DATA ANALYSIS REPORT FOR WORLD OF BARGAINS

STUDENT ID: 2644995

MODULE CODE: ITNPDB6

Executive Summary

Data mining is an approach to find patterns, anomalies and correlations from large datasets to generate valuable insights and better outcomes by using various machine learning techniques. The information yielded from this multi-disciplinary approach can be used to increase revenue, reduce business risks, cut costs and ultimately improving customer relationships.

This report has been prepared to assist management team of World of Bargains, a retail chain in UK in effectively running the business by building a computer program for prediction and classification. For this purpose, various machine learning techniques have been used and it was found that Ridge Regression outperforms Lasso Regression and Multilayer Perceptron (MLP) in case of prediction task while Multilayer Perceptron (MLP outperforms Decision Tree & Logistic Regression in case of classification task (the techniques and results have been explained in later stages in this report).

Contents

1. Background

As we are aware, troves of data have been generated and collected by Retail organizations on a daily basis. However, mere generating and collecting it for recording purpose doesn't make sense if is not translated into useful insights to make better decisions in the future. Using this approach, companies can stay ahead of their competitors by lowering operating costs (Jain, Menon, Chandra, 2015) and open doors for them to gain considerable market share.

1.2 Introduction to the task

The primary task of this project is to help Mr. Ivor Buquetlowd, the owner of World of Bargains Chain, which has over 100 shops in the UK. He wants a better system to run his shops, which enables him to select new shops more efficiently. Through this system he wants to predict the revenue of the shops as and when he puts data (generating actionable insights) based on which he wants to assess the performance of the shops so as he can harness productive information which will improve his decision making. The objective of this project is to extract hidden patterns in the data given by him by using different data mining algorithms to compare the results and then suggesting with the one with maximum accuracy.

1.3 Goal of the project

The goal of the project is to build a computer program to assist management team of World of Bargains to achieve following business objectives:

- To *predict* revenue of existing shops and compare with actually generated one with the help of specific feature values. It helps to know what shop features are mostly contributing to the revenue and can be considered for new shop locations as well.
- To efficiently *classify* performance of shops into 'Poor', 'Reasonable', 'Good' or 'Excellent' categories.

To achieve the objectives of the project, a data-mining framework called as CRISP-DM (explained below) is used and the technique which has been used is 'Machine Learning' that learn from existing data and look for patterns in order to adapt behaviour for future events. Various machine-learning models such as Linear Regression, Logistic regression, Multilayer Perceptron, Decision tree have been applied to reach at the best model. These models are explained at the later stage in methodology section and results are illustrated in Results and Errors section along with recommendations in the last section.

1.4 Data Mining Framework: CRISP-DM

For a part of this project, I have used Cross Industry Standard Process for data mining (CRISP-DM) for designing the prediction and classification models. It is a cycle of six stages that comprises of the tasks to analytically solve a business problem (Azevado, Santos, 2008) which are described as follows:

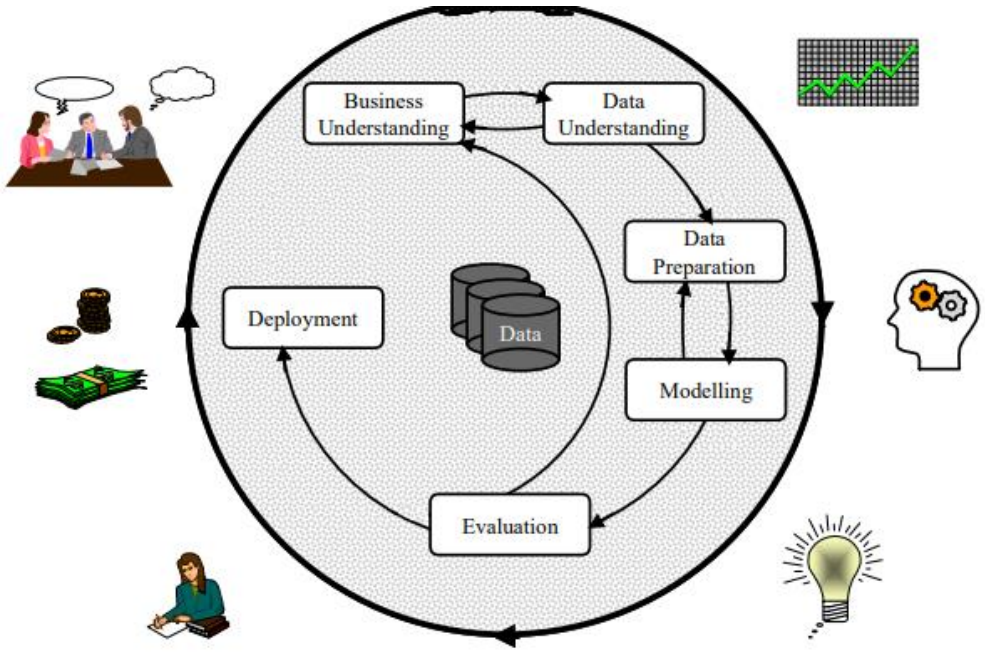


Figure (1): *Stages in a CRISP-DM approach.*

Source: https://s2.smu.edu/~mhd/8331f03/crisp.pdf

1. **Business Understanding**: It refers to convert your knowledge from business perspective into a data mining tasks by focusing on understanding the core objectives of a business problem in question.
2. **Data Understanding**: It involves the task to understand the data from quality point of view and discover first insights for further analysis.
3. **Data Preparation**: After understanding the data given by the enterprise, the next step is to study it and examine any anomalies at the initial stage in order to correct them before building a model for the final output. It involves finding hidden and unknown patterns from the provided data set.
4. **Modeling**: It involves calibration of parameters, selecting and applying various modelling techniques to the business problem in question.
5. **Evaluation**: It involves critical evaluation of the models obtained along with thorough analysis of the steps involved in building the model in order to achieve the business objectives and for successful implementation.
6. **Deployment**: The last step is to deploy the best model in the business enterprise so as the knowledge obtained should be used for its successful implementation.

1.5 World of Bargains & Data Mining Application:

The knowledge gained after computer-driven exploration of the data enhancing pattern recognition, visualization, artificial intelligence capabilities (Fayyad et al., 1997) is very crucial to implement it for productive use. It provides with the novel and understandable insights from the analytical data during the data mining process. (Fayyad et al.1996).

The data preparation, data cleaning forms part of Knowledge Discovery Process (KDD) and then with the help of mathematical models, knowledge can be mined to make sense out of the data (Feyyad, 1996)

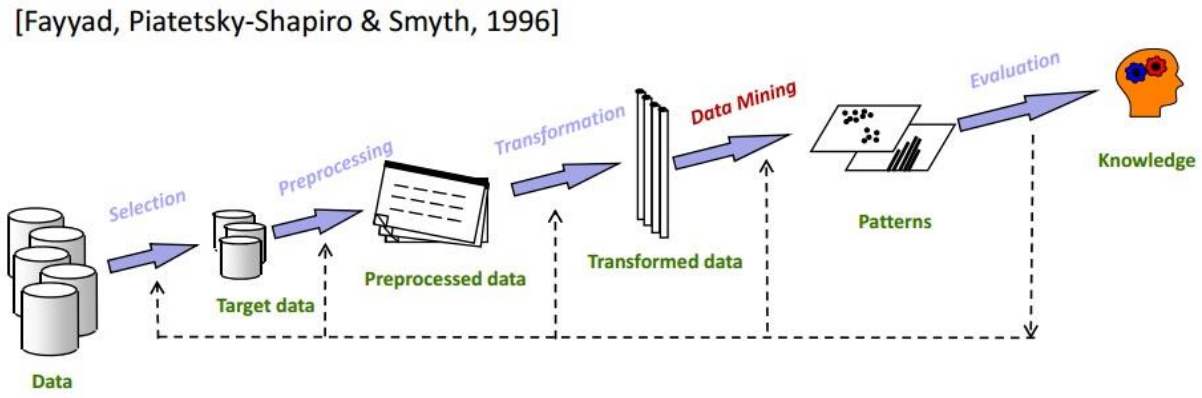


Figure (2): *Data mining v/s KDD.*

Source: https://nocodewebscraping.com/difference-data-mining-kdd/

Pursuant to the goal of the project, i.e. to empower the decision making process of World of Bargains by building a computer program which direct them to run the business more effectively. Furthermore, the key task is to build a model for predicting the revenue so as the enterprise can choose store locations wisely and to help them in reducing the costs by building a model for classifying the performance of the specific stores. Therefore, the problem is related with extracting hidden patterns from the data given by them and discovers the appropriate knowledge, to implement it, for the success of the business enterprise. **Hence, Data Mining is the most suitable approach to be used to provide solution to the business problem in question.**

2. Data Preparation

2.1 Data Summary

The s dataset provided, was received as a CSV file containing a (137x20) matrix of data. The table below shows the type of attributes (object=discrete and integer=continues).

```
Data columns (total 20 columns):
Town                  136 non-null object
Country               136 non-null object
Store ID              136 non-null int64
Manager name          136 non-null object
Staff                 136 non-null int64
Floor Space           136 non-null int64
Window                136 non-null int64
Car park              136 non-null object
Demographic score     136 non-null int64
Location              136 non-null object
40min population      136 non-null int64
30 min population     136 non-null int64
20 min population     136 non-null int64
10 min population     136 non-null int64
Store age             136 non-null int64
Clearance space       136 non-null int64
Competition number    136 non-null int64
Competition score     136 non-null int64
Profit                136 non-null int64
Performance           136 non-null object
```

Table (1): *store dataset's attribute description*

The features labels can be seen below:

- Floor space – the shop floor area
- The Window space - is the measurement of store window area, used for advertising/marketing
- The Car Park has four classes 'Yes', 'No' ,'Y','N'
- Demographic Score - a score of how well a given prospect fits your target audience
- Location - '*Shopping Centre*, '*High Street'*, '*Retail Park' or 'Village'*
- 40min, 30min, 20min, 10min – the total population for a distance time
- Store age – the age of the store
- Clearance space in store – the floor space in square feet, dedicated to reduced-price products
- Competition number – how many competing stores are in the proximity to a store branch
- Competition score – a value representing the competitor 'strength' associated with a given location

The figure below shows how each attribute is distributed, if look at the histogram of each attribute. The histograms show only profit and performance are normally distributed.

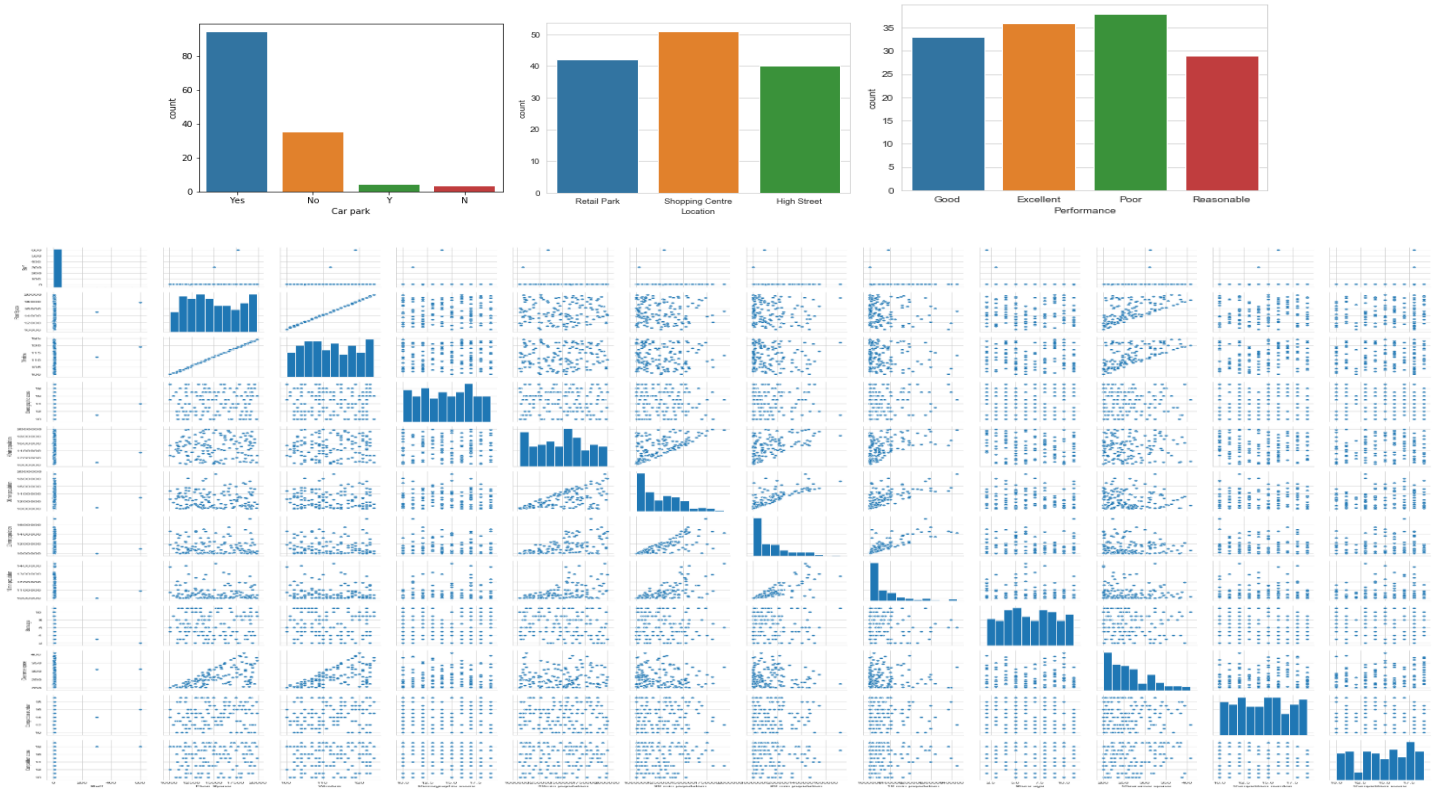


Figure (3): *Pair plot (to see how attributes are distributed)*

2.2 Data Pre-processing

This section describes how the data used in the project have been collected and prepared for the analysis. All three datasets have been imported from the original source, processed and converted into a format easy to manipulate in a Python environment.

2.3 Data Cleansing

Cleaning data is the process of preparing data for modeling or analysis.it is important because data never comes in processed form that it does not required to be studied which forms part of cleaning process. During the exploratory data analysis in Python, it was noted that four columns contained the data, which have no impact on the performance of the model. These variables are labeled flat or wide variables and add no value to the model performance. More specifically, the '*Town'* feature vector consisted entirely unique entries and the '*Country'* feature vector consisted of identical entries down the vector. After removing attributes Town, Country, ID and Manage name , The firs step is checking if there is any outlier or missing value or any correction need to be done if different nominated used for same mining .

- **Missing value**

After checking and visualizing the dataset by using info () attribute in python, the result showed the dataset does not have missing value.it also showed us the type of dataset ‘s attributes and how many columns and rows the data set has (136 rows and 20 columns)

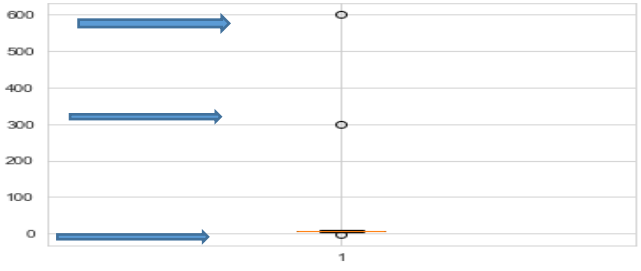
```
RangeIndex: 136 entries, 0 to 135
Data columns (total 20 columns):
Town                    136 non-null object
Country                 136 non-null object
Store ID                136 non-null int64
Manager name            136 non-null object
Staff                   136 non-null int64
Floor Space             136 non-null int64
Window                  136 non-null int64
Car park                136 non-null object
Demographic score       136 non-null int64
Location                136 non-null object
40min population        136 non-null int64
30 min population       136 non-null int64
20 min population       136 non-null int64
10 min population       136 non-null int64
Store age               136 non-null int64
Clearance space         136 non-null int64
Competition number      136 non-null int64
Competition score       136 non-null int64
Profit                  136 non-null int64
Performance             136 non-null object
dtypes: int64(14), object(6)
```

Table (2): *it shows .dataset has 136 entries for each columns*

- **Outliers**

Outliers has many definition but in one definition means the value that is higher or lower than the rest data and it can pose a potential problem and need to be investigated .There are couple of ways for finding outliers , in this project attribute describe () used which showed attribute Staff has outliers because of big difference between maximum and minimum .For solving this issue the function of Outlier defined and it find three values in Staff columns as outliers which are 600,300,and -2. I decided to drop them.

	Store ID	Staff
count	136.000000	136.000000
mean	52172.580882	13.352941
std	28725.990237	56.592439
min	1437.000000	-2.000000
25%	24707.250000	6.000000
50%	56011.500000	7.000000
75%	76870.500000	8.000000
max	99854.000000	600.000000



Figure(4):*the table and boxplot show the staff outliers(-2,300,600)*

- **Inconsistent data**

Car park attributes has four classes but two of them are Inconsistent data which has been caused by data entry errors where the instances took values in the set {Yes, No, Y, N}.So “Y” and “N” (portrayed as bad data) were replaced as “Yes” and “No”. Also ,‘Country’ Feature – Potential data entry error (instance 41 & 97), where the town listed was confirmed geographically to be in the UK and not in France – these data entry errors were amended to reflect towns in the UK.

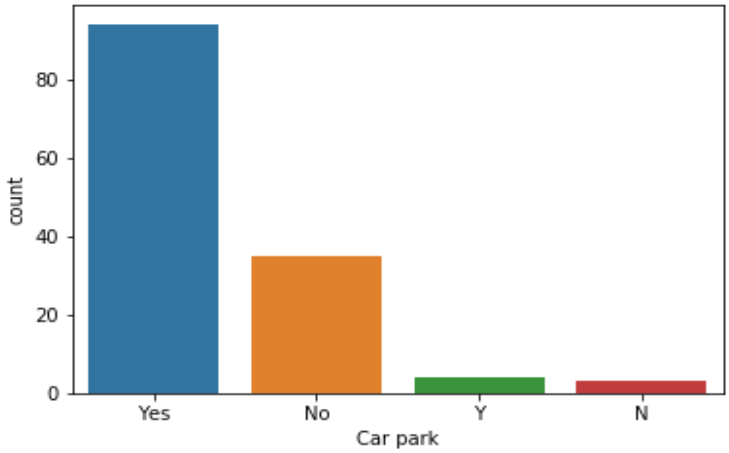


Figure (5): *Car park's class's frequency*

2.4 Encoding categorical variables

Models in scikit-learn require numerical input, then it necessary to encode them. However it is good practice to create a dataset where Performance class labels ("Excellent","Good","Poor","Reasonable") are encoded as integer values ('1' , '2','3','4') through a mapping process. IN this project two different ways used to encode categorical first with replacing them with numbers and other one applying **get_dummies ()** attribute from Panda package to encode 1s and 0s in dataset (location column).

Car park	Location_High Street	Location_Retail Park	Location_Shopping Centre
1	0	1	0
1	0	0	1
0	1	0	0
0	0	1	0
1	0	1	0

Table (3): *location as dummy variables*

2.5 Standardization

It is always possible; we will come across dataset with lots of numerical noise built, such as lots of variance or differently scaled data. The preprocessing solution for this is Standardization. Standardization is a preprocessing method used to transform continuous data to make it look normally distributed.in scikit-learn, this is often necessary step, because many models assume that the data we are training on is normally distributed and if isn't, you risk biasing our model. We can standardize our data in different way, but in this report we scaled data with **MinMaxScaler ()** function from scikit-learn in python environment This feature scales and translates each feature individually such that it is in the given range on the training set, between zero and one. It is also important to note that standardization is a preprocessing method applied to continues, numerical data.

	Staff	Floor Space	Window	Demographic score	40min population	30 min population	20 min population	10 min population	Store age	Clearance space	Competition number	Competition score
count	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000	133.000000
mean	0.475564	0.505461	0.501880	0.501253	0.476283	0.269606	0.187855	0.156699	0.506266	0.298185	0.496241	0.526316
std	0.360009	0.283020	0.287848	0.320346	0.290540	0.234372	0.207684	0.209869	0.308386	0.249166	0.324677	0.327644
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.250000	0.284169	0.291667	0.222222	0.235791	0.061511	0.017270	0.013178	0.222222	0.100962	0.222222	0.333333
50%	0.500000	0.460957	0.458333	0.555556	0.504428	0.225726	0.105672	0.079145	0.444444	0.235577	0.555556	0.555556
75%	0.750000	0.761356	0.750000	0.777778	0.689814	0.431052	0.291262	0.212585	0.777778	0.437500	0.777778	0.777778
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Table (4): statistical description for continues variable

2.6 Cross-validation

Cross-validation is a vital step in evaluating a model. It maximizes the amount of data that is used to train the model. For a high-quality measure of model performance, the dataset was once subdivided notably into two sets: a training set and separate test set. This is to ensure the model built now not only performs properly on the education statistics but also additionally generalizes nicely on unseen data. Furthermore, the education set is loaded into Python and the K-fold cross-validation approach is used to evaluate model performances and optimize the hyperparameters chosen. K-fold cross-validation can be conceptually seen under in **Figure 3**, where the education dataset provided is subdivided into K subsets. In large education datasets, K is generally chosen as 10, alternatively as the education dataset is extraordinarily small, K is chosen as 5. We begin by splitting the dataset into five group or folds, then we hold out the first fold as a test set, fit our model on the remaining four folds, predict on the test set, and compute the metric of interest, then similarly with second, third, fourth and fifth fold. This method avoids the problem of our metric choice being dependent on the train, test split.

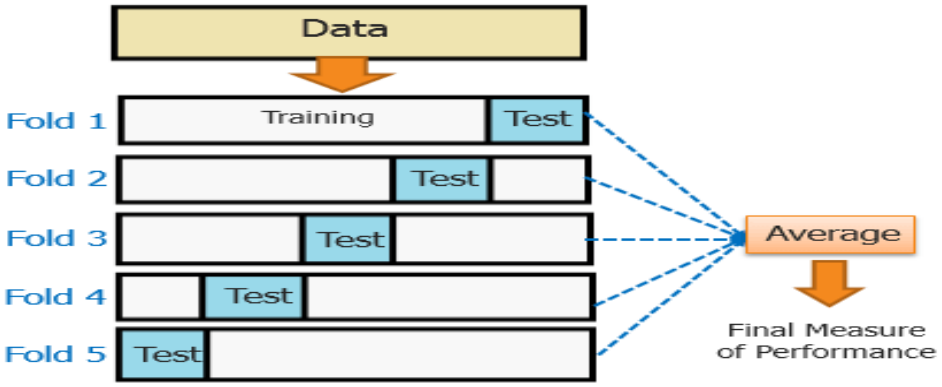


Figure (6): cross validation structure

To perform this method, the sikit-learn package is imported. Finally, the model is retrained on all the data provided by World of Bargains to produce a concluding set of models, ready for commercial implementation. Based on the dataset provided, our train set is going be all features except “profit” and “Performance”, and once test set as “profit” and once as “performance”.

2.7 Feature Selection

Once we've settled one a feature set for modeling ,its important to really consider these features.do we need all of them, and do we know how they will impact our model? Feature selection is a method of selection feature from dataset feature set to be used for modeling.it draws from a set of existing feature, so it is different feature engineering because it doesn't create new features. The overarching goal of feature selection is improve our model's performance. It helps to get rid of noise in our model Scikit-learn has several methods for automed feature selection, but in this report random forest**,** and Lasso Regression used as method to perform feature selection**.** Also correlation is a important way ,because maybe we have feature that are strongly statistically correlated, which breaks the assumption of certain model and thus impact model performance .

3. Modelling

3.1 Supervised Learning

In supervised learning, we have several sample and data points described using predictor variable or features and a targets variable. Our data is commonly represented in a table structure such as what we see in **figure (),** in which there is row for each data point and a column for each feature. The aim of supervised learning is to build a model that is able to predict the target variable, which are "profit" and "Performance" in our dataset. For supervised learning, we need labeled data and there are many ways to get it, but our data set is labeled, because we know that our output is going to be "profit" or "performance".

3.2 Regression

An approach which is used for finding the relationship between a dependent variable and one or more independent variables is known as Regression. When there is just one independent variable in relation to dependent variable it is called as simple linear regression, and the case of more than one independent variable, it is called as multiple linear regression. We know by now that linear regression gives us one line equation. Moreover, in multiple linear regression we will, again, get a line equation at the end, but here the problem that we consider is trying to predict the value of a response variable "y" based on the value of more than one independent variable. Multiple linear regression is an extension of simple linear regression. And here's the equation for a general linear model [y is equal to beta-0 plus beta-1 times x-1 plus beta-2 times x-2 plus beta-k times x-k plus error term]. And we can predict the dependent variable "y" more accurately by using these independent variables. "x1", "x2", "xk" are the independent variables, and "y" is the dependant variable. So "y" is the response variable that we want to predict. And "β0", "β1", "β2", and "βk" are the unknown constants. "x"'s here are the independent variables, and they are measured without error. So when we are predicting the "y", the dependant variable, then we see the error term. In addition, without the error term, this part of the equation is given by:

"E(y)" [$E(y)= \beta0+\beta1x1+\beta2x2+\ldots+\beta kxk$]. And this is the deterministic part of the model.

3.3 Ridge Regression

Feature selection recall that what fitting regression does is minimize a loss function to choose a coefficient ai for each feature variable if we allow those coefficient or parameters to be super large ,we can get overfitting. For this reason, it is common practice to alter the loss function so that it penalize for large

coefficients. This called Regularization .in this project the first type of regularized regression that I used in this report is Ridge regression ,in which our loss function is:

$$\text{OLS loss function} + \alpha * \sum_{k=1}^{n} \alpha^2$$

Thus, when minimizing the loss function to fit our data, model are penalized for coefficient with a large magnitude: large positive and large negative coefficient. Note that alpha is a parameter we need to choose in order to fit and predict then choosing alpha is a hyperparameter for Ridge regression. The method of performing ridge regression is also from Sckit-leanrn package in python environment. I used this method as model to predict "profit".

3.4 Lasso Regression

Another type of regularized regression that used in this report is Lasso regression, in which our loss function is:

$$\text{OLS loss function} + \alpha * \sum_{k=1}^{n} |\alpha i|$$

The method of performing lasso regression in scikit-learn mirrors ridge regression. One of the aspect of lasso regression is that it can be used to select important features of a dataset. This is because it tends to shrink the coefficients of less important features to be exactly zero. The feature whose coefficient are not shrunk to zero are 'selected by LASSO algorithm .For lasso alpha is a hyperparameter . I used this method as a model to predict "profit" and also see which features are most importance to building the best model for profit prediction.

3.5 Logistic Regression

Logistic regression is used in classification problems, not regression problems. Binary classification, that is, when have two or more possible labels for variable. For instance a 'False' or 'TRUE' category represented by an output in the form of a probability mapping, [1, 0] represents 'yes' and [0, 1] represents 'no'). Given one feature, log reg will output a probability, p, with respect to the target variable (here in dataset our target is "Performance"). Note that reg produces a linear decision boundary.in this report logistic regression is perform from Scikit-learn in python environment However, as the goal of this classification task is to categorise a multi class classification ('Poor', 'Reasonable', 'Good' or 'Excellent'), an approach called one-vs-all method is used. This extends the binary approach outlined above, however trains the logistic regression classifier for each class category required, to predict the probability that the output calculated, falls into a given category. This output solution can be expressed in a similar fashion as "1", representing 'Poor', "2", representing 'Reasonable', "3", representing 'Good' and finally "4", representing 'Excellent'. The logistic regression model is represented or defined by a matrix of weights (or coefficients) - one vector of coefficients for each class (which are combined to

produce the model coefficient matrix). Note that in Logistic Regression the “penalty”,”C “and “Solver “are hyper parameter for model.

3.6 Multi-layer Perceptron

A multi-layer perceptron MLP is a feed-forward artificial neural network model where a set of input data is mapped onto a set of appropriate outputs. It consists of multiple layers or nodes in where each layer is fully connected to the next one in the form of a directed graph. Figure below shows how a typical structure can look like. As figure shows the input layer (from dataset) has multiple inputs here-

X1,X2,…Xp are input which are sent to the network.it also shows it has three hidden layer and that is to intermediary layers to the output layer . MLPRegressor has been used to make prediction for profit and MLPClassifier has been used (from scikit-learn) to classify Performance

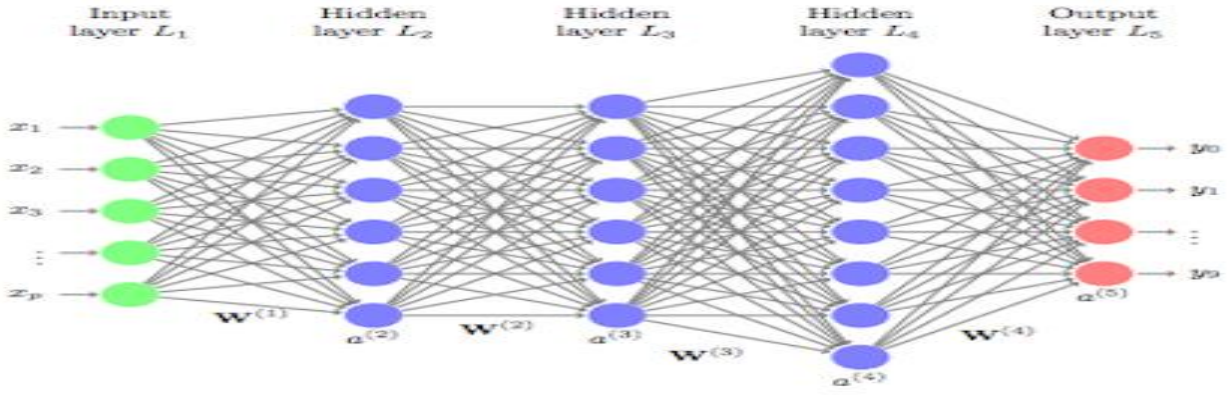


Figure (7):*multi-layer perceptron process*

3.7 Decision Tree

Sometimes a difficult or complex decision can be made simpler by breaking it down into a series of smaller decisions. Decision tree are used to find a set of if/else condition that are helpful for taking action. Decision tree structure closely resembles rea-world trees. The goal is to model the relationship between predictors and an outcome of interest. Beginning at the root node, data flows through if/else decision nodes that split the data according to its attributes. The branches indicate the potential choices ,and leaf nodes denote the final decision. These are also known as terminal nodes because they terminate the decision making process.

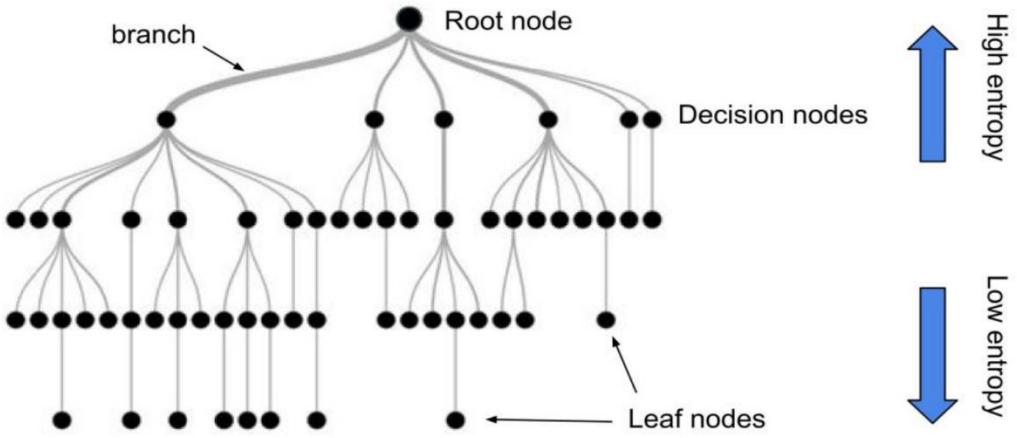


Figure (8): *Decision tree structure*

3.7.1 Decision tree for classification

Given a labeled dataset, a classification tree learns a sequence of if-else question about individual features in order to infer the labels. In contrast to linear models, trees are able to capture non-linear relationship between features and labels.(in this report our target labels are :'Poor', 'Reasonable', 'Good' or 'Excellent'). In addition, trees do not require the features to be on the same scale through standardization for example.(DecisionTreeClassifier is imported from Sicikt-learn into python environment).Note that ,A classification-model divides the feature-space into regions where all instances in one region are assigned to only one class-label.this regions are known as decision-regions.

3.7.2 Decision tree for regression

Recall that in regression, the target variable is continuous("Profit"). DecisionTreeRegressor is imported from Sicikt-learn into python environment. Its important to know that when Decision tree is trained on a dataset, the impurity of node is measured using mean-squared error of the targets in that node.

$$I(node) = MSE(node) = \frac{1}{Nnode} \sum_{i \in node}^{\infty} \left(y^i - \hat{y}\right)^2$$

$$\hat{y} = \frac{1}{Nnode} \sum_{i \in node}^{\infty} y^i$$

This means that the regression tree tries to find the splits that produce leafs where in each leaf the target values are on average,the closest possible to the mean-value of the labels in that particular leaf.as mentioned the targets variabl'y' is computed as the average of the target-variables contained in that leaf as shown in this formula :

$$Y^{(leaf)} = \frac{1}{Nleaf} \sum_{i \in leaf}^{\infty} y^i$$

4. Hyperparameter tuning

All the models described can be configured to elicit extraordinary behaviour via choice of the mannequin hyperparameters. This system is often called model hyperparameter optimization and is an empirical system of trial and error to select the combination of hyperparameters, which yields a suited model performance.for instance when we are fitting a linear regression,what are really doing is choosing parameters for the model that fit the data best. In addition, as said we need to choose the alpha in ridge and lasso regression before fitting it. Such parameters, one that need to be specified before fitting a model, are called Hyperparameter .In other word, these are parameters that cannot be explicitly learned by fitting the model. When fitting different values of hyperparameter,it is to use cross-validation as using train test split alone would risk overfitting the hyperparameter to the test set. The basic idea that used in this project is as follows: I chose a grid of possible values that I wanted to try for the hyperparameter.this is called a grid search and in Sikit-learn I implement it using the class GridSearchcv.

5. Results

5.1 Correlation

Before fitting machine-learning models (for both targets), it is a good approach to check features and targets using correlation. From the panda library of Python, the function Corr() is used to calculate Pearson correlation, and the heat map function has been used from Seaborn library to plot correlations. The plot shows colors ranging from purple for negative correlations to gray for positive correlation. The numeric values are also shown in each square. To understand it, look for the intersection of the two variables in the plot in the figure displayed below. As it shows the correlation between “Floor Space” and “window” is 1 or between all “population classes ” are high and also between “clearance space “and “Floor Space”, “window” is 0.63 which is high, as a result for modeling we can neglect one of floor space or window due to high correlation and only choose one of “population class”.

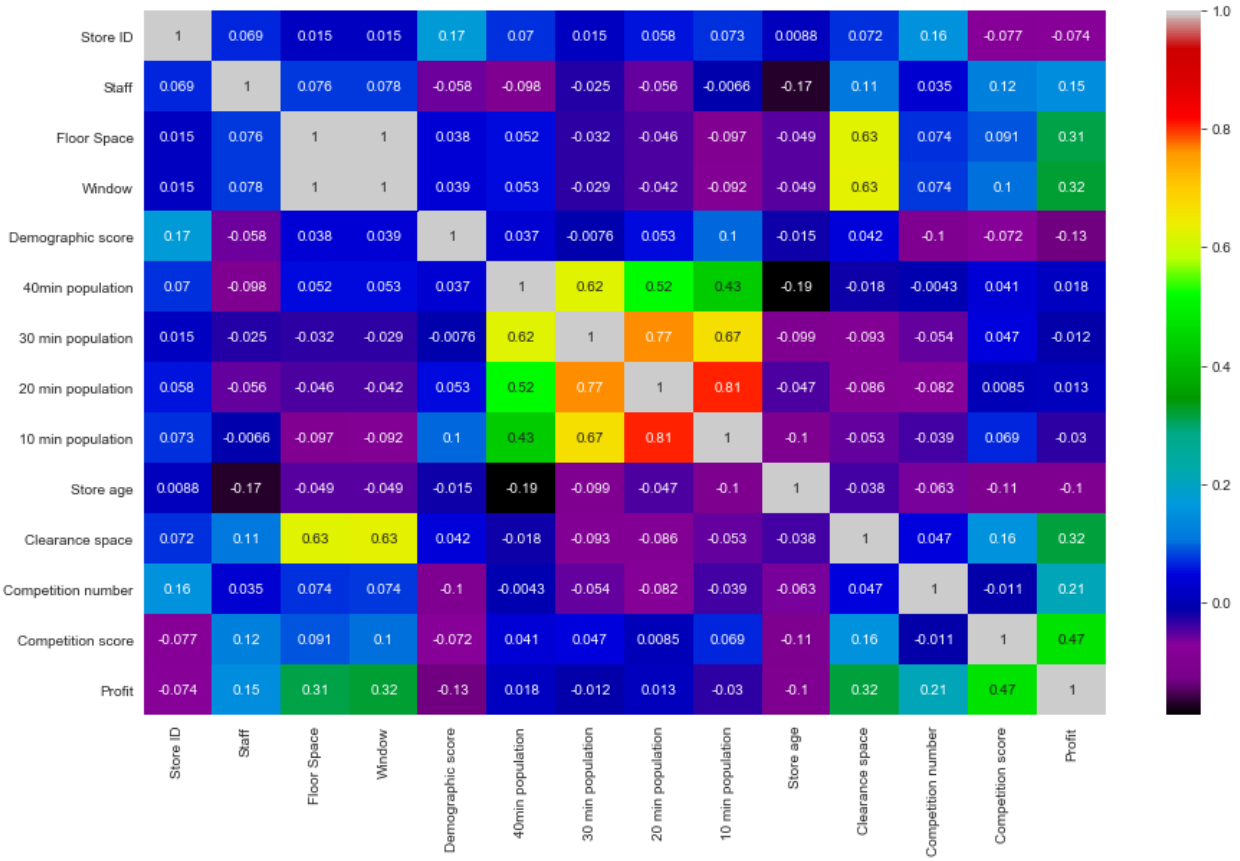


Figure (9): heatmap correlation between continuous variables

5.2 Profit Prediction

5.2.1 Checking the target normalization

First step is to see how our target is distributed,one of the way is to visualise it using histogram. For doing so, the histogram function has been used from Seaborn library to plot profit histogram. figure () depicts that, the profit has been distributed normally.

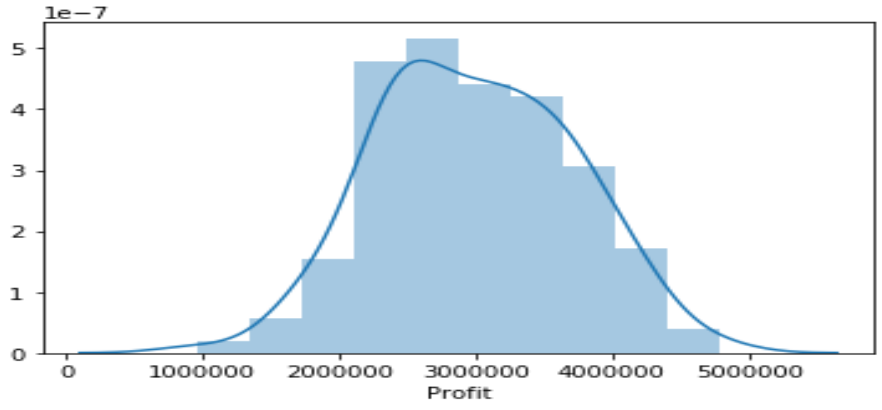


Figure (9): Profit distribution histogram

5.2.2 Feature Selection (Random forest)

In this research, three methods were used to find the best feature for predicting profit that are random forest, Lasso regression and correlation as mentioned earlier, by using ridge and lasso regression, they automatically choose the best features for themselves. All tree-based methods allow us to get 'Important features'. These are the scores representing how much each feature contributes to the prediction. For regression (when our target is continuous), this is how much each feature can help reduce the variance when it splits the data. Once random forest fitted, (RandomForestRegressor is imported from Sicikt-learn into python environment) the important features are extracted.

The hyperparameters evaluated for the Random forest were "bootstrap", "criterion", "max_depth",

- **Criterion:** The function to measure the quality of a split. Supported criteria are 'MSE' for the mean squared error, which is equal to variance reduction as feature selection criterion, and 'MSE' for the mean absolute error (1).
- **Bootstrap:** Whether bootstrap samples are used when building trees. If false, the whole dataset is used to build each tree (1).
- **max_depth:** The maximum depth of the tree. If none, then nodes are expanded until all leaves are pure or until all leaves contain less than 'min_samples_split' samples (1).

The result for hyperparameter after implementing Grid Cvsearch for Random Forest Regressor is shown as follows:

```
RandomizedSearchCV(cv=5, error_score='raise-deprecating',
          estimator=RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators='warn', n_jobs=None,
           oob_score=False, random_state=None, verbose=0, warm_start=False),
          fit_params=None, iid='warn', n_iter=100, n_jobs=-1,
          param_distributions={'max_features': ['auto', 'sqrt'], 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110,
None], 'bootstrap': [True, False]},
          pre_dispatch='2*n_jobs', random_state=42, refit=True,
          return_train_score='warn', scoring=None, verbose=2)
```

After finding the hyper parameter, Random forest model has been built to find the important Feature:

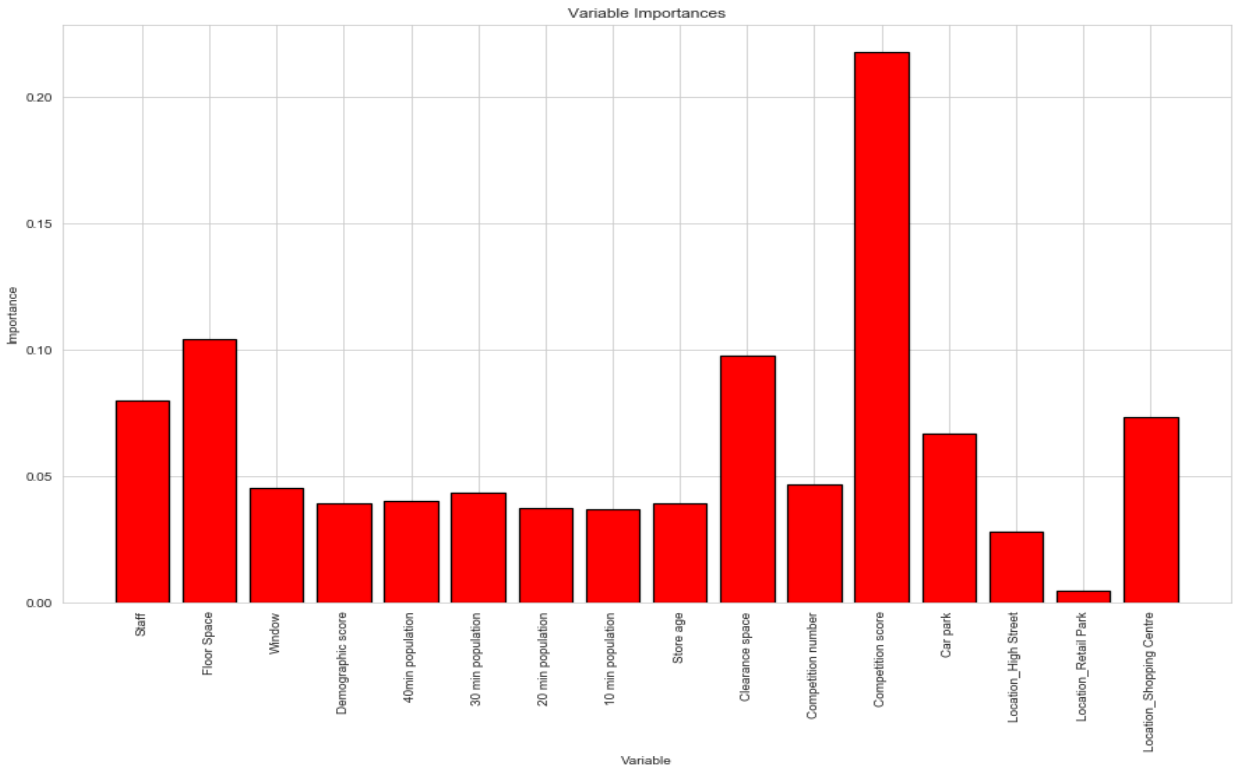


Figure (10): Random forest feature selection .The result regard random forest model shows that "Clearance space ", "Floor Space", "location", "Staff", "Window" , "Car Park", "Competition score " and '30 min population' are the most importance features(but it not final decision)

5.2.3 Regression

To evaluate the regression models, in this project Lasso and Ridge method are used and hyper parameter for both is α.

5.2.3.1 Ridge Regression

First, regression model has been built by using Ridge method (imported it from scikit-learn package into python environment). This method influences the extent by which the model complexity is reduced. This is done by preventing any learned coefficient, from becoming too large.

- **Alpha:** Regularization strength; must be a positive float. Regularization improves the conditioning of the problem and reduces the variance of the estimates. Larger values specify stronger regularization.
  By Using GridSearchCV method {'alpha':1:0}

The key metrics to measure Ridge Regression is Root Mean Squared Error (MSE), which means the aver age amount of error made on the test set in the profit variable, which we are trying to predict. As you can see below the Error difference between test and train is: 9322

R^2: 0.8198192814342431

Root Mean Squared Error for Test: 338802.47717484174
Root Mean Squared Error for Train: 431993.00912941934

The linear regression function:

Ridge model: 624145.356 * X0 + 288405.536 * X1 + 337718.462 * X2 + -141714.721 * X3 + -38083.538 * X 4 + 77645.424 * X5 + 357997.89 * X6 + -248746.395 * X7 + -38382.978 * X8 + 68740.454 * X9 + 383624.05 * X10 + 766812.261 * X11 + 356359.789 * X12 + -145797.791 * X13

Coefficients in this function portrayed the rate of change of Profit as a function of changes in the other fe atures. An R-squared of 0.81 means a perfect fit. Finally plotting needs to be done which is a quick way t o see how the model is going to make to predictions versus actual values.

Note that a perfect predictions yield a straight line.

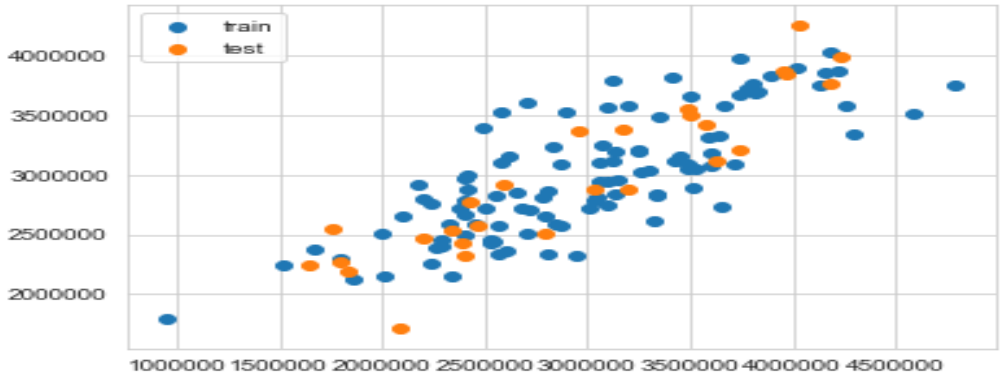


Figure (10): train and test actual vs predictions

Cross-Validation has been used because if R Squared is computed on train test (which is dependent on the way by which data has been splitted up), the data points (in the test set) may have some peculiarities that the computed value for R Squared is not representative of models ability to generalize to unseen data. *To deal with it, data has been splitted into five groups or folds*. In which first fold has been used a test set, model has been fitted on the remaining folds. As a result we got 5 values of R squared from which can compute statistics of interest. To perform 5 –cv in python, **cross –val-score** has been imported from scikit learn package. The result is sum of all R-squared values for our five folds, and the computed average is almost **70%** which is illustrated below:

[0.69039893 0.72059672 0.62807444 0.55715491 0.71707471]
Average 10-Fold CV Score: 0.6826599425511911

5.2.3.2 Lasso Regression

The method of performing lasso regression in scikit-learn is similar to what has been used for ridge regression. As mentioned earlier, Lasso is a good way for extracting important features of dataset. For Lasso Regression, **repressor** has been fitted into data and then **coefficient attribute** has been extracted and stored in **Lasso coef.**

In first step, the hyper parameter explored for lasso is alpha=20, then the model has been performed using it:

The accuracy of model is **85%** (80% training data):

R^2: 0.8547966121810538
Root Mean Squared Error for Test: 304145.07921865047
Root Mean Squared Error for Train: 427407.1925829806

The results shows that the error difference is 123,262.12

Root Mean Squared Error (MSE) which is the average amount of error made on the test set in the profit variab le, which we are trying to predict.
Plotting the result is a quick way to see how the model is going to make predictions versus actual values.

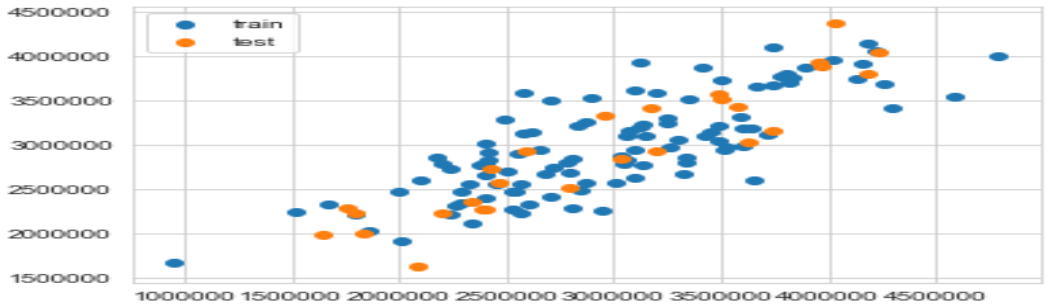


Figure (11): train and test actual vs predictions

The result is sum of all R-squared values for five folds, and as depicted below, the average is almost **66%**.
[0.63820349 0.65046262 0.63870569 0.57165498 0.68593054]
Average 10-Fold CV Score: 0.6569914627391467

Plotting the coefficients as a function of feature name yields following figure:

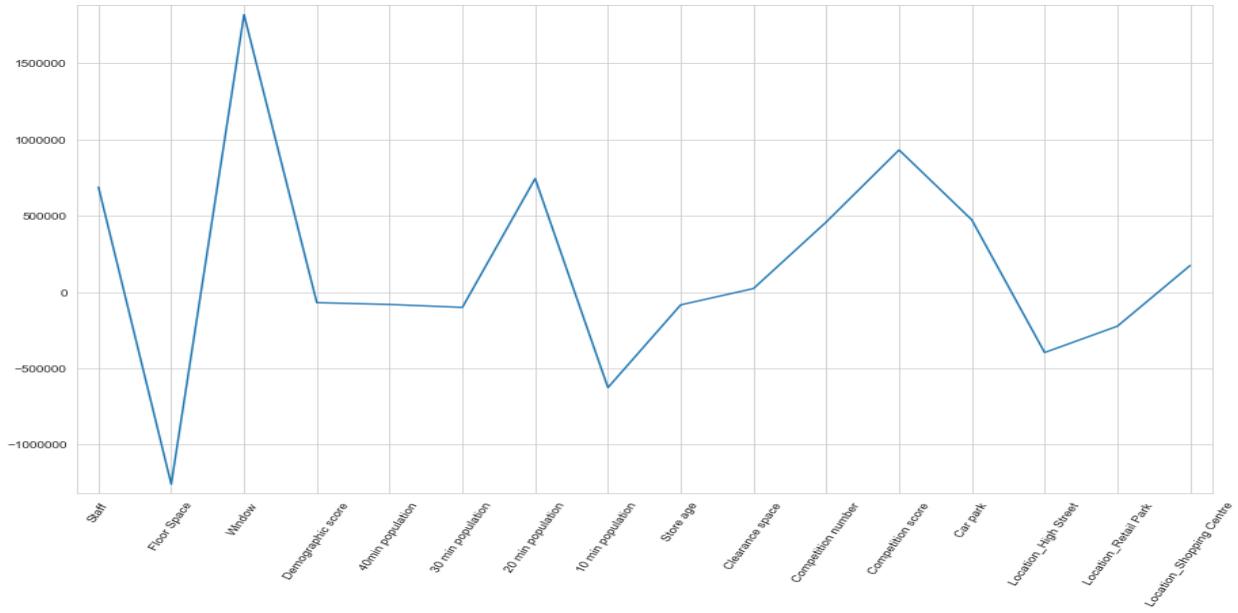


Figure (12):This figure show that "Staff", "Floor Space", "Window", "Competition score", "Car park" ,and "Location" are the most important predictors for our target(profit).but as you an see floor space and window performing same that because the correlation between them is 1 .

5.2.3.2 Conclusion on feature selection

The results in this segment portrayed that the Ridge regression has more accuracy in comparison to lasso but Lasso is more reliable due to the feature selection ability. Also in favor of correlation, based on results given by Random forest and lasso, **'Staff', 'Competition Score', 'Floor Space', 'Car park', 'Location'** have been chosen as most important features ,to build model for profit target(i.e. y=profit.)

5.2.3 Multi-layer perceptron :(MLPRegressor)

The aim of this segment is to make prediction for profit (target) by implementing multi-layer perceptron method. To do so, firstly MLPRegressor has been imported from scikitlean -neural network. Secondly, following hyper parameters have been explored for the model. (Most important hyperparameters)

Hyperparameters:

- **Solver:** The solver for weight optimization.
  Note: The default solver 'adam' works pretty well on relatively large datasets (with thousands of training samples or more) in terms of both training time and validation score. For small datasets, however, 'lbfgs' can converge faster and perform better (1).
- **Alpha:** L2 penalty (regularization term) parameter (1).
- **Hidden_layer_sizes :**
  The ith element represents the number of neurons in the ith-hidden layer (1).

The results:

```
GridSearchCV(cv='warn', error_score='raise-deprecating',
       estimator=MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(100,), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
       random_state=None, shuffle=True, solver='adam', tol=0.0001,
       validation_fraction=0.1, verbose=False, warm_start=False),
       fit_params=None, iid='warn', n_jobs=None,
       param_grid={'hidden_layer_sizes': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], 'activation': ['relu'], 'solve
r': ['adam'], 'learning_rate': ['constant'], 'learning_rate_init': [0.001], 'power_t': [0.5], 'alpha': [0.0001], 'max_ite
r': [1000], 'early_stopping': [False], 'warm_start': [False]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=True)
```

After finding, the hyper-parameter for the model, the model has been implemented. In order to attain this, 80% of total data has been trained and fed into the model.

Note that the features for profit in this model are ‘”Staff”, ”Competition Score”, ”Floor Space”, ”Car park” and ”Location”.

The results are shown below:

R^2: 53.57669497460548
Root Mean Squared Error for Test: 3047347.300458938
Root Mean Squared Error for Train: 3072501.5644306787

MLPregressor with K-fold based on the hyperparameters has been gotten from GridCV
Compute 5-fold cross-validation scores: cv_scores
[38.0340912 ,25.19218153, 29.24107833 ,16.46304435 ,14.73850082]
Average 5-Fold CV Score: 23.733779247102262

As results showed that the accuracy of the model is very low. However, the difference between errors for test and train set is low (25423.12) but it depicted (in lieu of machine learning and pursuant to the problem in question) that the Multi-layer perceptron is not a good model for our data set to predict profit due to the low accuracy.

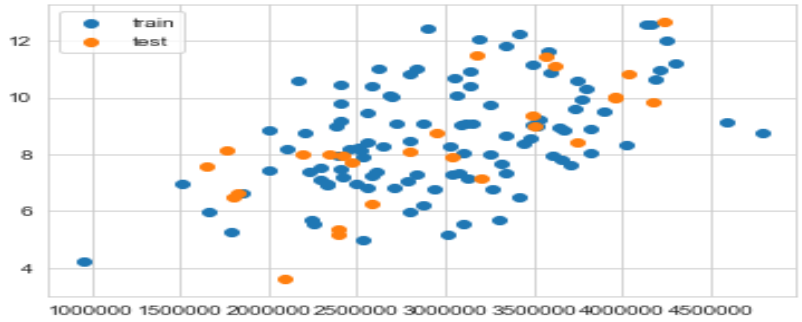


Figure (13): train and test actual vs predictions

5.2.4 Conclusion

To conclude with the best model, following table can been seen which illustrated that the best model with high accuracy and less RMS error is best fit for profit (in terms of hyper parameter i.e. Alpha) is Ridge regression:

Model	Hyper parameter	Accuracy (training80%)	Accuracy (cv=5)	Train RMS	Test RMS
Ridge Regression	Alpha=0.1	81%	68.26%	431993	338802.47
Lasso Regression	Alpha=0.1	85%	65.69%	427407.19	304145.1
Multi-layer perceptron	solver: Adams, Alpha: 0.0001, Hidden layer: 2	53.57%	23.13%	3072501.5	3047347.3

Table (5): the model result for prediction of profit

5.3 Performance Classification

In this segment tried to predict and classified where our target is performance, the target variable are as t he set of performance categories: '*Poor'*, '*Reasonable'*, '*Good'*, '*Excellent'*. To do so logistic regression , decision tree and multi perception models are used.

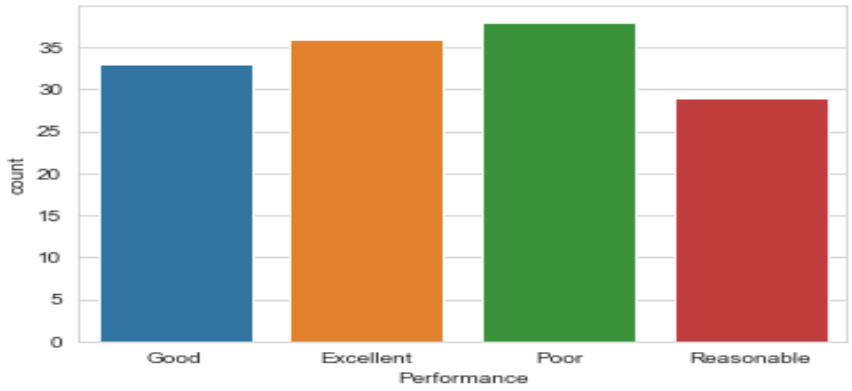


Figure (14): Performance distribution histogram

The first step before building model is selection the features which having more impact to the target. two metho d are used, correlation and random forest.From scikit-learn package Random Forest Classifier is imported to py thon environment.

Note: hyperparameter are "bootstrap", "criterion", "max_depth" like profit section

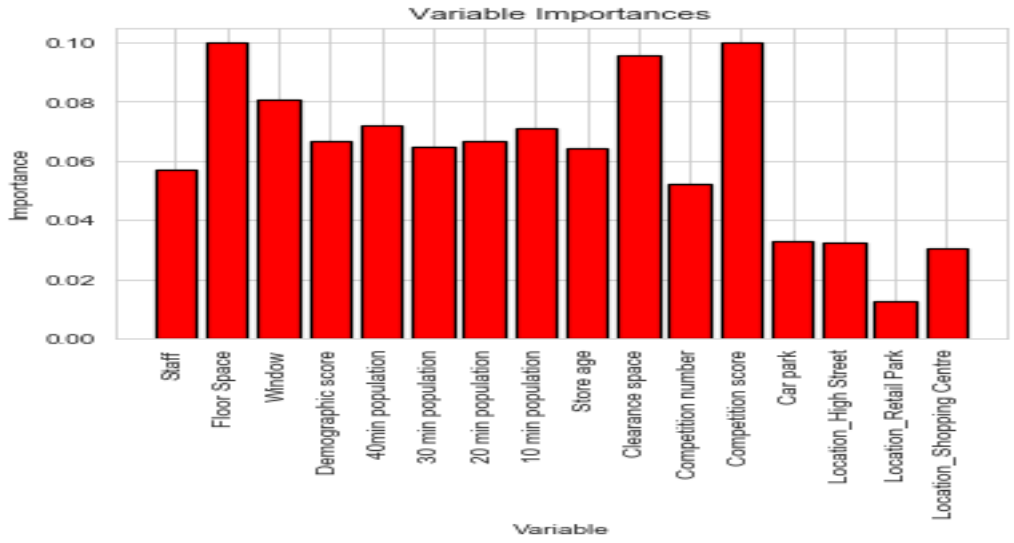


Figure (15): Random forest feature selection .The result regard random forest model shows that "Clearance space ", "Floor Space", "location", "Staff", " Demographic Score" ,"Location", "Competition Number", "Competition score " and '40 min population' are the most importance features(in favour of correlation table)

5.3.1 Logistic Regression

The first model for Performance classification is logistic regression. Despite its name, logistic regression is used in classification problem not regression problems. We have four possible labels for the target variable (performance).logistic regression will output a probability 'p'with respect to the target variable .To build model, from Scikit-learn, Logistic Regression imported to python environment. The next steps was finding hyper parameter to improve model performance and then split data into training and test set, fit model on training data ,and predict on test set. To evaluate the Logistic Regression, the hyperparameters considered for variation are:

- **Penalty:** Used to specify the norm used in the penalization. The 'newton-cg', 'sag' and 'ibfgs' solvers support only l2 penalties.(1)
- **Solver:** Algorithm to use in the optimization problem.(1)
- **C:** Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.(1)

Results:

- **Tuned hpyerparameters :(best parameters) {'C': 1.0, 'penalty': 'l1','solver': "lbfgs"}**

Accuracy: 0.59796992481203

- **Model accuracy for test and train :**

Training accuracy: 0.5660377358490566
Test accuracy : 0.6666666666666666

- **Test prediction:**
  array ([4, 4, 2, 1, 1, 4, 3, 2, 2, 4, 1, 1, 1, 2, 1, 4, 4, 4, 1, 3, 3, 1,
  2, 4, 4, 4, 4], dtype=int64)
- **train prediction:**
  Array ([3, 4, 4, 2, 1, 4, 1, 2, 1, 2, 3, 2, 1, 3, 2, 3, 3, 1, 3, 1, 2, 1,
  4, 2, 4, 1, 3, 4, 1, 4, 4, 1, 1, 2, 1, 1, 2, 1, 3, 2, 3, 3, 4, 3,
  4, 3, 3, 3, 4, 2, 1, 4, 3, 4, 4, 2, 2, 1, 3, 2, 3, 1, 2, 1, 4, 2,
  2, 4, 4, 3, 3, 1, 2, 1, 2, 2, 3, 4, 4, 4, 3, 4, 1, 3, 4, 1, 4, 3,
  1, 3, 1, 4, 3, 3, 2, 2, 1, 4, 4, 1, 1, 3, 4, 1, 4, 2], dtype=int64)

- **Cross validation :**

[0.66666667 0.33333333 0.6 0.35714286 0.53846154 0.61538462
0.53846154 0.58333333 0.5 0.45454545]
Average 10-Fold CV Score: 0.5187329337329337

In classifications model, we can use accuracy, the fraction of correctly classified sample to measure model performance. However, accuracy is not always a useful metric. For our model we can draw up a matrix that summarize predictive performance called a confusion matrix. The classification report and confusion matrix were imported to python from scikit-learn.metrics package and split data into train and test set .then fitted the model ,and predicted the labels of the test set. To compute the confusion matrix, I passed the test set labels and the predicted labels to the function confusion matrix.to compute the resulting metric, passed the same argument to classification report, which output a string containing all the relevant metrics, which you can see below:

```
[[10  0  0  1]
 [ 0  2  1  1]
 [ 0  3  6  0]
 [ 0  0  1  2]]
              precision    recall  f1-score   support

           1       1.00      0.91      0.95        11
           2       0.40      0.50      0.44         4
           3       0.75      0.67      0.71         9
           4       0.50      0.67      0.57         3

   micro avg       0.74      0.74      0.74        27
   macro avg       0.66      0.69      0.67        27
weighted avg       0.77      0.74      0.75        27
```

Precision mean the number of true positive divided by the total number of positive and false positive. In our case, this number is the correctly labeled for each performance four classes. As you can see, for instance it was 100% for the first class, 40% for the second class, 75% for the third class and 50 % for the fourth class. Recall, which the number of the true positives divided by the total number of true positives and false negatives (true positive rate).To put it all in plain language, high precision means that our classifiers predicted correctly .and high recall means our classifier predicted most positive (labels) correctly.

Notice that in defining logistic regression, we specified a threshold of 0.5 for the probability threshold that defines our model.to plot the roc curve, roc curve was imported to python environment(from scikit-learan.metric package),then called the function roc-curve; the firs argument is given by the actual labels, the second by the predicted probabilities . The larger the area under ROC curve, means the better model.

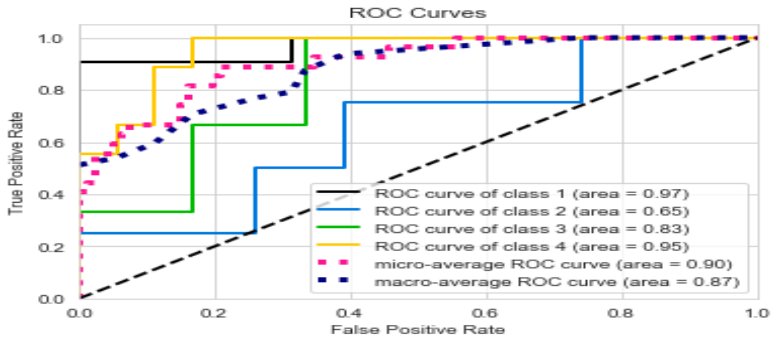


Figure (16): ROC Curve for performance 4 classes .As you can see the area under curve for class 4,3,2 and 1 means how good was our model for each classes("Excellent":1,"Good":2,"Poor":3,"Reasonable":4)

5.3.2 Decision Tree

The aim of this segment is to see how a classifications-tree learnt for our data. Like other modeling process, the hyper-parameters needed to be find:

- **Criterion:** ***string, optional (default="gini")*** the function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.
- **Max depth:** The maximum depth of the tree. If none, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

```
Best Score: 0.5037593984962406
Best params: {'criterion': 'entropy', 'max_depth': 7}
```

Then, set the hyper parameters to the model, fitted the model to the training set and predicted the test set labels. Then determined the test and train accuracy.

	precision	recall	f1-score	support
1	1.00	0.73	0.84	11
2	0.43	0.75	0.55	4
3	0.33	0.67	0.44	3
4	0.67	0.44	0.53	9
micro avg	0.63	0.63	0.63	27
macro avg	0.61	0.65	0.59	27
weighted avg	0.73	0.63	0.65	27

```
training accuracy: 0.8018867924528302
test accuracy    : 0.6296296296296297
```

The result shows that how well model performed when we trained it on 80% data, as portrayed class one ("excellent") has trained better than other classes .Also the accuracy of model for test and train are 62% and 80%.and the cross validation result is 50%

```
[0.39285714 0.42857143 0.62962963 0.38461538 0.625
Average 10-Fold CV Score: 0.4921347171347171
```

The next step was graphing the Decision tree of model to see how model performed:

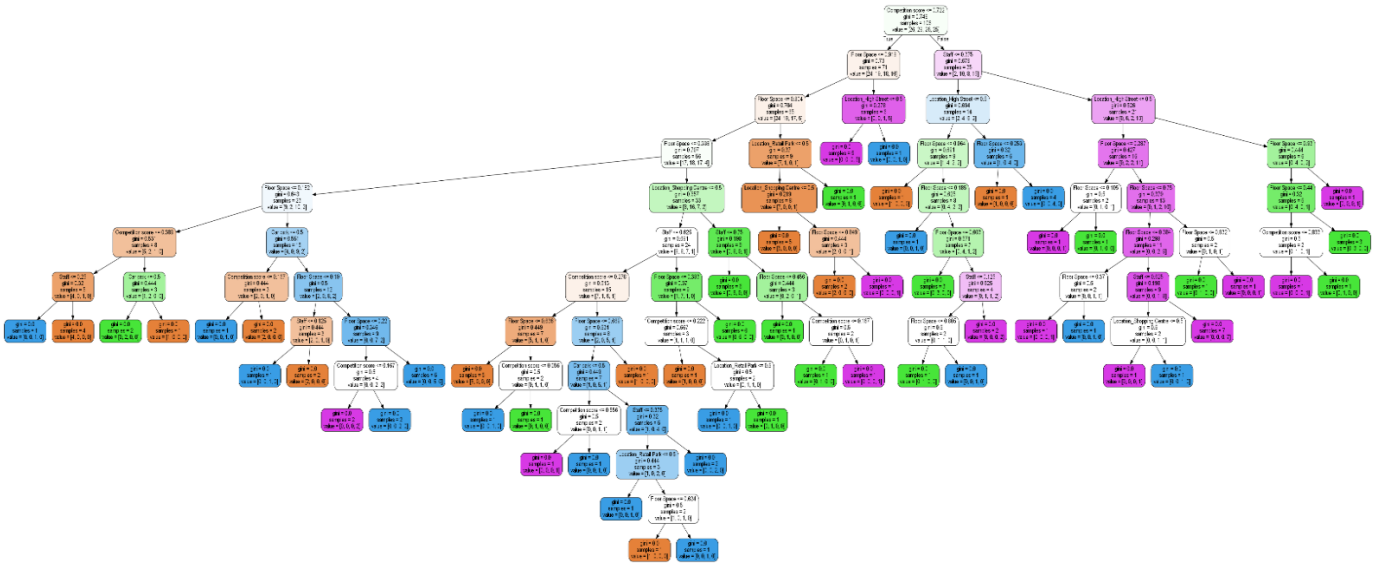


Figure (17): The top node here is competition score, and each color present each of the performance 4 classes.

5.3.3 Multi-layer perceptron :(MLPClassifier)

The aim of this segment is to classify performance classes in favour by impalement multi-layer perceptron method. For to do so, first imported MLPClassifier from scikitlean -neural network, second I found hyper parameter for the model. (Most important hyper parameter):

- **Hidden layer sizes:**
  The ith element represents the number of neurons in the ith-hidden layer (1).
- **Activation:**
  Activation function for the hidden layer (1)
- **Solver:**
  The solver for weight optimization .L2 penalty (regularization term) parameter (1)

```
Best parameters found:
 {'activation': 'tanh', 'alpha': 1, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'}
```

Iteration 895, loss = 0.44180346
Iteration 896, loss = 0.44164902
Iteration 897, loss = 0.44149953
Iteration 898, loss = 0.44133731
Test accuracy: 0.6666666666666666
Training accuracy : 0.69763321

Then after fitting model it time to check confusion matrix:

	precision	recall	f1-score	support
1	1.00	0.55	0.71	11
2	0.50	0.75	0.60	4
3	0.50	0.67	0.57	3
4	0.64	0.78	0.70	9
micro avg	0.67	0.67	0.67	27
macro avg	0.66	0.68	0.64	27
weighted avg	0.75	0.67	0.67	27

Compute 5-fold cross-validation scores:

[0.55714286 0.59285714 0.33333333 0.67692308 0.41666667]
Average 5-Fold CV Score: 0.58538461538461535

5.3.4 Conclusion

On the basis of relevant features selection, the result shows Multilayer Perceptron Classifier(MLP) is a best model. The justification has been laid on the ground s that it gives best accuracy (CV Accuracy=0.58) which is shown in the table below as compared to Logistic Regression and decision tree.

Model	Accuracy CV	Test Accuracy	Train Accuracy
Logistic Regression	0.52	0.66	0.57
Decision Tree	0.50	0.62	0.80
MLPClassifier	0.58	0.66	0.70

Table (6): The classification model result (target is performance)

6. Recommendations

Based on the results in this report, it should be noted that using machine learning technique, prediction models have been built to yield estimate for retail chain's profit/income/revenue. The models have built after taking a set of relevant features (extracted using feature selection algorithms) for the target i.e. profit. Similarly, classification models have been built using machine learning technique to estimate performance ['Good', 'Reasonable', 'Poor', 'Excellent'] for a shop with relevant feature selection. Since they statistical models, no guarantee can be made in terms of complete certainty for both prediction and classification because the performance of the model will mainly depend on the data on which they have been trained. Also, various external forces such as *PESTLE (Political, Economic, Socio-cultural, Technological, Legal and Environmental)* have high impact on the business, which should also be taken care of, for the survival of business in the long run.

After selecting the attributes which have high influence on both the targets i.e. PROFIT & PERFORMANCE, various models were compared to conclude with the best model. To attain *Cost Leadership advantage* in the retail industry (in terms of cost implications), it is suggested to consider identified set of attributes for future data collection. It will help in cost minimization which will further results in higher profit as well as better cash flow for the business.

A set of attributes suggested for both targets in terms of data collection for future period are illustrated below:

- **Profit:** ['Staff', 'Floor Space', 'Location', 'Car Park', 'Competition Score']
- **Performance:** ['Staff', 'Floor Space', 'Location', 'Car Park', 'Competition Score', 'Demographic Score', 'Clearance Space', '40 min population']

For Prediction task in question, the data has been trained and evaluated using Multivariate Regression and Multilayer Perceptron (MLP). Based on the findings, those were achieved using Python (Scikit Learn Package), Ridge Regression Model is suggested for commercial implementation because it gives higher accuracy with less error as compared to Lasso Regression and Multilayer Perceptron (MLP) for predicting the profit. Ridge Regression Model represented a low error on both train and test set which is equal to £385397.5 (average error) with 68.26%accuracy (on cross-fold). The result of this model is in alignment to the goals of a business enterprise i.e. the attained error on prediction is less than a million pounds.

For Classification task in question, the data has been trained and evaluated using Logistic Regression and Multilayer Perceptron (MLP) and decision tree. Based on the findings in this report which were arrived at after exploring these three models in Python (Scikit Learn Package) environment, Multilayer Perceptron (MLP) classifier is suggested for commercial implementation at World of Bargains because it gives high accuracy (based on cross-validation training and test evaluation) which is 58% as compared to Logistic Regression and Decision Tree.

Though the models used for prediction and classification task to achieve the business objectives have performed in an effective way, for further review following improvements have been suggested for better growth opportunities:

For Prediction and classification task in future, population size should also be considered as an important attribute to build a model because business survives when it has sufficient customers and for long term survival of the business, the area with high population size needs to be chosen as the overall sales will increase, economies of scale will be more and cost will be less. Furthermore, advertisement, branding and brand positioning is really important aspects to be taken care of. Also, features like Demographic space and Clearance space can be neglected because in reality they are less relevant in terms of impact on the business and by removing them, cost to acquire data in relation to them can be saved. In terms of assessing the business performance, 'Cash flow' is more relevant attribute than 'profit' as it is essential for the smooth running of a business and efforts should be made to collect some data on it in order to predict the future cash flows which will give better insights on choosing areas of investment by the business enterprise.

References:

Azevedo, A.I.R.L. and Santos, M.F., 2008. KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*. http://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf

Brownlee, J. (2019). How to Calculate Precision, Recall, F1, and More for Deep Learning Models. [Blog] *Machine Learning Mastery*. https://machinelearningmastery.com/blog/

DataCamp Community. (2019). *Regularization: Ridge, Lasso and Elastic Net*. https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

Digitalocean.com. (2019). *How To Build a Machine Learning Classifier in Python with Scikit-learn | DigitalOcean*. https://www.digitalocean.com/community/tutorials/how-to-build-a-machine-learning-classifier-in-python-with-scikit-learn

Fayyad, U. and Stolorz, P., 1997. Data mining and KDD: Promise and challenges. *Future generation computer systems*, *13*(2-3), pp.99-115. https://www.sciencedirect.com/science/article/pii/S0167739X97000150

Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P., 1996, August. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *KDD* (Vol. 96, pp. 82-88). https://www.aaai.org/Papers/KDD/1996/KDD96-014.pdf

Feyyad, U.M., 1996. Data mining and knowledge discovery: Making sense out of data. *IEEE expert*, *11*(5), pp.20-25. https://ieeexplore.ieee.org/abstract/document/539013

Jain, A., Menon, M.N. and Chandra, S., 2015. Sales Forecasting for Retail Chains. https://pdfs.semanticscholar.org/76a2/44f4da1d29170a9f91d381a5e12dc7ad2c0f.pdf

Lemaître, G., Nogueira, F. and Aridas, C.K., 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, *18*(1), pp.559-563. http://www.jmlr.org/papers/volume18/16-365/16-365.pdf

Pal, S.K. and Mitra, S., 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on neural networks*, *3*(5), pp.683-697. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=159058

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), pp.2825-2830. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Scikit-learn.org. (2019). *1. Supervised learning — scikit-learn 0.20.3 documentation*. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Torgo, L., 1995. Data fitting with rule-based regression. In *Proceedings of the 2nd international workshop on Artificial Intelligence Techniques (AIT'95)*. Brno, Czech Republic. https://www.researchgate.net/profile/Luis_Torgo/publication/2378698_Data_Fitting_With_Rule-Based_Regression/links/0912f509af103b00b3000000.pdf

Weston, J. and Watkins, C., 1999, April. Support vector machines for multi-class pattern recognition. In *Esann* (Vol. 99, pp. 219-224). http://tka4.org/materials/lib/Articles-Books/Speech%20Recognition/from%20Nickolas/SVM%20for%20multi-class%20pattern%20recognition.pdf