the target and the transformed score value. Weights are adjusted by the error multiplied with some constant

learning rate. Assuming that records A and B are known to be a match, the target value would be 1. The score, now with an added bias input, would be calculated as:

$$score = \sum_{i=1}^{13} (f_i)(w_i) = 12.9 \quad (3.2)$$

The logistic function then delivers:

$$output = \frac{1}{1 + e^{-12.9}} = 0.999997502 \quad (3.3)$$

The error is:

$$error = 1 - 0.999997502 = 2.5 \times 10^{-6} \quad (3.4)$$

Taking a constant learning rate of 0.01, the weight adjustment is:

$$w_i = w_i + (0.01)(2.5 \times 10^{-6})(f_i) \quad (3.5)$$

This algorithm forces the weights to be adjusted of only the inputs that were used, as  $f_i$  would be zero otherwise and the weights will remain unchanged. The perceptron (Figure 3.1) uses the training dataset of 70000 record pairs to train the weights. After the weights are calibrated they are then used in a probabilistic model on the 22650 row test set, once again for a few hundred values of  $\theta$ , just as in the regular probabilistic model.

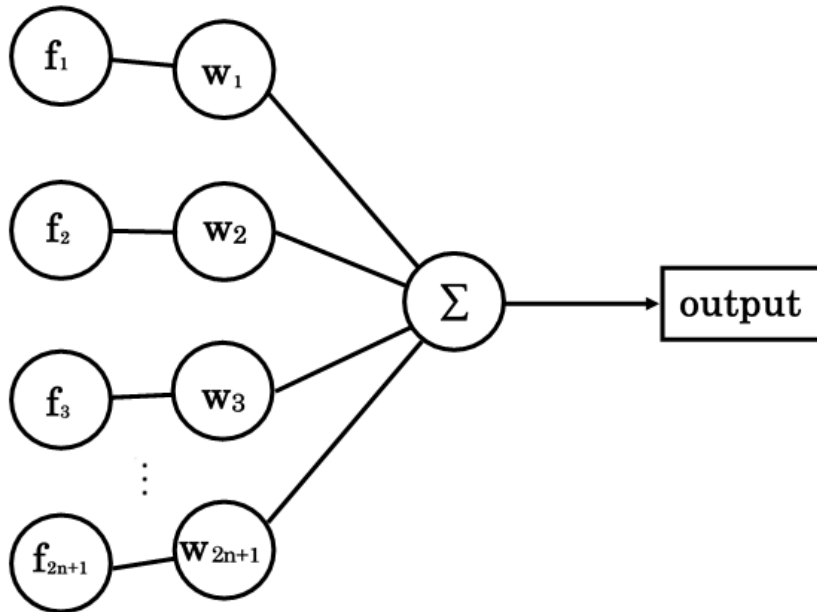


Figure 3.1: The single layer perceptron used to do weight training in this project.

## Chapter 4

# Model evaluation and comparison

The test data set contained 22650 record pairs of which 9348 were matches.

### 4.1 Deterministic model results

The deterministic model correctly matched 2022 record pairs out of 2083 attempts. This means it achieved a precision of:

$$\begin{aligned}\text{precision} &= 100(2022/2083) \\ &= 97.07\%\end{aligned}\tag{4.1}$$

and recall:

$$\begin{aligned}\text{recall} &= 100(2022/9348) \\ &= 21.63\%\end{aligned}\tag{4.2}$$

This shows that while the deterministic model was extremely accurate in the attempts it made to match record pairs, it missed a large portion of matches. This is usually preferred with this type of dataset because of the risks associated with matching record pairs incorrectly.

### 4.2 Probabilistic model results

The traditional probabilistic model proved to be very effective (Figure 4.1), with a mixture of high precision and recall values for the different threshold values. Given a precision level of 97.07% the probabilistic model would have a recall level of approximately 92%, a significant improvement over the deterministic model's 21.63%.

### 4.3 Probabilistic model with refined weight results

The improved probabilistic model (Figure 4.1) showed an improvement on the traditional probabilistic model, but only for precision values lower than 94%. This implies that for this dataset more matches will be found at the same level of precision by using the improved model. Even though the improvement seems marginal percentage wise, when considering the large number of cash-in-transit records that need to be matched, any improvement will have a significant money-value impact.

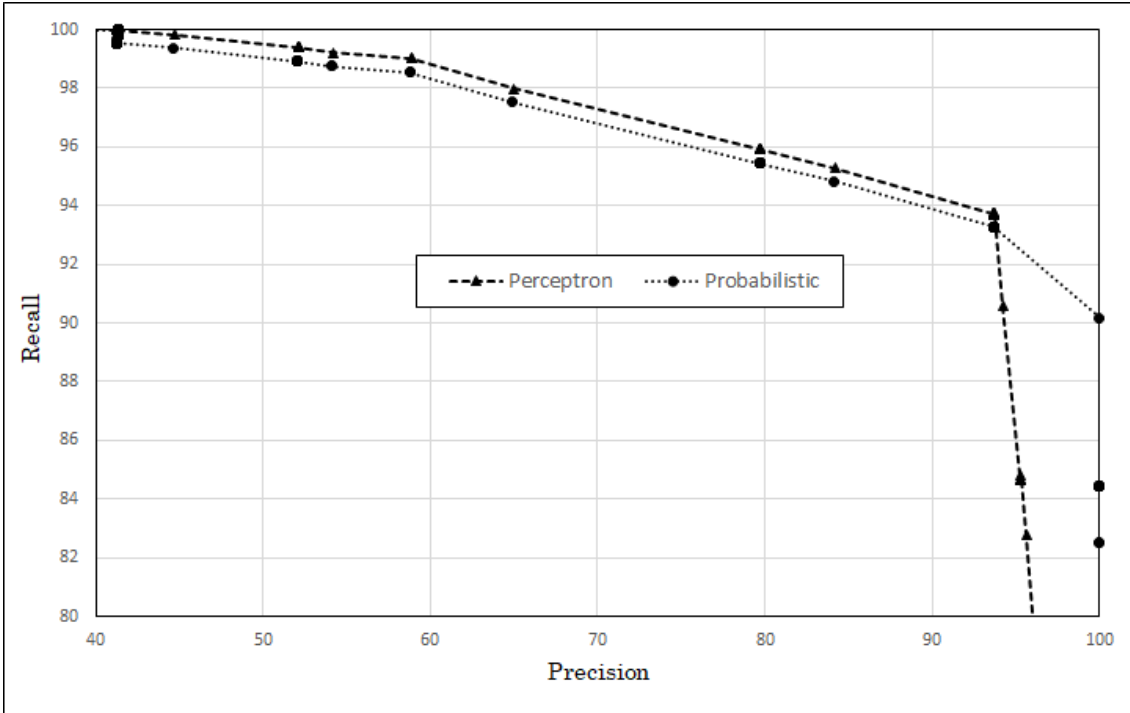


Figure 4.1: The traditional and improved probabilistic model precision and recall values for different values of  $\theta$ .

## 4.4 Discussion

It was established that the deterministic approach was predominantly used in cash-in-transit record linkage due to the importance of maintaining a high accuracy when matching. This was proved to be at the cost of recall: only 21.63% of all matches were found. However, it was shown that the traditional probabilistic model yielded results far superior to the deterministic model, by achieving 100% precision while finding approximately 90% of the matches. Figure 4.1 indicates that the perceptron refinement produced a recall range that was higher than the traditional probabilistic model, but only for precision values below 94%. This indicates that for the dataset used for the purpose of this project, the traditional probabilistic approach would be sufficient as precision is prioritised above recall. Essentially, the refinements did show an improvement in record matching, but mostly in terms of recall. It is possible that the learning algorithm can be improved to better refine the input weights and improve the model in terms of precision as well. There are several ways that a weight adjustment algorithm can be used to improve probabilistic matching methods, including:

1. Increasing the number of fields that can be compared.
2. Increasing the number of derived input features.
3. Investigating more effective learning function alternatives.

By including more data (perhaps even fields that seem irrelevant to the human eye) to be compared, the learning algorithm has more inputs to choose from when assigning weights, ultimately increasing its options. When irrelevant or superfluous features are fed into the learning algorithm, it should adjust the weights of those features to ultimately

disregard features that show no correlation to accurate matching. It is therefore advised that as many features as possible be defined and fed into the probabilistic model to add more depth and dimensions for it to work with. This also holds for features that are derived from the fields. An example of this would be calculating the difference between dates or the distance from locations. Features such as these, although not strictly “matchable” features, can also deliver insight and show correlations that can not easily be spotted. Arguably any additional dimension can help the learning algorithm refine its decision-making weights even more.

## Chapter 5

# Conclusion

The problem of matching records from different sources without a unique identifier is a challenge which is prevalent in many industries. By developing better matching techniques, more insightful data interpretations are possible, decision-making ability may be improved, and matches can be made without requiring unique identifiers or primary keys. Traditional probabilistic record linkage methods, while useful, are not as efficient as they could be. It is possible that the decision-making weights can be refined with the use of a learning algorithm. Three models were built and compared using different record linkage methods to test this hypothesis. Cash-in-transit data was used to train and test the accuracy of the models. This proved interesting in the sense that precision is prioritised above recall when matching cash-in-transit data. This is understandable as mismatches of cash movement is arguably worse than not making matches at all. To this end, deterministic matching is primarily used for this type of data. The results indicated that a traditional probabilistic model yields a massive matching accuracy improvement in terms of recallability when compared with the deterministic record linkage model, without sacrificing any matching accuracy. It was also shown that the use of a learning algorithm to refine the linkage weights improved the traditional probabilistic model, but only marginally. It is important to keep in mind that even a small increase in matching accuracy and recall will lead to a significant increase in the number of matches made. The learning algorithm application can be improved by increasing the number of input fields, derived input features, and investigating different learning algorithm techniques. The combination of learning algorithms and record linkage models merits further investigation as a method for increasing matching accuracy, as many systems exist that would benefit from it.



# Bibliography

- [1] Abdi, H., Valentin, D., & Edelman, B. (1999). *Neural networks*.
- [2] Behrens, J. (1997). Principles and procedures of exploratory data analysis. *Psychological Methods*, 2(2), 131–160.
- [3] Bell, R. M., Keeseey, J., & Richards, T. (1994). The urge to merge: linking vital statistics records and medicaid claims. *Medical Care*, 32(10), 1004–1018.
- [4] DEEPLARNING4J (2017-05-15). Introduction to Deep Neural Networks. <https://deeplearning4j.org/neuralnet-overview>.
- [5] DuVall, S., Kerber, R., & Thomas, A. (2010). Extending the Fellegi—Sunter probabilistic record linkage method for approximate field comparators. *Journal of Biomedical Informatics*, 43, 24–30.
- [6] Fleming, M., Kirby, B., & Penny, K. (2012). Record linkage in Scotland and its applications to health research. *Journal of Clinical Nursing*, 82, 2711–2721.
- [7] Mason, C. & Tu, S. (2008). Data linkage using probabilistic decision rules: a primer. *Birth Defects Research (Part A): Clinical and Molecular Teratology*, 82(11), 812–821.
- [8] Tromp, M., Ravelli, A., Bonsel, G., Hasman, A., & Reitsma, J. (2011). Results from simulated data sets: probabilistic record linkage outperforms deterministic record linkage. *Journal of Clinical Epidemiology*, 64, 565–572.
- [9] Wallisch, P., Lusignan, M., Benayoun, M., Baker, T., Dickey, A., & Hatsopoulos, N. (2009). *Matlab for neuroscientists*.
- [10] Wilson, D. (2010). Beyond probabilistic record linkage: using neural networks and complex features to improve genealogical record linkage. *IEEE: Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011*.

**Department of Industrial & Systems Engineering**  
**Final Year Projects**

**Identification and Responsibility of Project Sponsors**

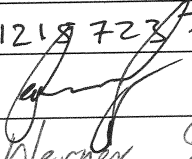
All Final Year Projects are published by the University of Pretoria on *UPSpace* and thus freely available on the Internet. These publications portray the quality of education at the University and have the potential of exposing sensitive company information. It is important that both students and company representatives or sponsors are aware of such implications.

**Key responsibilities of Project Sponsors:**

A project sponsor is the key contact person within the company. This person should thus be able to provide the best guidance to the student on the project. The sponsor is also very likely to gain from the success of the project. The project sponsor has the following important responsibilities:

1. Confirm his/her role as project sponsor, duly authorised by the company. Multiple sponsors can be appointed, but this is not advised. The duly completed form will be considered as acceptance of sponsor role.
2. Review and approve the Project Proposal, ensuring that it clearly defines the problem to be investigated by the student and that the project aim, scope, deliverables and approach is acceptable from the company's perspective.
3. Review the Final Project Report (delivered during the second semester), ensuring that information is accurate and that the solution addresses the problems and/or design requirements of the defined project.
4. Acknowledges the intended publication of the Project Report on UP Space.
5. Ensures that any sensitive, confidential information or intellectual property of the company is not disclosed in the Final Project Report.

**Project Sponsor Details:**

<b>Company:</b>	Fourier - E
<b>Project Description:</b>	Applying Neural Networks to Refine Probabilistic Record Linkage Accuracy
<b>Student Name:</b>	Kris Hamersma
<b>Student number:</b>	12197237
<b>Student Signature:</b>	
<b>Sponsor Name:</b>	Werner Schoeman
<b>Designation:</b>	Manager Process & BI
<b>E-mail:</b>	SchoemanW@fourier.co.za
<b>Tel No:</b>	082 0766 314
<b>Cell No:</b>	
<b>Fax No:</b>	
<b>Sponsor Signature:</b>	