Machine Learning in Logistics: Machine Learning Algorithms

Data Preprocessing and Machine Learning Algorithms

Viktor Andersson

Computer Game Programming, bachelor's level
2017

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Bachelor Thesis

Machine Learning in Logistics: Machine Learning Algorithms
Data Preprocessing and Machine Learning Algorithms

Author: Viktor Andersson
Examiner: Patrik Holmlund
Supervisor: Mario Toffia

July 2, 2017
Luleå University of Technology, Skellefteå
Department of Computer Science, Electrical and Space Engineering
Abstract

Data Ductus is a Swedish IT-consultant company, their customer base ranging from small startups to large scale cooperations. The company has steadily grown since the 80s and has established offices in both Sweden and the US.

With the help of machine learning, this project will present a possible solution to the errors caused by the human factor in the logistic business. A way of preprocessing data before applying it to a machine learning algorithm, as well as a couple of algorithms to use will be presented.

Sammanfattning

Data Ductus är ett svenskt IT-konsultbolag, deras kundbas sträcker sig från små startups till stora redan etablerade företag. Företaget har stadigt växt sedan 80-talet och har etablerat kontor både i Sverige och i USA.

I would like to thank everyone who has been a part of this project for the last two months. I can not mention all people in this report, however special thanks to Rasmus Lind and Erik Hedman who both have been my colleague in making this large scale project possible. This project would not have been possible without them.

I would like to thank Patrik Holmlund for whom has been giving continuous feedback on my report. I would like to throw out final thanks to Mario Toffia, who has been my supervisor during the project.
# Contents

1 Introduction .................................................. 1
   1.1 Data Ductus ............................................ 2
   1.2 Artificial Intelligence .................................. 2
      1.2.1 Machine Learning ................................... 2
      1.2.2 Machine Learning Process and Terminology ....... 3
   1.3 Limitations .............................................. 7
   1.4 Background .............................................. 7
      1.4.1 Python ............................................. 7
      1.4.2 Scikit Learn ....................................... 8
      1.4.3 Pandas ............................................ 8
      1.4.4 Assignment Background ............................ 8
   1.5 Goals and Purpose ..................................... 9
   1.6 Method .................................................. 9
   1.7 Social, Ethical and Environmental Considerations ... 10
   1.8 Abbreviations and Terms .............................. 11

2 Design and Implementation .................................. 12
   2.1 Pandas .................................................. 12
   2.2 Data Preprocessing .................................... 12
      2.2.1 Roster Update Process ........................... 12
      2.2.2 Encoding .......................................... 13
         2.2.2.1 Encoding with Roster data .................. 13
         2.2.2.2 Dummy Encoding with Random Data ........ 14
   2.3 Training Set & Test Set .............................. 15
   2.4 Algorithms ............................................. 15
      2.4.1 Random Forest, Naive Bayes, and Logistic Regression ... 16
      2.4.2 Printing Information ............................ 16

3 Result ......................................................... 17
   3.1 Encoding Results ...................................... 17
   3.2 Algorithms Result ..................................... 19
      3.2.1 Scenario One, Valid Letter ....................... 20
      3.2.2 Scenario Two, Correct Delivery .................. 21

4 Discussion ................................................... 22

5 Conclusion .................................................. 23
1 Introduction

In the Logistic Business, there is a long pipeline of multiple checkpoints, inorder to deliver a package from a supplier to its receiver[4]. During this process, there is a chance of incorrect end results occurring during the many manual routines and caused by the human factor[7].

Machine Learning has been around for over 60 years, one of the first papers on the subject was Alan Turing in 1950 with his paper “Turing Test”[19], which is to determine if an AI has achieved true intelligence.

We are living in the Golden Era of Machine Learning[20]. Machine Learning is one of those technologies that is all around us and we may not even realize it[20]. It reaches from helping us translate words and sentences from over 100 languages (Google Translate), to more delicate matters such as self-driving cars and detecting possible credit frauds.

Machine Learning is a part of the Artificial Intelligence(AI) family. On a given dataset it finds underlying patterns, it then uses the knowledge from the data it was trained on, to predict the future or finding similar patterns on new data of the same data type.

Figure 1.1: A simple model of how Machine Learning works.
1.1 DATA DUCTUS

Data Ductus[1] is a Swedish IT-consultant company which offers IT product services within system development & integration, network services management solutions & orchestration, IoT-expertise & operations and support.

The company was founded in the late 80's and is now a multinational consultancy company which specializes in advanced engineering solutions tailored to their customers specific needs, which reach from small start-ups to large scale corporations. Since the late 80s has the company steadily expanded and established offices in both Sweden and the US.

1.2 ARTIFICIAL INTELLIGENCE

The idea of Artificial Intelligence is nothing brand new, it has been around since the early 1300[16]. AI is a way for devices to act intelligently, and can often be divided into two fundamental groups, applied or general. Applied[13] AIs is more common, it is used to intelligently trade stocks and share. Whilst Generalized[13] AIs are considered to be able to carry out any task. A definition can be described as,

"Artificial Intelligence is the broader concept of machines being able to carry out tasks in a way that would be considered “smart”"[13].

1.2.1 Machine Learning

Machine Learning[11] is not to be confused with traditional programing. Traditional programing differs from machine learning a bit. In traditional programing a whole lot of if-statements, for- and while-loops are required to get the desired result. Resulting in a situation that requires an if-statement for every possible outcome of any case.

Machine Learning is a subject within computer science, and it is a method that explores the study and construction of algorithms which it can learn from to make predictions on data. With the help of machine learning a computer can find patterns in the data it is provided without explicit being programmed[14]. By ”constantly” being exposed to new data the AI can iteratively interact with an algorithm learn from it and independently alter its previous ”beliefs” In short terms a definition of machine learning can be,

*It is an AI that learns how to predict patterns in a provided dataset, these predictions can later be applied and tested on real-world cases.*
1.2.2 Machine Learning Process and Terminology

The Machine Learning workflow,

"An orchestrated and repeatable pattern which systematically transforms and process information to create prediction solutions.”[9].

It is not very hard to understand what Machine Learning does. However, the Machine Learning process is a lot more complicated to understand and manage to finalize it to behave correctly.

The different learning/training methods of Machine Learning can be divided into three categories, Supervised, Semi-supervised and Unsupervised. With Supervised learning the AI gets a response for every step it takes to complete a task e.g, When playing chess the AI would get a response for every move it does with a unique chess piece. Semi-Supervised learning will only give partial response when the AI is performing a task e.g, The AI would only get a response once it has won/lost a chess match. In Unsupervised learning it gets no response at all, it has to learn what completely by itself.

Figure 1.2: The Machine Learning Process.

Figure 1.3: Shows the different Training cateogize.
The first step is to choose the raw data that is desired to create a prediction solution on. Once the desired data is acquired, it is time to start the Machine Learning process. The raw data which is provided for the AI can be divided into two categories, a Training dataset and a Test dataset. During the training process, it is only, the Training dataset which will be accessed by the model. The Training dataset is used to train our model to recognize patterns in the provided data. The Test dataset is to act as real data to test the model’s accuracy.

![Figure 1.4: Shows an example split of the raw dataset.](image)

The more tricky part with the Machine Learning process, is the preprocessing stage. Since the chosen raw data will almost never come in the format needed. Which means the chosen data needs to go through a preprocess before it can be applied. Some Machine Learning algorithms, e.g Logistic Regression\(^6\) are unable to handle words as an input, all words which exists within the data frame must then be converted into a numerical representation of that specific word. Converting words to numbers can be done using e.g Dummy Coding\(^8\) or One-Hot Encoding\(^18\). The difference between the two encoding types is that Dummy Coding only assigns a numerical representation of a word. While with One-Hot Encoding representation of a word becomes a vector of n-columns.

![Figure 1.5: Left picture Shows One-Hot Encoding and right picture Shows Dummy Coding.](image)

Once a word if it was deemed necessary to have a numerical representation. It is necessary what features in the data has the biggest impact. The data being preprocessed can contain NaN values, or it might even contain data that has negligible impact on the training of the AI and the desired result. By dropping or replacing values which deems unnecessary for the result. However it is important to keep in mind which data was removed and altered, since a justification might be needed for why it was removed.

A simple example of how a machine learning process can be executed. The data frame is kept small, and no missing values exists in it.
Table 1.1: Step 1. The raw data collected for the machine learning algorithm to learn from.

<table>
<thead>
<tr>
<th>Age</th>
<th>Gender</th>
<th>True/False</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Male</td>
<td>True</td>
</tr>
<tr>
<td>23</td>
<td>Female</td>
<td>False</td>
</tr>
<tr>
<td>8</td>
<td>Female</td>
<td>False</td>
</tr>
<tr>
<td>40</td>
<td>Male</td>
<td>True</td>
</tr>
</tbody>
</table>

Table 1.2: Step 2. The raw data after being preprocessed.

<table>
<thead>
<tr>
<th>Age</th>
<th>Gender</th>
<th>True/False</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Male/Female got the numerical representation of 0/1 and True/False are represented in a numerical way by 1/0.

Step 3. Create AI model, with desired machine learning algorithm.

Figure 1.6: The Decision Tree created by the Machine Learning Algorithm.

Step 4. Expose the model to new data. Now if the AI were to be asked if a Male, 35 years old would be afraid of dogs the AI would response with yes. Since the data it has been trained on, tells it all males above 31 years are afraid of dogs.
There is a large pool of Machine Learning Algorithms, and all of them have their pros, cons and usable areas. There is no number one algorithm for every possible scenario they all have their uses in different areas.

Figure 1.7: 60+ Machine Learning Algorithms[5].

There are a few ways to find a plausible Machine Learning Algorithm, one of the better techniques is by asking the right question. Which can be defined as,

"A solution statement to finding the end goal, starting point, and how to achieve the end goal"[9].

The purpose of asking the right question, is not to advice the number one most ultimate algorithm. It is to narrow down the possible algorithms to be applied on the problem at hand. Once a proper question has been formulated it is time to do something called a "spot-check". A "spot-check" is finding n-algorithms to be applied and trained on the model. Once a arbitrary amount of algorithms has been chosen, it is time to go through the learning algorithms to find the model that fits the problem at hand the best. Once the model that shows the most promising success rate has been discovered, the final step is to deploy the AI.
1.3 LIMITATIONS

The two main limitations of the project was the given time period and the knowledge base surrounding Machine Learning. A project of this magnitude was from the beginning a given fact that it would not be possible to finish it to a 100% with a ten week time limit.

The knowledge about Machine Learning was limited at the beginning whilst its concept is not hard to understand or to craft a very simple AI considering there are so many libraries out there that are available for usage. However to do it right and understand how the technicalities work behind the curtains is a big leap concerning knowledge.

1.4 BACKGROUND

When working with technology which has existed for a long time, there is a lot of background information that could be considered necessary to read up on and learn from. The following section will cover some of the basic and more outstanding features used to conduct this research.

1.4.1 Python

Python is one of the leading programming languages out there today[10] for conducting Machine Learning and it is the main language of this project. The language itself is an interpreted, object-oriented, high-level programming language with dynamic semantics. The syntax is both easy to write and to understand. Since Python combines both dynamic typing and dynamic binding, it makes the language very attractive for quick application development, as well as a scripting language to connect existing components together and the language has no compilation step. It supports custom written modules and packages, which encourage to modular programing and re-usage of already written code. Python is a freely distributed interpreter, for all major platforms. Python 3.x is the present and future of the language, and was released in 2008.
1.4.2 Scikit Learn

Scikit-learn[15](sklearn) is available for Python legacy version 2.6 or greater, or Python 3.3 or less. It is a module for python which provides a wide range of machine learning algorithms for supervised problems as well as unsupervised problems. Scikit-learn is distributed under BSD license, which encourage its usage in both academic and commercial settings. Scikit-learn is built on top of three other modules SciPy, NumPy and matplotlib. SciPy is a open-source software used for mathematics, science, and engineering. NumPy is a fundamental package for scientific computing with the help of python, and matplotlib is a 2D plotting library in Python.

1.4.3 Pandas

Pandas[2] is a powerful Python data analysis toolkit, which provides fast, flexible and expressive data structure design. One of its main focus is to make working with data both easy and intuitive. Since pandas has support for Series and DataFrames, it makes it possible to handle a wide majority of typical use cases. Pandas is build on top of NumPy, which is intended to work well with other 3rd party libraries. A few uses pandas offers are such as, Easy handling of missing data, insertion/deletion of columns/rows and many more.

1.4.4 Assignment Background

There are several manual steps involved in which a package has to go through from the supplier to its receiver, within the logistics business. The package has to go through several steps which consist of manual routine and a human factor. With these factors playing a critical role, there is a chance for error. Even with the skilled administrators whose task is to correct this problem by manually tracking the event of a package and eventually route it to the correct receiver. This requires a high degree of expertise, which makes the process fragile.

Figure 1.8: Shows the Post Office Pipeline [4].
1.5 GOALS AND PURPOSE

The goal with this project is to simulate a flow of packages where errors are introduced. Then detect errors with the help of Machine Learning, and finally suggest a correction.

The entire project is divided into three parts,

- Decision algorithms and function around them and correction of incorrect info.
- Learning material and how to automate the learning of decision algorithms and the correction.

This report's focus will be on, Decision algorithms and function around them and correction of incorrect info. For more information on the other two parts of the project, thesis papers, *Machine Learning In Logistics: Increasing the performance of machine learning algorithms* by Rasmus Lind covers the, *Learning - Manual Semi Manual, Auto* and *Data for Machine Learning* by Erik Hedman covers the, *Learning material and how to automate the learning of decision algorithms and the correction*, are recommended to read as well.

The purpose of this project is to show it is possible to use Machine Learning as a solution to the errors caused in the post office sorting department by e.g the human factor.

1.6 METHOD

Every morning started with a group meeting where each member of the group filled in the others on what he/she did the day before and what he/she is planning on doing that same day. A couple of times there were a bigger meeting planned, concerning larger parts of the project.
This particular subpart of the project, has had three different sessions. The most important and largest part is the Research session. The research session has existed from week one of the project until the very end. Once a decent amount of research had been done it was time to move on to the Implementation part of various Algorithms and preprocessing data models. Then when the implementation of the desired functionality was considered completed it was time to test the implementation. If the implementation did not show the desired result, a rinse and repeat loop was initiated.

1.7 SOCIAL, ETHICAL AND ENVIRONMENTAL CONSIDERATIONS

All user data presented in this project is completely fictional, whilst the zip codes, addresses, and cities might be real the people are however not. And the "letters" which has been sent are non-existing.

The intent of this research is to help and improve the efficiency of the post office sorting center. However, depending on how the research is deployed in the real world it will bring different outcomes. If the AI were to be deployed more as an alert system, which would give out a buzzing sound every time a person were to misplace a package or letter. This would mean the employees in the post office sorting center would keep their jobs and with help of the product, it would show an improvement in their accuracy rate. However, if the research were to be deployed as a fully automatized sorting system replacing all the humanoid employees it, would mean a lot of people would become unemployed.

From an environmental perspective, this research could prove to improve the carbon dioxide pollution. If fewer letters were to be misplaced that could lead to less carbon dioxide emissions.
1.8 ABBREVIATIONS AND TERMS

- Artificial Intelligence - AI, Mimic human intelligence with a computer.
- Machine Learning - ML, Learns how to make predictions on data.
- Python - Is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Pandas - A data frame library, which handles Series & Data Frames.
- Scikit Learn(sklearn) - Simple tool for data mining and data analysis.
- NumPy - Is the fundamental package for scientific computing with Python.
- Training Data & Test Data - The two different categorize the raw data can be split into.
- Confusion Matrix - A matrix showing the predicted and actual classifications.
- Classification Report - Builds a text report showing the classification metrics.
- Data Frame - Is a table of data, not to be confused with a database.
2 Design and Implementation

This chapter will provide enough information to be able to replicate the progress made in the project. The IDE for the project is Python and the package used is called Anaconda[3]. A batch based and supervised learning system was used to train and test the AI accuracy. Supervised learning was chosen because a binary result was sufficient for the two problems.

Two problems were solved with this implementations, problem one recognizes letters with too little information on them to be sent and problem 2 recognizes if a letter arrived at its correct location.

2.1 PANDAS

Pandas[2] plays a large part in this project since it handles all the data frame functionality. All the data frames and rosters were saved with a .csv file format. To use Pandas a simple import is needed,

```python
import pandas
```

Figure 2.1: Shows an example data frame, with only one row.

<table>
<thead>
<tr>
<th>Name</th>
<th>Surname</th>
<th>Street</th>
<th>StreetNr</th>
<th>ZipCode</th>
<th>City</th>
<th>Legitimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corinne</td>
<td>ucker</td>
<td>Aberga</td>
<td>175</td>
<td>30146</td>
<td>Halmstad</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

2.2 DATA PREPROCESSING

This section will explain how the preprocessing stage of data work. This includes,

- How to create a new roster if no one is found, or update an already existing roster.
- How to replace a feature in a data frame with a numerical representation.

2.2.1 Roster Update Process

The create/update roster process is the first out of the two steps in the process of preprocessing data in a data frame.

This function takes three different parameters,

- dataFrame - This is the data frame desired to copy new information from.
- columnName - This is the column name which, data is to be copied from.
- rosterName - The name of the roster to be Updated/Created.

First, a check is made, if the roster with the file name exists or not. If the file does or does not exist, two separate scenarios will happen.

If the roster does not exists, a new .csv file will be created in the working directory. This file will contain a temporary column, e.g tempColumn which will be removed later. Once the .csv file has been created, it will be loaded with Pandas as a temporary roster data frame. Another check will be made if, the first column has the name of tempColumn the program will know this is a newly created file. With help of Pandas, a simple function call is all that is needed to merge the rosters data frame with the desired column in the data frame we want to copy from, this is done with the concat function. Once the two data frames have been merged, the temporary column in the roster data frame can be dropped. Leaving it with the only necessary information for the roster.

If the file exists, no new file will be created and the program jumps directly to the part where the rosters data frame is being loaded. The desired data frame which new information is being added from is copied into a temporary data frame variable. Which drops any duplicates that may appear in the frame, to only contain unique features. Any missing values are dropped as well since this is unnecessary information to save in the roster. The temporary data frame variable is then converted to a list, which is only containing unique features. This list is then looped through and inclemently appended to the temporary data frame, with Pandas append function.

After any of the scenarios are fulfilled, there is a chance that a duplicate or a null value is existing within the roster. Pandas provide two useful functions, one which drops all duplicates, this function takes care of any duplicates which may exist within the roster as well as removing values which may be missing or are null values. The final step is to save the file in a .csv format. Pandas have a function to do this as well.

2.2.2 Encoding

Four different methods of encoding were tested. This section will be presenting the two final solutions which showed the best performance, and most promising results.

2.2.2.1 Encoding with Roster data

Encoding with Roster data utilizes the implementation from section 2.2.1. This is a more valid solution to this specific project since, e.g Ham and Spam needs to have
the same numerical representation even if they were to appear in two different data frames.

The encoding function takes four parameters:

- **dataFrame** - Desired data frame to encode.
- **columnNameInRoster** - This is the name of the loaded Roster.
- **columnNameInDataFrame** - This is the name of the column in the data frame.
- **rosterFile** - Name of the roster to be loaded.

Since the name in the roster may not be the same as in the data frame, two separate parameters are used in this solution. The first step in the function is the loading of the roster file and storing it in a temporary variable. Once the roster has been loaded, a list is made of all the features which appear in the roster. This list is also saved in a temporary variable. A temporary empty dictionary is needed to be assigned before next step, this dictionary is used to map a feature with its numerical representation. Once a dictionary is created it is time to loop through the list of features created from the roster. A check is made inside the loop, which has the purpose of making sure only features which appear in the data frame to be encoded with its numerical representation are added to the dictionary. Once the loop has finished, each feature in the data frame is swapped out for its numerical representation. Since supervised learning was used in this project, the final column 'Correct' which contains True and False, needs to be encoded to a numerical representation as well. True was assigned with '1' and False with '0'. The final step is to change all missing values in the data frame, for this project '-1' was used.

### 2.2.2.2 Dummy Encoding with Random Data

This way of encoding does not require a roster to fetch information from, since it just assigns a feature on the spot with a numerical representation, however this way of encoding is vague since it encodes a feature depending on its order in a data frame. Meaning if e.g ham, can in one data frame have the value of '7' and in another, the word have been assigned a numerical representation of '0'.

The first step is to import sklearn’s LabelEncoder library, which will provide necessary functions to do dummy encoding.

```python
from sklearn.preprocessing import LabelEncoder
```

Once the LabelEncoder library has been imported, replacing all null features is a necessary step. In this project it was done manually, one thing which needs to be considered here is that the LabelEncoder can not handle a column which contains
features of both string and numerical values. As same as in 2.2.2.1, the True and False column got the numerical representation of '1' and '0'. However, one thing differs here, depending on which column a NaN value appeared in, a different representation had to be made. If a NaN value were to appear in a column with string features it had to be given a string e.g 'missing' or if it appeared in a column with numerical features it was given as above '-1'. After the True and False and NaN values have been dealt with, creating a list of each feature in a column is required. Once the list is created, looping through each feature in the list and assigning a numerical representation is required and at the same time that numerical representation is assigned to the data frame. All this is done with the help of a function in the LabelEncoder library.

2.3 TRAINING SET & TEST SET

This section will describe how to create a training and test set of a data frame, to later be applied to an algorithm. Creating a training set and test set, is quite simple since sklearn has a built in function for doing this.

from sklearn.cross_validation import train_test_split

It is also necessary to create eight variables, four which will be saved and used later and four which will act as temporary variables. Since this is supervised learning we need to tell the AI which provided features are true and which are false. Two of the temporary variables should hold two different lists of n-features. The first temporary variable should hold the feature columns names, e.g Name, Street, City. The second temporary variable should hold the prediction columns name. Pandas provided with a function to create an NDarray of each feature existing in one of the previously stored temporary variables. In this case, the feature NDarray will be (6 x n), whilst the prediction NDarray will be a (1 x n). A built in function in Scikit-learn is used to divide, the test and train data set. The function then returns four NDarrays, which is desired to store and be reused later on.

2.4 ALGORITHMS

This section will explain how to implement the machine learning algorithms used in this project.

An important part when choosing an algorithm as mentioned in section 1.2.2, was to ask the right question when narrowing down the possible machine learning algorithms. However, this is just one of many ways to choose an algorithm, for this project three of the most commonly used machine learning algorithms were chosen. Sklearn was used to import already constructed algorithms.
2.4.1 Random Forest, Naive Bayes, and Logistic Regression

The implementation of Random Forest, Naive Bayes, and Logistic Regression are almost identical. To be able to use the Random Forest, Naive Bayes and Logistic Regression, the following libraries needs to be called,

```python
from sklearn.ensemble import RandomForestClassifier, from sklearn.naive_bayes import GaussianNB and from sklearn.linear_model import LogisticRegression.
```

All three algorithms are fairly easy to implement, only three function calls are required to train the AI with these algorithms. First step is to create the model, this is done by calling RandomForestClassifier(), GaussianNB() or LogisticRegression(). Once a model is created it is time to process the model with the given training data. The final step is to pass "real data" to the AI and uses its set parameter values which were set during training to make a prediction on the new data.

The Logistic Regression has a try except error handling if a to small data frame were too be passed.

2.4.2 Printing Information

One common attribute all algorithm share is the printing of information, on how well the AI learned with the provided data. There is a library from sklearn needed, the library provides score functions, performance and pairwise metrics, and distance computations. There are four main statistics readings of most relevance for this project, is the Training accuracy, Test accuracy, the produced confusion matrix and a classification report. Scikit-learns metric provides with functions to print, the Metric Score Accuracy, a Confusion Matrix and a Classification Report.
3 Result

The question which was asked before this project started was, which algorithm(s) would be able to provide with sufficient correct behavior to detect incorrect information and provide with a correction solution of the incorrect information. To be able to answer this question, a whole lot of research had to be conducted. The main focus of the research phase went into, learning how machine learning works. Several iterations were made and several sources were looked into. However the two main sources were Jerry Kurata[9] and Artificial Intelligence: A Modern Approach written by Russell, S. and Norvig, P[17].

Three different IDEs were looked into, R, Java, and Python. However, in the end, Python was chosen because it provides with a lot of already existing libraries within machine learning e.g, scikit-learn and pandas and it was the language I was most familiar with out of the three.

The result in this section present the test result of two scenarios that had to be solved,

- Is the letter valid?
- Did the letter arrive at the correct end destination?

It also provides the final results of encoding features in a data frame.

3.1 ENCODING RESULTS

Figure 3.1 and 3.2, shows the end result of two different encoded data frames using a register to load a numerical representation of a feature. In figure 3.1, row five, column City and in figure 3.2 row four, column City. Both the data frames have the numerical representation ”18” for ”Vaxjo”. The reason for why this is an important implementation, is because if the program were to start to give suggestions on where to send a letter with insufficient information on it. The AI would not be able to learn in a correct way, if a word were to have multiple numerical representations.
Figure 3.1: First Data Frame. Top picture shows the data frame before being encoded, whilst the bottom one shows the data frame encoded.

<table>
<thead>
<tr>
<th>Name</th>
<th>Surname</th>
<th>Street</th>
<th>StreetNr</th>
<th>ZipCode</th>
<th>City</th>
<th>Legitimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Debert</td>
<td>Alegatan</td>
<td>155</td>
<td>55125</td>
<td>Jonköping</td>
<td>TRUE</td>
</tr>
<tr>
<td>1</td>
<td>Shu</td>
<td>Daggbag</td>
<td>6</td>
<td>25125</td>
<td>Helsingborg</td>
<td>TRUE</td>
</tr>
<tr>
<td>2</td>
<td>Perry</td>
<td>Hanntgen</td>
<td>118</td>
<td>97114</td>
<td>Luleå</td>
<td>TRUE</td>
</tr>
<tr>
<td>3</td>
<td>Nilsa</td>
<td>Tensvik</td>
<td>43</td>
<td>60133</td>
<td>Närköping</td>
<td>TRUE</td>
</tr>
<tr>
<td>4</td>
<td>Val</td>
<td>Borje</td>
<td>97</td>
<td>55167</td>
<td>Jonköping</td>
<td>TRUE</td>
</tr>
<tr>
<td>5</td>
<td>NaN</td>
<td>Jacoby</td>
<td>42</td>
<td>35150</td>
<td>Vasa</td>
<td>FALSE</td>
</tr>
<tr>
<td>6</td>
<td>Dann</td>
<td>Bouldin</td>
<td>118</td>
<td>50125</td>
<td>NaN</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Table 3.1: Time to encode a data frame.

Figure 3.2: Second data frame.

The final iteration of the encoding process time in seconds grows linearly, the table below shows the time it takes to encode a data frame of arbitrary size.

<table>
<thead>
<tr>
<th>Data Frame Size</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.25</td>
</tr>
<tr>
<td>10 000</td>
<td>2.00</td>
</tr>
<tr>
<td>100 000</td>
<td>15.56</td>
</tr>
<tr>
<td>1 000 000</td>
<td>158.39</td>
</tr>
</tbody>
</table>

Table 3.1: Time to encode a data frame.
3.2 ALGORITHMS RESULT

The chosen algorithms provided with the desired end result, which was binary. All algorithms test results for scenario one were done with one batch each of 900,000 generated letters. For scenario two, 536,409 generated letters were used, to test the performance of each algorithm. Each letter had a 50% chance of causing an error in the generation on both the scenarios and a 10/90 split was used for the training data and test data.

The rows in a Confusion Matrix is the predictions made by the AI, whilst the columns are the actual true/false in the data frame.

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>True-Positive</td>
<td>False-Negative</td>
</tr>
<tr>
<td>Negative</td>
<td>False-Positive</td>
<td>True-Negative</td>
</tr>
</tbody>
</table>

Table 3.2: A Confusion Matrix for a binary classification model.

A useful tool to visualize how well the AI performed on the test data, is by using a Confusion Matrix[12]. This tool is most useful when there is only a binary outcome, of a prediction.

A confusion matrix tells the number of correct and incorrect predictions made by the AI, compared to the actual outcomes of the data. In Table 3.2, the rows are the presentation of the predictions made by the AI, whilst the columns are the actual positive (true) and negative (false) in the data frame. The three most important out of the four to know about are, True-Positive, False-Positive and True-Negative.

- **True-Positive (TP)** - Features which were predicted positively by the model and were positive in the data.
- **False-Negative (FN)** - Number of features which were negative, however, predicted positively by the model.
- **False-Positive (FP)** - Predicted to be negative by the model, where in fact positive in our data.
- **True-Negative (TN)** - Predicted negative feature by the model and were negative in the data.

The results presented for each algorithm are made on them in their raw form. No modification or improvements were made.
3.2.1 Scenario One, Valid Letter

The first scenario was to test the AI’s capability to detect if a letter contained sufficient information, in order to be delivered. The following figures show the Confusion matrix of the corresponding machine learning algorithms, Random Forest (Table 3.2), Naive Bayes (Table 3.3) and Logistic Regression (Table 3.4). The False-Positive in this scenario means that the AI predicted a letter with sufficient information to not have sufficient information.

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>536420</td>
<td>0</td>
</tr>
<tr>
<td>Negative</td>
<td>0</td>
<td>363580</td>
</tr>
</tbody>
</table>

Table 3.3: Random Forest

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>466201</td>
<td>70219</td>
</tr>
<tr>
<td>Negative</td>
<td>82126</td>
<td>281454</td>
</tr>
</tbody>
</table>

Table 3.4: Naive Bayes

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>468918</td>
<td>67502</td>
</tr>
<tr>
<td>Negative</td>
<td>156452</td>
<td>207128</td>
</tr>
</tbody>
</table>

Table 3.5: Logistic Regression

Random Forest performed flawlessly in finding out which letters are valid or not. It predicted all valid letters to be valid and all non-valid letters to be non-valid.

Naive Bayes and Logistic Regression gave a similar performance, with a few differences. Naive Bayes were better to distinguish which letters were not eligible for delivery (281 454 to 207 128), whilst Logistic Regression performed better in finding out which letters were eligible for delivery (468 918 to 466 201). Naive Bayes had a lower number in False-Positive than Logistic Regression(82 126 to 156 452).
3.2.2 Scenario Two, Correct Delivery

The second scenario was to test the AI’s capability to detect which letters arrived at their final destination. Table 3.6-3.8 is each machine learning algorithms confusion matrix. The False-Positive in this scenario, mean that the AI predicted a letter which did arrive at the correct address did not.

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>351183</td>
<td>2792</td>
</tr>
<tr>
<td>Negative</td>
<td>3634</td>
<td>178800</td>
</tr>
</tbody>
</table>

Table 3.6: Random Forest

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>338897</td>
<td>15078</td>
</tr>
<tr>
<td>Negative</td>
<td>52823</td>
<td>129611</td>
</tr>
</tbody>
</table>

Table 3.7: Naive Bayes

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>341747</td>
<td>12228</td>
</tr>
<tr>
<td>Negative</td>
<td>105051</td>
<td>77383</td>
</tr>
</tbody>
</table>

Table 3.8: Logistic Regression

Random Forest gave a decrease in performance compared to scenario one. However, the algorithm still gave the best performance out of the three.

Naive Bayes and Logistic Regression gave a similar performance in scenario two as they did in scenario one. Naive Bayes as in the first scenario were better to learn which letters did not arrive at the correct end address(129 611 to 77 383), whilst Logistic Regression was better to learn which letters had arrived at the correct address(341 747 to 338 897). As in the first scenario, Naive Bayes had a lower number of letters in False-Positive than Logistic Regression(82 126 to 156 452). 

21
4 Discussion

A vast majority of this project went into the research phase. A basic knowledge about AI existed from the start and machine learning and its workflow were completely unexplored terrain. However, there was negligible information about what machine learning is and the logical/technical thinking concerning the area was supposed to be approached.

Performance is something that can always be improved, and this project is no exception. As shown in Table 3.1 the time complexity $O(n)$ of encoding a data frame increases as the roster increases. There are as few possible functions within Pandas that could be of assistance to improve the encoding time, these were acknowledged however never used since the time did not exist. When updating a roster a few unnecessary steps are executed. All features which exist within the data frame, even if they exist in the roster they are added and later dropped from the roster. This could easily be avoided by doing a single check to see if there are any anomaly that exists in the data frame which is not present in the roster and just add the anomaly.

The algorithms are the main body of this project and need to be given more thought. Random Forest, Naive Bayes, and Logistic Regression were chosen for this project because they provided a variate of different attributes to be tested, and with scikit learn they were easy to implement.

Random Forest did provide with the desired result, it performed flawlessly on scenario one and with an inferior decrease in performance on scenario two. Out of the three, algorithms are provided with the best result. However it does this with a cost, Random Forest uses ensemble learning which means it is expensive to use. In a small data frame, it is okay, though once an arbitrarily large data frame is used the time complexity of the algorithm will increase.

Naive Bayes, will not be able to work as a final solution for the end goal since it provided a mediocre result. Naive Bayes assumes each features share no relationships and will not be able to connect previous learned data with newly presented.

Logistic Regression did provide a desired result, and would also be a valid solution to the end goal with a little modification. The algorithm in its basic form can only provide with a binary result.
5 Conclusion

A lot of hours went into this project, and a lot of new information was acknowledged. A working system exists, the AI is able to learn new information in a supervised fashion in the form of batches.

The encoding process is fully operational and provides with the functionality intended. However, it could use some modification, even though the current implementation is decently fast it can be improved to provide better performance.

The process of choosing and implementing the native form of the algorithms was quick and simple. Not much consideration was given to the algorithm’s time complexity and structural complexity. This project had one portion which was not completed, the correction of incorrect information. The reason to this is because the time limit was not sufficient and this was known from the start that the time to complete the entire task was not a possible outcome. A theory of using neural network or deep learning was considered to move the project forward. This would have been used to allow the AI to be able to give suggestions on letters which had insufficient information on them.

Overall this project was more successful than expected.
Bibliography


[8] FAQ WHAT IS DUMMY CODING?


[18] H.H Strand. “What is one hot encoding and when is it used in data science?”
In: (2016)

Available: https://drive.google.com/open?id=0Bx1LjriH2jVBTnVvVnFjSWtoSzQ (2017-05-04).