***Customer Transaction Prediction***

Priyanka Kumari

(23/11/2019)

**Contents**

**1 Introduction**

1.1 Problem Statement

1.2 Data

**2 Methodology**

2.1 Pre Processing

2.1.1 Exploratory Data Analysis

2.1.2 Missing Value Analysis

2.1.3 Outlier Analysis

2.1.4 Feature Selection

2.1.5 Feature Scaling

2.2 Modeling

2.2.1 Model Selection

2.2.2 Multiple Linear Regression

2.2.3 Decision Trees

2.2.4 Random Forest

**3 Conclusion**

3.1 Model Evaluation

**1. INTRODUCTION**

**1.1 Problem statement**

In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

At Santander, mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals. Our data science team is continually challenging our machine learning algorithms, working with the global data science community to make sure we can more accurately identify new ways to solve our most common challenge, binary classification problems such as:

• is a customer satisfied?

• Will a customer buy this product?

• Can a customer pay this loan?

According to past data and from the given problem the output is Classification and it comes under Supervised Machine Learning . We train the model with past data and when the newdata is given we predict the outcome

**1.2 Data**

The goal is to build classification models which will predict the number of bikes used based on the environmental and season behaviour. Given below is a sample of the data set that we are using to predict the number of bikes:

Given data contains numeric feature variables, the binary target column, and a string ID\_code column. The task is to predict the value of target column in the test set..

* ID\_code (string);
* Target;
* 200 numerical variables, named from var\_0 to var\_199;
* It has 201 predictors or independent variables and 1 target variable ‘target’

ID\_code object

target int64

var\_0 float64

var\_1 float64

var\_2 float64

var\_3 float64

var\_4 float64

var\_5 float64

var\_6 float64

var\_7 float64

var\_8 float64

var\_9 float64

var\_10 float64

var\_11 float64

var\_12 float64

var\_13 float64

var\_14 float64

var\_15 float64

var\_16 float64

var\_17 float64

var\_18 float64

var\_19 float64

var\_20 float64

var\_21 float64

var\_22 float64

var\_23 float64

var\_24 float64

var\_25 float64

var\_26 float64

var\_27 float64

...

var\_170 float64

var\_171 float64

var\_172 float64

var\_173 float64

var\_174 float64

var\_175 float64

var\_176 float64

var\_177 float64

var\_178 float64

var\_179 float64

var\_180 float64

var\_181 float64

var\_182 float64

var\_183 float64

var\_184 float64

var\_185 float64

var\_186 float64

var\_187 float64

var\_188 float64

var\_189 float64

var\_190 float64

var\_191 float64

var\_192 float64

var\_193 float64

var\_194 float64

var\_195 float64

var\_196 float64

var\_197 float64

var\_198 float64

var\_199 float64

Length: 202, dtype: object

**2. Methodology**

**2.1 Pre Processing**

Any predictive modelling requires that we look at the data before we start modelling. However, in data mining terms looking at data refers to so much more than just looking.Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

**2.1.1 Exploratory Data Analysis**

In exploring the data we have

 Omitted ID\_code and Target variable from train data

**2.1.2 Missing Value Analysis**

Missing value analysis is done to check is there any missing value present in given dataset. Missing values can be easily treated using various methods like mean, median method, knn method to impute missing value.

In R function(x){sum(is.na(x))} is the function used to check the sum of missing values. In python bike\_train.isnull().sum() is used to detect any missing value

| **Variables** | **Missing\_percentage** |
| --- | --- |
| **0** | ID\_code | 0.0 |
| **1** | var\_136 | 0.0 |
| **2** | var\_126 | 0.0 |
| **3** | var\_127 | 0.0 |
| **4** | var\_128 | 0.0 |
| **5** | var\_129 | 0.0 |
| **6** | var\_130 | 0.0 |
| **7** | var\_131 | 0.0 |
| **8** | var\_132 | 0.0 |
| **9** | var\_133 | 0.0 |
| **10** | var\_134 | 0.0 |
| **11** | var\_135 | 0.0 |
| **12** | var\_137 | 0.0 |
| **13** | var\_149 | 0.0 |
| **14** | var\_138 | 0.0 |
| **15** | var\_139 | 0.0 |
| **16** | var\_140 | 0.0 |
| **17** | var\_141 | 0.0 |
| **18** | var\_142 | 0.0 |
| **19** | var\_143 | 0.0 |
| **20** | var\_144 | 0.0 |
| **21** | var\_145 | 0.0 |
| **22** | var\_146 | 0.0 |
| **23** | var\_147 | 0.0 |
| **24** | var\_125 | 0.0 |
| **25** | var\_124 | 0.0 |
| **26** | var\_123 | 0.0 |
| **27** | var\_122 | 0.0 |
| **28** | var\_101 | 0.0 |
| **29** | var\_102 | 0.0 |
| **...** | ... | ... |
| **172** | var\_93 | 0.0 |
| **173** | var\_94 | 0.0 |
| **174** | var\_95 | 0.0 |
| **175** | var\_96 | 0.0 |
| **176** | var\_75 | 0.0 |
| **177** | var\_73 | 0.0 |
| **178** | var\_50 | 0.0 |
| **179** | var\_72 | 0.0 |
| **180** | var\_51 | 0.0 |
| **181** | var\_52 | 0.0 |
| **182** | var\_53 | 0.0 |
| **183** | var\_54 | 0.0 |
| **184** | var\_55 | 0.0 |
| **185** | var\_56 | 0.0 |
| **186** | var\_57 | 0.0 |
| **187** | var\_58 | 0.0 |
| **188** | var\_59 | 0.0 |
| **189** | var\_60 | 0.0 |
| **190** | var\_61 | 0.0 |
| **191** | var\_62 | 0.0 |
| **192** | var\_63 | 0.0 |
| **193** | var\_64 | 0.0 |
| **194** | var\_65 | 0.0 |
| **195** | var\_66 | 0.0 |
| **196** | var\_67 | 0.0 |
| **197** | var\_68 | 0.0 |
| **198** | var\_69 | 0.0 |
| **199** | var\_70 | 0.0 |
| **200** | var\_71 | 0.0 |
| **201** | var\_199 | 0.0 |

202 rows × 2 columns

There is no missing value found in given dataset.

**2.1.3 Outlier Analysis**

In statistics, an outlier is defined as a data point that differs significantly from other observations. Outlier analysis is a technique to find these points. Causes of Outliers

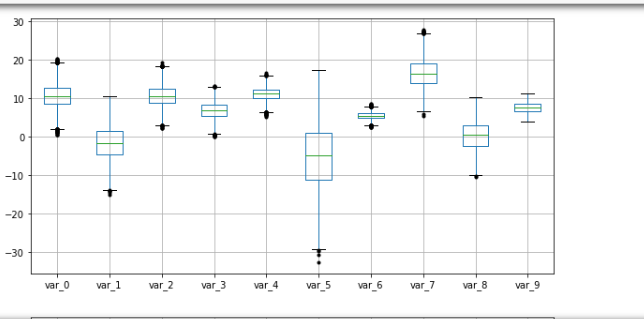
• Poor data quality/contamination

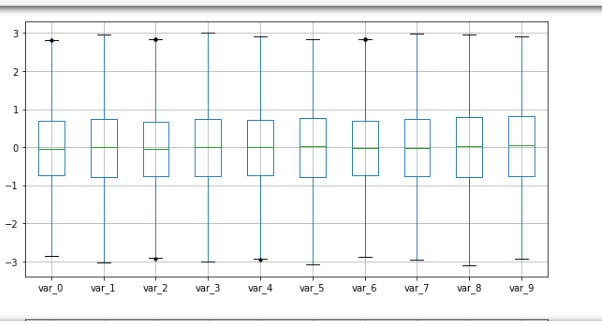
• Low-quality measurements, malfunctioning equipment, manual error

• Correct but exceptional data

One of the steps of pre-processing apart from checking the normality is the presence of outliers. We visualize the outliers using box plot.

Outlier analysis is done to handle all inconsistent observations present in given dataset. As outlier analysis can only be done on continuous variable. Below figure are visualization of numeric variable present in our dataset to detect outliers using boxplot. Outliers will be detected with black color .





According to above visualizations there is few outliers found in variables.

**2.1.4 Feature Selection**

Before performing any type of modelling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. This process of selecting a subset of relevant features/variables is known as feature selection. There are several methods of doing feature selection. I have used correlation analysis. In our dataset, the correlation between the train attributes is very small. So, there is no need to remove variables

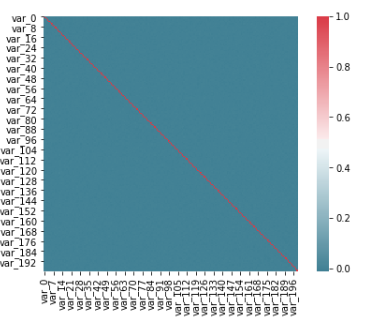


Fig:Correlation Plot

In our dataset, the correlation between the train attributes is very small.So, there is no need to remove variables

**2.1.5 Feature Scaling**

Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

# #Standarisation

for i in num\_train:

print(i)

df\_train[i] = (df\_train[i] - df\_train[i].mean())/df\_train[i].std()

**2.2 Modeling**

**2.2.1 Model Selection**

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

 *Logistic Regression*

 Decision Tree

 Random Forest

**2.2.2 Logistic Regression**

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Confusion Matrix:

[[31165 419]

[ 2526 905]]

Accuracy: 0.915893

precision: [0.92502449 0.68353474]

recall: [0.98673379 0.2637715 ]

fscore: [0.95488319 0.38065195]

**2.2.3 Decision Tree**

Decision tree builds classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

Accuracy score 0.902013

Confusion Matrix:

[[31584 0]

[ 3431 0]]

precision: [0.90201342 0. ]

recall: [1. 0.]

fscore: [0.94848271 0. ]

**2.2.4 Random Forest**

For Random Forest also we have divided train data into 80% train and 20% test datasets. Let’s look at the random forest model development code in python.

Random forests are based on a simple idea: 'the wisdom of the crowd'. Aggregate of the results of multiple predictors gives a better prediction than the best individual predictor. A group of predictors is called an ensemble. Thus, this technique is called Ensemble Learning. To improve our technique, we can train a group of Decision Tree classifiers, each on adifferent random subset of the train set. To make a prediction, we just obtain the predictions of all individuals trees, then predict the class that gets the most votes. This technique is called Random Forest. Random forest chooses a random subset of features and builds many Decision Trees. The model averages out all the predictions of the Decisions trees.

Accuracy score 0.902385

[[31583 1]

[ 3417 14]]

precision: [0.90237143 0.93333333]

recall: [0.99996834 0.00408044]

fscore: [0.94866635 0.00812536]

**3. Conclusion**

**3.1 Model Evaluation**

Now that we have three models for predicting the *Customer Transaction Prediction*, we need to decide which one to choose. There are several criteria that are used for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance

2. Interpretability

3. Computational Efficiency

In our case, we have used predictive performance criteria to select the best model. That means model which gives the best accuracy we will select that model.

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables and calculating some error metrics.

**3.2 Model Selection**

As we can see from the above tables the Logistic Regression gave the best accuracy both in R and python. That’s why we selected the Random Forest model for predicting the count.