**Data Analytics Final Project Report**

Fall 2022

**Dataset:**

taxi trips in New York City

**Target Variable:**

Extra (Categorical)

|  |  |
| --- | --- |
| Author: Niloufar Shokri  Date: 2022/12/10 | Prof. Adalbert F.X. Wilhelm |

**Table of Content**

Contents

[1 Introduction 4](#_Toc121407475)

[2 Data Set 4](#_Toc121407476)

[2.1 Taxi trips in New York City 4](#_Toc121407477)

[3 Data Pre-Processing 6](#_Toc121407478)

[3.1 Data Cleaning 6](#_Toc121407479)

[3.1.1 Checking for missing values 6](#_Toc121407480)

[3.1.2 Checking for garbage values 6](#_Toc121407481)

[3.1.3 Datetime variables 7](#_Toc121407482)

[3.1.4 Dropping unnecessary columns and rows: 7](#_Toc121407483)

[3.1 Data Cleaning 7](#_Toc121407484)

[3.1.1 Checking the distribution of each variable 7](#_Toc121407485)

[3.2 Data Transformation 8](#_Toc121407486)

[3.2.1 Dummy variables 8](#_Toc121407487)

[3.2.2 Normalization, standardization, scaling 8](#_Toc121407488)

[4 Data Exploration 8](#_Toc121407489)

[4.1 Finding outliers 8](#_Toc121407490)

[4.2 Correlation between different features: 8](#_Toc121407491)

[5 Data Analysis 8](#_Toc121407492)

[6 Results and Conclusions 8](#_Toc121407493)

[7 References 8](#_Toc121407494)

# Introduction

The data used in the attached datasets were collected and provided to the NYC Taxi and Limousine Commission (TLC) by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs (TPEP/LPEP).

The general task of my project is to build a predictive model for the target variable “**Extra**” which is a categorical variable. Therefore, the model is the classification type.

After loading the data and understanding each variable, I cleaned the data and performed simple data exploration using visualization,

# Data Set

## Taxi trips in New York City

The dataset is provided in ‘csv’ type. After importing the data set (with read\_csv() from Pandas library), the first step is getting some information about shape of the data set and variables.

The shape of the dataset is (28454, 23) which essentially means that there are 28454 rows and 23 columns in the dataset.

The features of the raw data set with their corresponding descriptions are as below:

| **Features** | **Description** | **Type** |
| --- | --- | --- |
| Unnamed: 0 | Index of the dataset | int64 |
| VendorID | A code indicating the TPEP provider that provided the record which is integer value of 1 for Creative Mobile Technologies and 2 for VeriFone Inc. | int64 |
| tpep\_pickup\_datetime | The date and time when the meter was engaged. | object |
| tpep\_dropoff\_datetime | The date and time when the meter was disengaged. | object |
| passenger\_count | The number of passengers in the vehicle. This is a driver-entered value. | int64 |
| trip\_distance | The elapsed trip distance in miles reported by the taximeter. | float64 |
| pickup\_longitude | Longitude where the meter was engaged. | float64 |
| pickup\_latitude | Latitude where the meter was engaged. | float64 |
| RatecodeID | The final rate code in effect at the end of the trip which is integer values of 1 to 5 where 1 is for Standard rate, 2 is for JFK, 3 is for Newark, 4 Nassau or Westchester and 5 is for Negotiated fare. | int64 |
| store\_and\_fwd\_flag | This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” as the vehicle did not have a connection to the server.  Y= store and forward trip  N= not a store and forward trip | object |
| dropoff\_longitude | Longitude where the meter was disengaged. | float64 |
| dropoff\_latitude | Latitude where the meter was disengaged. | float64 |
| payment\_type | A numeric code (integers from 1 to 4) signifying how the passenger paid for the trip where 1 is for Credit card, 2 is for Cash, 3 is for No charge and 4 for Dispute | int64 |
| fare\_amount | The time-and-distance fare calculated by the meter | float64 |
| extra | Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges. | float64 |
| mta\_tax | 0.50 MTA tax that is automatically triggered based on the metered rate in use. | float64 |
| tip\_amount | This field is automatically populated for credit card tips. Cash tips are not included. | float64 |
| tolls\_amount | Total amount of all tolls paid in trip. | float64 |
| improvement\_surcharge | 0.30 improvement surcharge assessed trips at the flag drop. the improvement surcharge began being levied in 2015. | float64 |
| total\_amount | The total amount charged to passengers. Does not include cash tips. | float64 |
| GoodTip | Categorical variable indicating an above average tip | bool |
| Extra | An indicator for additional charges included. | bool |
| Cash | An indicator whether payment was made by cash or not | bool |

# Data Pre-Processing

## Data Cleaning

The steps followed for the data set is given below:

### Checking for missing values

In most of the cases, we do not get complete datasets. They either have some values missing from the rows and columns or they do not have standardized values.

So, before going ahead with the analysis, it is a good idea to check whether the dataset has any missing values. Fortunately, the data set didn’t have any missing values.

### Checking for garbage values

Garbage value is generally a term meaning that the value in a variable doesn't have some sort of planned meaning.

By checking the statistical information of the data, some variables have negative values and some have 0 values which are not compatible with the definition (corresponding to the dataset).

The detail of these values is given in the following tables:

| **Variable with negative value** | **Description** |
| --- | --- |
| fare\_amount | -3.5, 3 times / -4.5, 2 times / -3.0, 2 times / -2.5, 2 times / -60.0, 1 time |
| extra | -0.5, 1 time / -1.0, 1 time |
| mta\_tax | -0.5, 10 times |
| improvement\_surcharge | -0.3, 10 times |
| total\_amount | -3.5, 3 times / -4.5, 2 times / -3.0, 2 times / -2.5, 2 times / -60.0, 1 time |

| **Variable with 0 value** | **Description** |
| --- | --- |
| passenger\_count | 1 time |
| trip\_distance | 154 times |
| fare\_amount | 6 times |
| pickup\_latitude and pickup\_longitude | 477 times |
| dropoff\_latitude and dropoff\_longitude | 444 times |

The pickup longitude and latitude and dropoff longitude and latitude equal to 0 belong to Null Island which is in international waters in the Atlantic Ocean. Therefore, these values should be filtered as well.

|  |  |
| --- | --- |
| Chart, histogram  Description automatically generated | Chart, histogram  Description automatically generated |

By removing the above-mentioned garbage values with filtering, the shape of our data set changes to (27755, 23).

### Datetime variable

In the data set, we have two variables named “tpep\_pickup\_datetime” and “tpep\_dropoff\_datetime” which are date time variables, but their format is string. Therefore, I converted them into datetime format and created 2 new variables, “duration\_permin” which is trip duration per minute and another one is the weekday.

By checking the statistical description of “duration\_permin”, there are 2 observations with 0 duration. So, I removed them as they are considered as garbage values, and the number of observations reduced into 27753.

### Dropping unnecessary columns

In this phase, first I checked the numeric variables with zero variance (threshold = 0), they do not have any contribution on the model. The near to zero variance variables are as follows:

* mta\_tax
* improvement\_surcharge

They can be removed from the data set. The variable “Unnamed: 0” is the dataset index which can be removed as well.

## Data Transformation

### Changing the categorical variables

There are 4 variables in the data set, one is “store\_and\_fwd\_flag” which its data type is string of Y or N, and three are “GoodTip”, “Extra” and “Cash” with the data type of Boolean. In order to use these variables in the model, I transformed all of them into 0 and 1.

# Data Exploration

## Correlation between different features

Correlation is the way of understanding the strength of the relationship between 2 variables or features in a dataset. Correlation coefficients determine this strength by indicating a value between [-1,1] where -1 indicates a very strong negative relationship, 0 indicates no relationship and 1 indicates strong positive relationship. Pearson correlation is one of the most widely used correlation method and it indicates the linear relationship between 2 variables. The heatmap of correlation between all variables of the dataset is given bellow:

Chart, treemap chart

Description automatically generated

The sorted correlation matrix for the target variable is as follows:

|  |  |
| --- | --- |
| **Variable** | **Correlation with Extra** |
| extra | 0.788995 |
| GoodTip | 0.055842 |
| dropoff\_longitude | 0.033074 |
| tip\_amount | 0.010252 |
| trip\_distance | 0.009459 |
| total\_amount | 0.008006 |
| pickday | 0.006360 |
| VendorID | -0.002636 |
| passenger\_count | -0.002936 |
| duraion\_permin | -0.006169 |
| store\_and\_fwd\_flag | -0.007480 |
| fare\_amount | -0.022386 |
| payment\_type | -0.032448 |
| Cash | -0.037907 |
| dropoff\_latitude | -0.041754 |
| tolls\_amount | -0.044502 |
| pickup\_longitude | -0.061198 |
| pickup\_latitude | -0.064492 |
| RatecodeID | -0.081118 |

Regarding the heatmap and our correlation matrix, our target variable “Extra” has the strongest linear correlation with “extra”. We look at this correlation closer by visualizing these 2 variables.

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

The above bar and box plot show that “Extra” is driven from “extra” regarding the rule that if extra = 0, Extra = False, if extra>0, Extra = True.

To check this more precisely, I chose all the numerical variables as my predicters, split my data set into train (80% of the observation) and test (20% of the observation), fitted a RandomForestClassifier model on train data and did a prediction on my test data. The classification report is as follows, the model with 100% accuracy and roc\_auc of 100%.

A screenshot of a computer

Description automatically generated with low confidence

Chart, line chart, scatter chart

Description automatically generated

If I remove “extra” variable from the model, the changes in model output are as follows:

Table

Description automatically generated

Chart, line chart

Description automatically generated

In order to have a correct model, I will remove “extra” from the model.

## Checking the distribution of each variable and Finding outliers

In this part of report, there are some visualizations to understand the distribution of the variables and to check if they have outliers or not.

### Continues Variables

For each continuous variable, I use boxplot and histogram for visualization. The plots are as bellows:

**duration\_permin** is right skewed. There are 144 observations which their duration\_permin is more than an hour. The outliers are larger than median.

Chart

Description automatically generated

**trip\_distance** is right skewed. There are 123 observations which their trip\_distance is very larger than 20 miles. The outliers are larger than median.

**Chart

Description automatically generated**

**pickup\_longitude** is not skewed but there are still some outliers in both sides of the median.

**Chart

Description automatically generated**

**pickup\_latitude** is not skewed but there are still some outliers in both sides of the median.

Chart, histogram

Description automatically generated

**dropoff \_latitude** is a bit right skewed but there are some outliers in both sides of the median.

Chart

Description automatically generated

**dropoff\_latitude** is not skewed but there are still some outliers in both sides of the median.

Chart

Description automatically generated

**fare\_amount** is right skewed. There are 90 observations which their fare\_amount is very larger than 52 dollors. The outliers are larger than median.

Chart

Description automatically generated

**extra** variable has 4 values, but there are 85 observations that have the value of 4.5.

Graphical user interface, chart

Description automatically generated

**tip\_amount** is right skewed. There are 40 observations which their tip\_amount is larger than 15 dollors. The outliers are larger than median.

Chart

Description automatically generated

**tolls\_amount** is right skewed. There are 24 observations which their tolls\_amount is larger than 5.54 dollors. The outliers are larger than median.

Chart, histogram

Description automatically generated

**total\_amount** is right skewed. There are a few observations which their total\_amount is larger than 50 dollors. The outliers are larger than median.

Chart

Description automatically generated

### Discrete Variables

For each discrete variable (including Booleans), I used bar charts highlighting the target variable for each value. The plots are as bellows:

|  |  |
| --- | --- |
| **VendorID**  1: 12793 observations  2: 14960 observations  The numbers of observations for 2 values of VendorID is almost balanced. | Chart, bar chart  Description automatically generated |
| **passenger\_count**  1: 19751 observations  2: 3954 observations  3: 1136 observations  4: 504 observations  5: 1471 observations  6: 937 observations  The numbers of observations for 6 values of passenger\_count is imbalanced. Most of the observations had 1 passenger. |  |
| **RatecodeID**  1: 27223 observations  2: 506 observations  4: 10 observations  5:14 observations  The numbers of observations for 4 values of RatecodeID is imbalanced. Most of the observations had RatecodeID of 1 and there was not any RatecodeID for Newark. |  |
| **payment\_type**  Credit card: 18770 observations  Cash: 8878 observations  No charge: 68 observations  Dispute: 37 observations  The numbers of observations for 4 values of payment\_type is imbalanced. Most of the observations had payment type of credit card or cash. |  |
| **store\_and\_fwd\_flag**  No: 27607 observations  Yes: 146 observations  The numbers of observations for 2 values of store\_and\_fwd\_flag is highly imbalanced. Almost all of the observations had store\_and\_fwd\_flag of No. |  |
| **pickday**  Monday: 4127 observations  Tuesday: 4283 observations  Wednesday: 4288 observations  Thursday: 3675 observations  Friday: 3952 observations  Saturday: 3566 observations  Sunday: 3862 observations  The numbers of observations for 7 values of pickday is almost balanced. |  |
| **Cash**  False: 18875 observations  True: 8878 observations  The numbers of observations for True value are almost half of False Value. |  |
| **GoodTip**  False: 16959 observations  True: 10794 observations  The numbers of observations for True value are almost two thirds of False Value. |  |

Regarding the above bar charts, the variables such as “RatecodeID” and “store\_and\_fwd\_flag” are near to zero variance variables and can be removed from our features of model.

For other variables, the variable “Extra” is almost balanced distributed in each value.

# Modeling

Now that the data is clean, the values of our target variable is balanced (True: 13402 , False: 14351) and we decided to select which features to put in the model, it is time to choose our classifier. A summary of each algorithm is described below.

**Logistic Regression** is a classification method used when the Response column is categorical with only two possible values. The probability of the possible outcomes is modeled with a logistic transformation as a weighted sum of the Predictor columns. The weights or regression coefficients are selected to maximize the likelihood of the observed data.

**Linear Discriminant Analysis** or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space. Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data

**K-Nearest Neighbors** algorithm, also known as KNN or k-NN, is a non-parametric algorithm (which means it does not make any assumption on underlying data), supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. a class label is assigned based on a majority vote.

**Decision Tree** is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization. The tree can be explained by two entities, namely decision nodes and leaves.

**Random Forest** is a collection (a.k.a. ensemble) of many decision trees. A decision tree is a flow chart which separates data based on some condition. If a condition is true, you move on a path otherwise, you move on to another path.

I selected 17 independent variables:

|  |  |
| --- | --- |
| VendorID | payment\_type |
| duraion\_permin | fare\_amount |
| passenger\_count | tip\_amount |
| trip\_distance | tolls\_amount |
| pickup\_longitude | total\_amount |
| pickup\_latitude | pickday |
| RatecodeID | Cash |
| dropoff\_longitude | GoodTip |
| dropoff\_latitude |  |

## Model Selection

In order to select my classifier, I performed a 10-fold cross validation algorithm on the above-mentioned classification models and compared their accuracy (average of all 10 folds) of each model. The result of the cross validation is as follows:

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Model Accuracy**  **(average of 10 folds)** | **Standard Deviation**  **(of 10 folds)** |
| Logistic Regression | 0.999685 | 0.000288 |
| Linear Discriminant Analysis | 0.996802 | 0.000792 |
| K-Nearest Neighbors | 0.724124 | 0.010515 |
| Decision Tree | 0.760876 | 0.021105 |
| Random Forest | 0.807765 | 0.005760 |

Regarding the accuracy score, I chose “**Logistic Regression**” as my classifier.

# Results and Conclusions

## Fitting the model

In this part of the report, I explain the steps I took for the algorithm:

**Step 1:** After deciding what variables to choose, I split my data into train (80% of observations) and test (20% of observations) dataset.

**Step 2:** I selected following hyper parameters for my model:

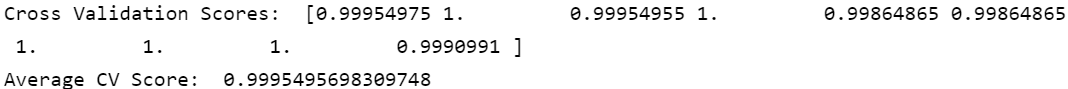
**Solver = 'lbfgs'**, Stands for Limited-memory Broyden–Fletcher–Goldfarb–Shanno. It approximates the second derivative matrix updates with gradient evaluations. It stores only the last few updates, so it saves memory. Algorithm to use in the optimization problem. **'lbfgs'** is the default solver.

**C = 0.3**, Inverse of regularization strength which must be positive float. Smaller values specify stronger regularization.

**Penalty = l2**, l2 penalty function uses the sum of the squares of the parameters and Ridge Regression encourages this sum to be small. l1 penalty function uses the sum of the absolute values of the parameters and Lasso encourages this sum to be small.

**n\_jobs = -1**, Number of CPU cores used when parallelizing over classes. -1 means using all processors

**Step 3:** After fitting the model on my tarin data set, I used the 10-fold cross validation to evaluate the accuracy of my model. The result of cross validation is as follows:



**Step 4:** Classification metrics,Confusion matrix and classification report, ROC AUC

Based on the results of the model prediction on the test data, the confusion matrix and classification report are as follows:

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | Predicted as False | Predicted as True |
| Actual False | 2896 (TP) | 6 (FN) |
| Actual True | 37 (FP) | 2612 (TN) |

Table

Description automatically generated

The area under the curve of the model is 0.9919824627266599.

Chart, line chart, scatter chart

Description automatically generated

## Improving the model

Regarding the classification metrics, the model is performing very well. But let’s see if the performance can be improved or not.

L2 regularization adds an L2 penalty equal to the square of the magnitude of coefficients. By using L2, all coefficients are shrunk by the same factor, and none are eliminated. This is the method which is used in Ridge regression. As I used l2 penalty in my model, I removed the features with small coefficient to check the performance of the model.

|  |  |  |
| --- | --- | --- |
| **Row** | **Dependent Variable** | **Coefficient** |
| 1 | VendorID | -0.187343 |
| 2 | duraion\_permin | -0.000361 |
| 3 | passenger\_count | 0.049829 |
| 4 | trip\_distance | 1.621235 |
| 5 | pickup\_longitude | -0.360051 |
| 6 | pickup\_latitude | -0.280732 |
| 7 | RatecodeID | -2.03707 |
| 8 | dropoff\_longitude | 0.170097 |
| 9 | dropoff\_latitude | -0.258859 |
| 10 | payment\_type | 0.325254 |
| 11 | fare\_amount | -11.620426 |
| 12 | tip\_amount | -10.96205 |
| 13 | tolls\_amount | -12.620044 |
| 14 | total\_amount | 11.092745 |
| 15 | pickday | -0.389677 |
| 16 | Cash | -0.011924 |
| 17 | GoodTip | 1.952415 |

By looking at the coefficients of the model, I keep below model and check the model performance with new features.

|  |  |
| --- | --- |
| * fare\_amount * tip\_amount | * tolls\_amount * total\_amount |

I performed the 4 steps again and the result are as follow:

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | Predicted as False | Predicted as True |
| Actual False | 2902 (TP) | 0 (FN) |
| Actual True | 0 (FP) | 2649 (TN) |

Table

Description automatically generated

The area under the curve of the model is 1.

Chart, line chart, scatter chart

Description automatically generated

# References