**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | Programming for DA,  Statistics for Data Analytics,  Machine Learning for Data Analysis,  Data Preparation & Visualisation |
| **Assessment Title:** | CA1 |
| **Lecturer Name:** | Marina Iantorno  Muhammad Iqbal  David McQuaid |
| **Student Full Name:** | Rita Raher |
| **Student Number:** | sbs22115 |
| **Assessment Due Date:** | 11th April 2022 |
| **Date of Submission:** | 11th April 2022 |

Icon

Description automatically generated with medium confidence

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Dublin City Centre Footfall prediction using supervised machine learning algorithms:

Performance analysis and comparison

Table of Contents

[Abstract 3](#_Toc100602808)

[1. Introduction 3](#_Toc100602809)

[2. Methodology 3](#_Toc100602810)

[3. Data collection 4](#_Toc100602811)

[4. Data Understanding 6](#_Toc100602812)

[Poisson Distribution 11](#_Toc100602813)

[Normal Distribution Analysis 12](#_Toc100602814)

[5. Data preparation 14](#_Toc100602815)

[I. Handling missing values 14](#_Toc100602816)

[II. Handle duplicates 15](#_Toc100602817)

[III. Remove outliers 15](#_Toc100602818)

[IV. Transform Categorical values 16](#_Toc100602819)

[V. Apply feature scaling 16](#_Toc100602820)

[6. Modelling 16](#_Toc100602821)

[Supervised 17](#_Toc100602822)

[Unsupervised 19](#_Toc100602823)

[7. Evaluation 20](#_Toc100602824)

[Classification reports 20](#_Toc100602825)

[Confusion matrix 21](#_Toc100602826)

[Feature importance 21](#_Toc100602827)

[8. Conclusion 23](#_Toc100602828)

[Bibliography 24](#_Toc100602829)

## Abstract

This project aims to identify machine learning classifiers with the highest accuracy for predicting the street name of the footfall dataset. Four supervised and one unsupervised algorithm were compared to find the best model. Features were ranked based on their importance for each model. Later the results were cross-validated with GridSearchCV to find the optimal hyperparameter for each model. Based on footfall data collected from Dublin City Council, Random Forest and Decision tree delivered the best results. Random forest achieved a 93% accuracy, and Decision tree achieved a 92% accuracy.

## Introduction

In an ever-increasing digital world, it is interesting to observe the place of the high street.

More and more shops are moving away from expensive brick-and-mortar stores to online stores. The digital transformation of retailers’ businesses, for example, music shops being replaced by online stores or apps, is another reason retailers exit the high street (Roderick Duncan, 2014). Footfall density is one of the critical metrics for measuring the success of a retailer’s location; others include attraction rate, conversion rate, and average spend (Graham, 2016). Dublin City Council (DCC) have collected footfall data from the four leading high streets in Dublin city: Capel Street, Henry Street, Mary Street and O’Connell Street. This historical data is from 2015 and is available on [Data.gov.ie](https://data.gov.ie/dataset/pedestrian_footfall?package_type=dataset). The dataset will be merged with weather data from 2015 to determine whether precipitation or temperature impacted consumer behaviour. We will also examine the impact of bank holidays and weekends and whether these factors gave the city retailers the footfall boost they needed to survive.

Using this dataset, we will look at various machine learning classification techniques to help identify the street name based on the data. We will analyse the performance, accuracy and speed to help identify the best model for the task.

## Methodology

The Cross Industry Standard Process for Data Mining, as known as CRISP-DM, the methodology was developed with DiamlerChrysler, SPSS and NCR in 1996 (Santos, 2008). CRISP-DM, KDD, and SEMMA have all been compared, and CRISP-DM was found to be the most robust process for data mining projects (Qaiser, 2014). It consists of a cycle of 6 steps, as seen in Figure 1 (Chapman, 2000).

Diagram

Description automatically generated

Figure : Phases of the CRISP-DM reference model

1. Business understanding: Assess the data project from a business perspective and formulate a plan to achieve those goals.
2. Data understanding: Collect and understand the data to help gain insight and formulate the hypothesis.
3. Data preparation: begin activities for preparing the data for modelling, including pre-processing, transformation and cleaning.
4. Modelling: apply various modelling techniques to the data
5. Evaluation: evaluate and review the modelling techniques applied during the modelling step, and analyse whether the business goals were achieved.
6. Deployment: Create a real-world webpage for the model

## Data collection

For this study, data from footfall was collected from [Data.gov.ie](https://data.gov.ie/dataset/pedestrian_footfall?package_type=dataset). The data contained data from 2015 with four streets: Capel Street, Henry Street, Mary Street and O’Connell Street. There were four datasets for each street; the shape and info for each dataset were investigated to help identify if data for each day of the year was in the dataset. The total count and data types were observed in each dataset (Jupyter: Footfall\_1 [7]). As the street name was set as a column name, the melt function reshapes the data with the street as a column name and street name as a value for each dataset (Jupyter: Footfall\_1 [9]).

Each street dataset was combined into one dataset using a for loop, and the append function of the new dataset is called df\_total (Jupyter: Footfall\_1 [47]).

Significant data is missing from O’Connell street, about 25.7% is missing for IN, OUT and value, and only two values are missing from Henry street for the same columns, missing 0.5% of the data.

Taking that date column and applying datetime functions, additional columns were engineered for: day, month and week (Jupyter: Footfall\_1 [53]). As a result, we can determine which days, months, and weeks have the highest or lowest footfall.

Considering that people tend to stay indoors in bad weather, weather data was collected using a python library, Meteostat (Meteostat, n.d.). Meteostat is one of the most prominent open weather and climate data vendors. It has access to thousands of weather stations worldwide, and the closest station was Dublin. Adding the location point using: longitude and latitude (53.3331, - 6.2489) for Dublin and adding the start and end date for 2015 (Jupyter: Weather\_data [4]), climate data for 365 days in 2015 was obtained.

Bank holiday data from 2015 was collected from [bankholidaysireland.irish](https://bankholidaysireland.irish/year/2015), and a table was created with the bank holidays. Bank holiday data were merged to the primary dataset using the date column to do a left join on Date (Jupyter: Footfall\_1 [57]). The values are converted in the bank holiday column to Booleans: True if it is a Bank holiday and False if not. This new column will help analyse whether bank holidays affect consumer behaviour.

Because Christmas is such a busy time for retailers, an extra feature was engineered to countdown the days to Christmas and added to the dataset. A column for Christmas with “2015-12-25” as the value was created, and then created a function was defined to subtract the date column for the Christmas date. This function was applied to the dataframe using the pandas lambda function (Jupyter: Footfall\_1 [59]). All the columns in the dataset can be seen in Figure 2.

|  |  |
| --- | --- |
| Parameter | Description |
| Date | Date |
| IN | Footfall count into the street |
| OUT | Footfall count into the street |
| street | Name of street where footfall was collected |
| value | Total Footfall (IN and OUT) |
| day | Day of the week |
| month | Month |
| week | Week number |
| tavg | The average air temperature in °C |
| tmin | The minimum air temperature in °C |
| tmax | The maximum air temperature in °C |
| prcp | The daily precipitation total in mm |
| bank\_holiday | Whether it is a bank holiday |
| days\_countdown | Days until Christmas |

Figure : Columns in the dataset

## Data Understanding

Descriptive statistics is an effective way of summarising data and identifying patterns.

They are measures that show where the centre of the data line is and are called measures of central tendency, which is the measure of the centre (Weiss, 2017). Central tendency includes: mean, median and mode.

* The mean is the sum of observations and dividing by the total
* The median finds the middle of the data
* The mode finds the most frequently reoccurring value

Figure 3 shows that the mean values for IN, OUT and value are considerably greater than the median, which means that they have been affected by larger or extreme values in the dataset, also known as outliers (Jupyter: Footfall\_1 [68-70]).

|  |  |  |  |
| --- | --- | --- | --- |
| observation | Mean | Median | Mode |
| IN | 8510.521233 | 4889.5 | 7084.0 |
| OUT | 8383.567123 | 3523.0 | 7910.0 |
| value | 16901.051370 | 7878.5 | 15097.0 |
| Street |  |  | Capel\_Street, Henry\_Street, Mary\_Street, O\_Connell\_St |
| day | - | - | Thursday |
| month | - | - | January, March, May, July… |
| week | 27 | 27.0 | 2, 3, 4, 5, 6…. |
| Tavg | 9.551233 | 9.9 | 9.1 |
| tmin | 6.119452 | 6.4 | 1.0 |
| tmax | 13.929863 | 14.3 | 14.8 |
| prcp | 2.337808 | 0.3 | 0.0 |

Figure :Table of Mean, Median and Mode

Other descriptive statistics that look at the measure of variability in the dataset are standard deviation, and the range can be seen in the below table. Standard deviation tells us how spread the data is from the mean. A low standard deviation means that all the data points are gathered close to the mean. (Jupyter: Footfall\_1 [66]). In Figure 4, we can see that IN and value have big outliers as the mean is quite small in comparison and their standard deviation is also big, meaning they have a wide spread of data. tavg, tmin, tmax and prcp have data points close to the mean as their standard deviation is between 4 and 4.6.

Table

Description automatically generated

Figure : Table of Descriptive Statistics of quantitative variables

After analysing the bank holiday dates, there is a drastic difference in footfall on the bank holiday dates. It appears that bank holidays have a negative effect on retailers in Dublin see Figure 5 (Jupyter: Footfall\_1 [73]).

|  |
| --- |
| Figure : Box plot of IN with and without bank holiday |

Examining weekdays vs weekends, there is no drastic difference. Sunday has very low footfall, which lowers the mean for weekends see Figure 6 (Jupyter: Footfall\_1 [76]).

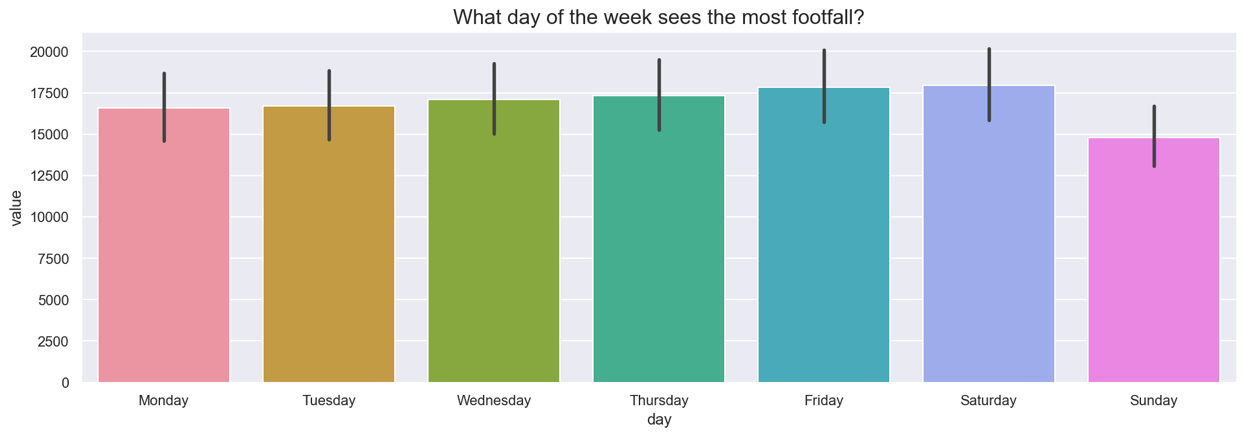


Figure : Footfall In vs day of the week

An elegant way to visualise variability is with a box plot. The box plot represents the variability or spread in the data is shows the minimum and maximum, the upper and lower quartiles and the median (Potter, 2006). Figure 7 plots street and OUT. Henry Street has the greatest mean and seems to be affected the most by extreme values (Jupyter: Footfall\_1 [80]).

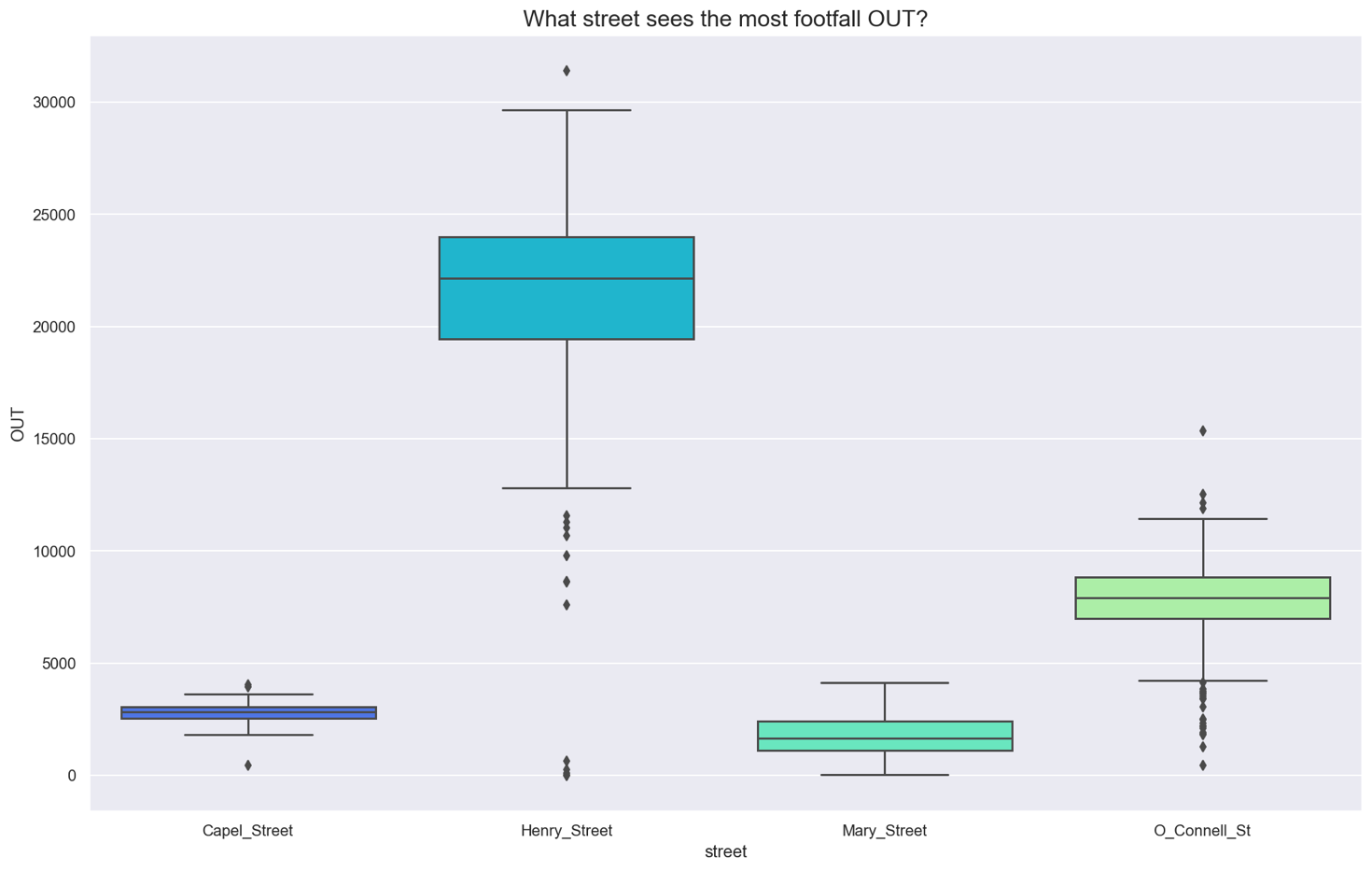


Figure : Boxplots of streets and OUT

Table

Description automatically generated

Figure : Pairplot of variables

The shape of distributions plays a vital role in selecting the correct statistical analysis (Weiss, 2017). The pair plot in Figure 8 has produced eight distributions. Value, OUT, IN could be considered multimodal as they have multiple peaks; however, they are not equal in height. prcp has a right-skewed distribution, and the rest of the variables look normally distributed (Jupyter: Footfall\_1 [81]). There is also some interesting clustering happening in the plots that will hopefully make it easy for the machine learning algorithms to create their models.

Insights from the data.

* Fridays and Saturdays see the most footfall
* Henry street is the busiest street
* Bank holidays are the quiet
* Sunday is the quietest day of the week

### Poisson Distribution

Siméon-Denis Poisson introduced the Poisson probability distribution, a discrete distribution that measures the probability of a number of events occurring in a certain period of time or space (Weiss, 2017). In the dataset, a few variables that could be considered discrete, including day, month and week. These are considered to be discrete and not continuous because they are categorical, they are countable, and they are not measurements.

Poisson will be used to determine the probability of seeing footfall of IN exactly 2,900 on a Monday in Capel street (Jupyter: Footfall\_1 [91]). In the Poisson formula, λ is the mean number of successes in the interval, the mean for Footfall for IN on a Monday is 3005.25 (Jupyter: Footfall\_1 [92]).

Formula for calculating Poisson distribution (Weiss, 2017):

f(x)= P(X= x) =

**x** = 2900

**λ** (average)= 3005.25

f(x) = P(X=2900) = (e-3005.25 3005.25 2900 )/2900!

The probability distribution app was used to confirm the calculation by selecting the Poisson distribution in the dropdown and entering the above values for x and λ, see Figure 9:

f(x) = P(X=2900) = (e-3005.25 3005.25 2900 )/2900! = 0.00115

According to the calculation, the probability of seeing footfall IN on a Monday exactly equal to 2,900 is 0.00115%.

Funnel chart

Description automatically generated with low confidence

Figure : Poisson distribution x = 2900

### Normal Distribution Analysis

When continuous data looks to have a bell-shaped curve, it is said to have a normal distribution (Weiss, 2017). Normal distribution, also known as the Gaussian distribution, is described as a variable where the mean, median, and mode are identical. The spread of data depends on the standard deviation (Villeneuve, 2018). It is one of the most commonly used distributions in statistical analysis.

A few variables in the dataset can be considered continuous data, for instance, tavg, tmin, tmax and prcp. These are variables that represent measurements.

Looking at all the numeric variables with matplotlib hist (Jupyter: Footfall\_1 [95])., the variable tmax, which is continuous, seems to be close to a bell-shaped curve see Figure 11.

For this project, tmax will be treated as having a normal distribution and try to analyse the probability of the temperature being **22°C and over** in Dublin.

A new dataset was created called df\_normal and filtered to Capel\_street to ensure no duplicate values in the analysis (Jupyter: Footfall\_1 [96]).

The formula for the normal distribution:

The average tmax, is 13.93 °C and the standard deviation, is 4.56 °C.

 = 1.7697367

P(x > 22) = P(z > 1.76973684211)

Probability is 0.038385506569695166 is calculated in the Jupyter notebook (Jupyter: Footfall\_1 [99]) using scipy and is compared to the Probability distribution in the app see Figure 10. The probability of tmax being **22°C and over** in Dublin is 3.83%.

Diagram

Description automatically generated with low confidence

Figure : Normal distribution of X > 22

However, the tmax variable in a histogram does not look like a perfectly symmetrical bell curve. Examining the measures of central tendency, the mean is 6.12, the median is 6.4, and the mode is 1 (Jupyter: Footfall\_1 [103]). For the variable to be normally distributed, these parameters should be identical. Shapiro and normaltest are from the scipy library, and both test whether the data comes from a normal distribution. A function loops through each column and determines if any are normally distributed. According to these tests, the data was not normally distributed (Jupyter: Footfall\_1 [104-105]).

Chart, line chart

Description automatically generated

Figure : tmax histogram and density plot

## Data preparation

To prepare the data for machine learning algorithms, here are the steps followed:

1. Handling missing values
2. Handle duplicates
3. Remove outliers
4. Transform categorical values
5. Apply feature scaling

### Handling missing values

It is common in most data projects to deal with missing values in the dataset. Some machine learning algorithms cannot perform on datasets with missing values (Brownlee, 2020). A simple way to handle this would be to drop the affected rows and continue the analysis. Another option is to perform data imputation, which replaces null values (Lan Yang, 2020). The SimpleImputer module from sklearn transforms missing values. It has four strategies for replacing data:

* mean strategy, where the mean of each column is used to replace the missing value
* median strategy where the median is replaced
* most\_frequent where the most frequent value is used
* constant where a value is passed in to replace the missing value

As seen above, two datasets from Dublin City Council have missing values: df\_henry and df\_oconnell. df\_henry has two missing values, with 0.5% of the overall data in the dataset.

The median imputation strategy to be the best approach to impute the missing data. After examining the histograms, the data for IN is right-skewed, and the data for OUT is left-skewed (Jupyter: Footfall\_1 [19-20]). The mean is not recommended for imputation when the data is skewed. SimpleImputer is applied on the columns in, out, value using the median strategy (Jupyter: Footfall\_1 [22]). As seen in Figure 12, replacing missing values has minimal impact on the central tendency of the data.

|  |  |  |
| --- | --- | --- |
| Parameter | Before | After |
| Mean (IN) | 20335.754821 | 20337.087671 |
| Standard deviation (IN) | 5400.429977 | 5385.603212 |

Figure : Before and after simpleimputer on df\_henry

df\_oconnell has the most significant proportion of missing data, with 25.7% missing for columns IN, OUT and value. Following the same median strategy as above, there is slight skewing on the histograms (Jupyter: Footfall\_1 [38-39]). The missing values for each column were replaced (Jupyter: Footfall\_1 [43]). Figure 13 shows there is a significant difference in the standard deviation for the column IN.

|  |  |  |
| --- | --- | --- |
| Parameter | Before | After |
| Mean (IN) | 6859.188192 | 6917.084932 |
| Standard deviation (IN) | 1799.990086 | 1553.370727 |

Figure : Before and after simpleimputer on Before and after simpleimputer on df\_oconnell

Figure 14 shows the effect the simpleImputer has on the underlying data: reduced variability with the interquartile ranges and greater outliers. Imputation of the data has a positive outcome on the accuracy of the algorithms, which we will see later.

|  |  |
| --- | --- |
|  |  |

Figure : df\_oconnell before and after using SimpleImputer

### Handle duplicates

Duplicates can cause overfitting in the model and should be removed before passing data into a machine learning model. Applying pandas drop\_duplicates()removes any duplicate rows. After running, it returns 1460 rows; there were no duplicate rows (Jupyter: Footfall\_1 [110]).

### Remove outliers

An outlier is an anomaly outside of the lower and upper quartiles of the data, generally representing either high or low extremes (GRUBBS, 1969). Boxplots visually show outliers and represent them as individual markings on a box plot (Potter, 2006). Outliers present variability in the dataset and have statistical significance over the mean of the data, leading to misleading interpretations. Therefore, it is best practice in machine learning to remove outliers.

The is\_outlier function removes outliers from IN, OUT and value grouped by street, ensuring outliers are removed from the street instead of the overall dataframe (Jupyter: Footfall\_2\_models [10-11]). Figure 15 shows before outliers are removed and Figure 16 shows after outliers are removed.

|  |  |
| --- | --- |
| Figure 15: Before outliers were removed | Figure 16: After outliers were removed |

### Transform Categorical values

Machine learning models require all input and output variables to be numeric to allow for them to perform mathematical computations and statistical analysis (Casari, 2018). Categorical data is transformed into numeric values using the category encoders library (Jupyter: Footfall\_2\_models [19]). The target street label, y, is also a categorical variable. We will use LabelEncoder from sklearn to help with the encoding (Jupyter: Footfall\_2\_models [21]).

### Apply feature scaling

Machine learning algorithms will give more weight to features with larger numeric values (Roy, 2020). For algorithms to perform better, it is necessary to scale features. Standardisation is one of the techniques of feature scaling which bring all features to the same size (Guido, 2017). Features in the dataset such as IN, OUT and value have numeric values with wide ranges, while others such as prcp, tavg and tmin have relatively narrow ranges of numeric values. Standardisation transforms the data to a mean of 0 and standard deviation of 1 but keeps the underlining shape of the data (scikit-learn, 2022). StandardScaler from sklearn was used to scale X\_train and X\_test (Jupyter: Footfall\_2\_models [20]).

## Modelling

Machine learning algorithms can be divided into categories:

* Supervising learning
* Unsupervised learning
* Semi-supervised learning
* Reinforcement learning

Supervised learning algorithms make predictions by learning from labelled data, and unsupervised algorithms make predictions based on observations. Supervised learning can often be expensive and time-consuming, especially when it comes to labelling data, it requires a great deal of manual work. Although unsupervised learning algorithms eliminate manual work by labelling the data, accuracy may be low, and the learning process may take longer.

As the dataset has labelled data, multiple supervised learning algorithms can help to solve this multiclass classification problem. It will be interesting to apply clustering as an unsupervised learning algorithm. Cross-validation will help later to identify the best models for this problem.

### Supervised

Supervised learning algorithms try to model relationships between independent and dependent variables, also known as the target variable (Fumo, 2017). They are trained with labelled data to produce the desired output. To prepare the data for supervised learning the X (features), y (target) and split into train and test using the train\_test\_split from sklearn (Jupyter: Footfall\_2\_models [18]). The reason for splitting the data is to ensure that the model is not overfit against the training dataset that is why the model is tested with different data.

#### Decision tree

Decision trees are primarily used as models for classification and regression tasks (Guido, 2017) (Lior Rokach, 2014). They learn from a hierarchy of if/else questions or decision nodes, which are represented by a tree, leading to a decision (a leaf) and in this case, classifying the street name. When used for classification problems, decisions tree are called classification trees (Lior Rokach, 2014). The root node represents the entire population as a decision is made, then a split occurs, and more nodes populate the tree. The tree will split until the classification is complete. Leaf nodes represent the class labels. There are 2 criterion options for measuring the quality of a split; they are gini and entropy.

* Gini – Gini impurity is used to predict the probability of wrongly classifying a node when numbers are randomly selected. The lower the Gini impurity, the higher the similarity of the nodes. A Gini impurity of 0 means it is a perfect match, and 100 means perfect impurity.
* Entropy – is the measure of disorder. When entropy is 0 the match is identical (Géron, 2019)

From the sklearn library, DecisionTreeClassifier was imported, and the model was fit to the training and test data so it could learn (Jupyter: Footfall\_2\_models [22]). The defaults are used for this classifier which means gini will be used as the criterion function for the node splitting. Later during cross-validation with GridSearchCV, entropy will be used to see which performs the best.

A tree is the best way to visualise the splitting of the nodes. Tree from sklearn supports a tree visualisation; see Figure 17 below. The different colours represent the street: green is Henry\_Street, purple represents 0\_Connell\_St, orange is Capel\_Street, and blue is Mary\_Street (Jupyter: Footfall\_2\_models [27]).

Chart, timeline

Description automatically generated

Figure : Decision tree

#### Random Forest

Random forest is a supervised machine learning algorithm used primarily for classification and regression. The forest consists of many decision trees (Donges, 2021) and when a class label is predicted by the tree, the forest uses voting to decide on the class name.

Using multiple trees reduces the risk of overfitting, which is often a problem seen when using decision trees (Guido, 2017). Similar to a decision tree, it uses gini and entropy to measure the quality of the split. A visual representation of the forest, Figure 18, was created using the tree from sklearn and a loop to loop through the trees in the forest (Jupyter: Footfall\_2\_models [30]).

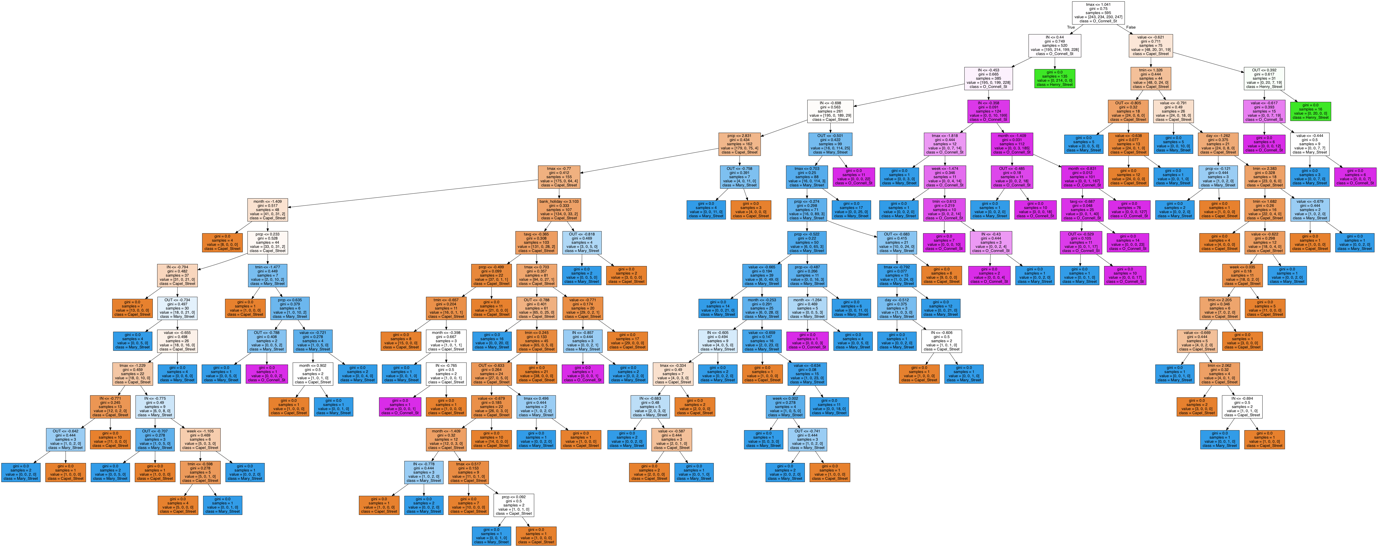


Figure : Random Forest

#### k-nearest neighbors

K-nearest neighbor(KNN) is a supervised machine learning algorithm. It predicts on what the target variable will be by finding the nearest data point to it in the training data. The first step is to determine the k value, the number of neighbours for the model. Using the KNeighborsClassifier from sklearn, a loop was created to iterate through 150 neighbours and fit the model to the training data and measure the accuracy score of each neighbours performance (Jupyter: Footfall\_2\_models [38]). Figure 19 shows the plot with KNN testing and training accuracy and about 31 neighbours are optimal, the lines are close, but there is still a gap, allowing for some space avoids overfitting the model to the training data. Based on the chart, 31 neighbours is the sweet spot for this model.

Chart, scatter chart

Description automatically generated

Figure : KNN number of neighbours vs accuracy

Another critical parameter in KNN is the distance metric and it can have a substantial effect on the performance (Lavrenko, 2014) of the model. There are many distance metrics to choose from, one is Euclidean distance which is the length of a line between two points.

The default distance metric, Minkowski, was used but later when using GridSearchCV, all hyperparameters will be passed in to find the optimal distance metric for the model.

As the project developed, KNN saw the most significant model improvement with fewer neighbours required and the accuracy significantly improved see Figure 23.

#### Logistic regression

Logistic regression is a supervised method where works for both classification and regression (Guido, 2017). It tries to model the probability of a particular class or event taking place. LogisticRegression classifier from sklearn was applied to training and test data (Jupyter: Footfall\_2\_models [45]). The coefficient shape is (4, 12) meaning that there are 4 classes and 12 features.

### Unsupervised

#### k-Means Clustering

k-means is an unsupervised learning algorithm and therefore, labelled data is unnecessary.

This clustering algorithm tries to divide data into clusters and discover underlying patterns (Guido, 2017). KMeans from the sklearn library was used as the classifier, the number of streets we need to identify are 4, therefore k is set to 4 (Jupyter: Footfall\_2\_models [52]). K is the number of centroids, centroids represent the center of the clusters. If K is unknown, an elbow test can be used to determine K when there is a bend in the plot that corresponds to K (Jupyter: Footfall\_2\_models [55]).

The columns IN and OUT are passed into the model without splitting into test and train as this is not needed for unsupervised algorithms. These columns were selected as they are the features with the most importance in previous models, see Figure 22. KMeans uses euclidean distance to measure the distance between two K dimensional vectors. Using Figure 19, the model has demonstrated how well it has done at displaying clusters (Jupyter: Footfall\_2\_models [54]).

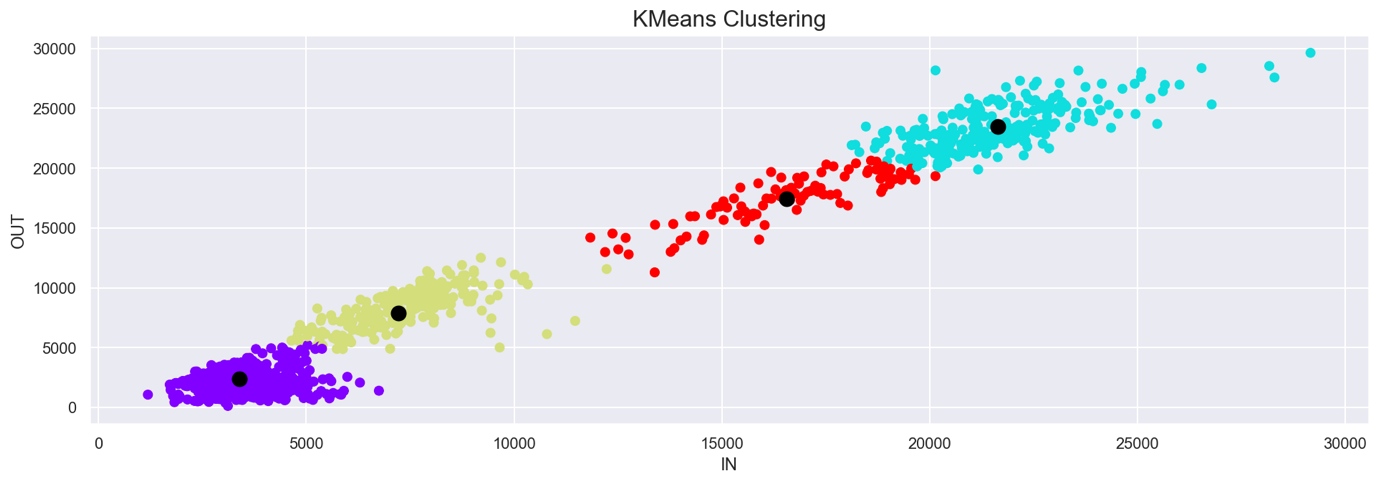


Figure : KMeans Clustering

## Evaluation

### Classification reports

Classification reports show the precision, recall, F1 score of the model, grouping it by the street name to identify what street had the best precision and performance, see (Jupyter: Footfall\_2\_models [24, 34, 42, 48]).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithms |  | Precision | Recall | F1-score |
| **Decision Tree** | Macro avg | 0.94 | 0.93 | 0.93 |
| **Random Forest** | Macro avg | 0.93 | 0.92 | 0.93 |
| **KNN** | Macro avg | 0.93 | 0.92 | 0.93 |
| **Logistic Regression** | Macro avg | 0.94 | 0.93 | 0.93 |

Figure 21: Classification report

### Confusion matrix

Another way to quickly visualise the accuracy of a classification algorithm is to create a confusion matrix, Figure 22. The diagonal points represent where the predicted and actual values are equal. The box with the most values has the lightest colour, and in this case, each model has light coloured boxes. Mary\_Street stands out as having a high number of mislabelled; it was mistaken for Capel\_Street by each of the models. There are about 20 points on average mislabelled. There are also some mislabelling issues with Capel\_Street being labelled as Mary\_Street. The data for Capel\_street and Mary\_Street and their values for OUT, IN, and value were very similar in range and it would be difficult to identify a pattern.

|  |  |
| --- | --- |
|  |  |
|  |  |

Figure : Confusion matrices for all the models

### Feature importance

Feature importance refers to the technique of assigning a score between 0 and 1 to the input features based on how useful they are to the predictive model, the greater the score the more significant the feature importance (Guido, 2017). Feature importance is a built into both Decision tree (Jupyter: Footfall\_2\_models [37]) and Random forest (Jupyter: Footfall\_2\_models [10]) algorithms. Figure 23 shows the feature importance in descending order of the input features. OUT and IN are top of the graph for both models. Logistic regression uses coefficients which unlikely feature important can be negative or positive. IN and value are negative

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Chart  Description automatically generated |  |  |

Figure : Feature importance for Decision tree and Random Forest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithms | Type | Accuracy | After removing outliers | Removing outliers with street group by | Replacing null values with median |
| **Decision Tree** | Classification | 0.89 | 0.89 | 0.87 | 0.91 |
| **Random Forest** | Classification | 0.91 | 0.91 | 0.91 | 0.91 |
| **k-nearest neighbors** | Classification | 0.58 | 0.57 | 0.67 | 0.69 |
| **Logistic regression** | Regression | 0.88 | 0.84 | 0.87 | 0.92 |

Figure : Machine learning performance

Finding the perfect parameters for a model can be time-consuming, especially with each model having multiple parameters and configurations. GridSearchCV from scikit-learn helps find the optimal parameters for each model which helps to improve the models’ overall performance and accuracy (Guido, 2017). GridSearchCV passes in all the available hyperparameters which allow to see what is the most optimal model configuration which can be accessed using best\_params\_. Initially, GridSearchCV was deployed on RandomForestClassifier passing the parameters and fitting to the training data to find the perfect model with the highest accuracy. It returned the best score of 93.4%, 2% better when compared to earlier model (Jupyter: Footfall\_2\_models [56-62]). The same premise is used for the rest of the models. However, a dictionary is defined, which contained all the models and their parameters (Jupyter: Footfall\_2\_models [64]) and a loop iterated through the dictionary. The list of models included: DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier, LogisticRegression. Once the loop finished it passed data into a dataframe such as the model, best\_score and best\_params which helps to compare to previous models (Jupyter: Footfall\_2\_models [65]). Figure 25 shows the algorithms vs algorithms finely tuned using GridSearchCV.

|  |  |  |
| --- | --- | --- |
| Algorithms | Non-optimised algorithms | GridSearchCV |
| **Decision Tree** | 0.91 | 0.93 |
| **Random Forest** | 0.91 | 0.94 |
| **k-nearest neighbors** | 0.69 | 0.76 |
| **Logistic regression** | 0.92 | 0.91 |

Figure 25: Non-optimised algorithms vs GridSearchCV

## Conclusion

Classification of the street name was easily predicted using supervised machine learning with decision tree and random forest returning the highest accuracy. Great improvement in the accuracy for each of the models occurred after each optimization steps was performed: imputation of null values with the median and removing outliers grouped by the street. After GridSearchCV was performed on the models and hyperparameters were passed in even further improvements were made in the models.

# Bibliography

Brownlee, J., 2020. *Feature Engineering and Selection (Book Review).* [Online]   
Available at: https://machinelearningmastery.com/feature-engineering-and-selection-book-review/#:~:text=Engineering%20and%20Selection-,%E2%80%9CFeature%20Engineering%20and%20Selection%3A%20A%20Practical%20Approach%20for%20Predictive%20Models,Modeling%2C%E2%80%9D%20publ  
[Accessed 30 03 2020].

Casari, A. Z. a. A., 2018. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists.* First Edition ed. Beijing: O’Reilly Media, Inc.

Chapman, P., 2000. *CRISP-DM 1.0,* s.l.: SPSS.

Donges, N., 2021. *A Complete Guide to the Random Forest Algorithm.* [Online]   
Available at: https://builtin.com/data-science/random-forest-algorithm  
[Accessed 26 03 2022].

Fumo, D., 2017. *Types of Machine Learning Algorithms You Should Know.* [Online]   
Available at: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861#:~:text=Supervised%20learning%20algorithms%20try%20to,from%20the%20previous%20data%20sets.  
[Accessed 26 03 2022].

Géron, A., 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.* Second Edition ed. Sebastopol: O’Reilly Media.

Graham, D. C., 2016. *The Relationship between High Street Footfall, Attraction & Conversion,* Cyprus: s.n.

GRUBBS, F. E., 1969. *Procedures for Detecting Outlying Observations in Samples,* s.l.: Technometrics.

Guido, A. C. M. a. S., 2017. *Introduction to machine learning with Python : a guide for data scientists.* 1st Edition ed. Beijing: O’reilly.

Lan Yang, J. A. C., 2020. *Use Case and Performance Analyses for Missing Data Imputation Methods in Big Data Analytics.* Pomona, ICCDE 2020: Proceedings of 2020 the 6th International Conference on Computing and Data Engineering.

Lavrenko, V., 2014. *k-NN 4: which distance function?.* [Online]   
Available at: https://www.youtube.com/watch?v=\_EEcjn0Uirw  
[Accessed 02 04 2022].

Lior Rokach, O. M., 2014. *Data Mining with Decision Trees: Theory and Applications.* 2nd Edition ed. Singapore: World Scientific Pub Co Inc..

Meteostat, n.d. *Access and analyze historical weather and climate data with Python..* [Online]   
Available at: https://pypi.org/project/meteostat/

Potter, K., 2006. *Methods for Presenting Statistical Information: The Box Plot,* Salt Lake City: University of Utah.

Qaiser, U. S. a. H., 2014. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research,* 12(1), pp. 217-222.

Roderick Duncan, T. B. ,. S. d., 2014. *The defence of bricks and mortar retailing.* Bordeaux, CAL-TEK S.r.l..

Roy, B., 2020. *All about Feature Scaling.* [Online]   
Available at: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35  
[Accessed 26 03 2022].

Santos, A. A. a. M., 2008. *KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW,* Amsterdam: IADIS European Conference on Data Mining 2008.

scikit-learn, 2022. *Importance of Feature Scaling.* [Online]   
Available at: https://scikit-learn.org/stable/auto\_examples/preprocessing/plot\_scaling\_importance.html  
[Accessed 26 03 2022].

Villeneuve, P. J., 2018. *Distribution, Normal.* [Online]   
Available at: https://www.encyclopedia.com/science-and-technology/mathematics/mathematics/normal-distribution#3  
[Accessed 30 03 2022].

Weiss, N. A., 2017. *Introductory Statistics.* Tenth Edition ed. London: Pearson Education.