# PROJECT REPORT

# EXPLORATION AND MODELLING USING THE NYC TAXI DATASET

Prepared by Sergey Polyarus

**2022**

# EXECUTIVE SUMMARY

Task: implementation of a model predicting the Cash variable value.

Data background: 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). No renewal information was presented.

Methods used:

- Exploratory data analysis based on correlation and visualization;
- Random forest modelling.

Results:

- The Cash variable is independent from geographic variables, days of week, pickup and drop-off hours and number of passengers;
- Random Forest is a suitable model to distinguish Cash values;
- To make the model meaningful, some of the variables that are presented in the dataset should be ignored;
- The strongest predictor for Cash variable is the difference between total_amount and fare_amount;
- Despite the effort applied, the model remains overfit and subject to further enhancement to deal with new data;
- Intuitive insight tells that using cash or not is a matter of personal attitude and random life circumstances, so a perfect model should somehow take personality of the passenger into account.

# DETAILED REPORT

## 1. Introduction

The data tells about features of taxi trips in New York City in February of 2016. The yellow taxi trip records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

The dataset includes 22 variables and initially 28 454 observations. Among the variables two represent date and time, 13 are numeric (12 of which are continuous and 1 is discrete), 7 are categorical (4 Boolean and 3 simple categorical). However, all the variables may ultimately be represented as numbers and used for analysis and modelling as such.

Diversity of numeric variables is notable. Four of them represent geographic latitude and longitude, one – number of passengers – is discrete and may assume only natural values. The rest of numeric variables are related to money count. In addition, variance across the numeric variables varies a lot: for instance, the mta_tax variable contains three unique values only while fare_amount contains over 150.

The description of the numeric data may be seen at fig. 1 (all the figures of any kind are presented in Annex 1 hereto). There we can see some abnormalities like negative values for money-related values, unrealistic geographic coordinates and distances (since we know all the data is collected in NYC and it concerns taxi, not jets or ships).

Representation of categorical variables is also of interest. Particularly, Boolean values are encoded differently (as True/False and Y/N) that has to be unified. As for the rest categorical variables, they are presented as integer numbers.

The data contains no NAs but has certain outliers, strange values and values with special meaning.

One-by-one variable description with definitions is presented in Annex 2 hereto. To avoid lack of clarity the values will be referred to by their literal names in the dataset.

The goal of the report preparation was to make a model predicting the value of Cash variable, that is – whether the payment of all types of costs (including tolls, surcharges, taxes etc.) was performed by cash.

Python code used for obtaining the report date is presented in Annex 3 hereto. Be advised that Python 3.10.2. was used.

## 2. Background of the Data

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). No renewal information was presented.

## 3. Data Preprocessing

Before starting some data exploration or modelling, the data was preprocessed. Namely:

- Were dropped:
    - a row where RatecodeID equals to 99 as 99 is not informative and does not correspond with particular location as it should;
    - a row where passenger_count equals to 0 as it was a row with mistakes: trip_distance was over 0, so there was a trip, but passenger_count was input incorrectly. A decision was made to sacrifice this single row;
    - rows where latitudes or longitudes equal as NYC is located approximately at 41N, 74W and no taxi reaches 0N or 0W from there;
    - rows with abnormally high trip_distance as no taxi may be used for travelling over 100 miles (the rows with trip_distance>100 overall looked erroneous);
- Money-related column values were:
    - changed to positive as no metrics like income or profit (that may be negative) are contained in the dataset;
    - fitted with supposed values. For example, improvement_surcharge variable may only equal to 0.3 or 0;
- store_and_fwd_flag values were converted into True/False notation;
- Datetime variables (both for pickup and drop off) were split into date and time, further into month (however, only isolated drop-offs month was different being equal to 3, the rest equal to 2), hour, minute, second. Thus new columns (month, weekday, hour, minute, second) for pickups and drop-offs were inserted;
- Weekday was computed from the date for both pickup and drop-off;
- Dummies for categorical variables (VendorID, RatecodeID, payment_type) were formed and inserted as columns. These variables could be used as they were but their numeric value should not be operated as such (e. g. payment_type value ‘4’ should not be considered numerically larger than value ‘2’ – it is just another category). The initial columns were not dropped for visualization purposes;

- Column order was changed and index was reset for more convenient operation.

Thus the dataset became more clean and then contained real-life values only. Moreover, all the variables could be further dealt with as numeric without any misinterpretation. By the end of preprocessing stage, 27946 observations and 43 columns remained in the dataframe.

Not included into preprocessing (at this stage or later):

- As no NAs were found, no drops of fill-ins took place in this respect;
- Since random forest was chosen as a future model type, no scaling (either standardization or normalization) was performed.

## 4. Data Exploration

Data exploration consists in performing initial investigation on data, discovering general patterns and forming initial assumptions to be used for model-building. A lot of patterns are sure to be hidden in the data, but only those related to the Cash variable were searched and analyzed.

4.1. The first step is a pair plot – attempting to plot each variable of the dataframe against every other variable. The result is presented in fig. 2.

Due to large number of observations and their various nature the pair plot is not very informative – yet we easily notice that the plots are either chaotic or monotonous. In other words, not a pair of variables shows wavy or other complex non-linear relation.

This leads us to the outcome that a correlation heatmap might be more insightful. At the same time we cannot state for sure if the correlation between variables is linear or not – the pair plot does not expressly exclude slightly curvilineal correlation. In such a way at this stage we decide to use Spearman correlation for the correlation heatmap.

4.2. The correlation heatmap is color representation of correlation between the variables. Looking at the colors, one can grasp how high the correlation is. It should be noted that the correlation matrix that is a base for the heatmap was computed for columns that could potentially act as predictors. This means that for purposes of making up the heatmap the raw datetime columns and month column were omitted for reason of improper data format and extremely low variance respectively.

In this particular case the heatmap may be used for:

- noticing unintuitive correlations to further explore them prior to model building;

- identifying collinearities of the variables that may be used for modelling in order to optimize model building;
- predictor selection for the model.

At fig. 3 the heatmap is presented. To read it better especially for the target value, fig. 4 duplicates the correlation values as numbers.

Significant observations based on the heatmap and the raw correlation values are:

- The Cash variable is weakly correlated with majority of the variables. It shows slightly positive or slightly negative correlation with nearly all of them;
- The Cash variable is strongly correlated with payment_type_2, tip_amount, total amount and GoodTip variables. Trip_distance and fare_amount are the last ones that seem to have significant correlation with the Cash variable;
- Payment_type_2, tip_amount, total amount and GoodTip variables strongly correlate with each other, and on top of that tip_amount and total_amount correlate strongly with trip_distance, fare_amount, tolls_amount;
- Correlation between trip_distance and Cash is surprisingly low and negative just like with the money-related variables.
- Geographic variables show stronger correlation with each other than with any other variables;
- Weekdays and hours for pickup and drop-off strongly correlate;
- Mta_tax strongly correlates with RateCodeID 3 and 5 that stand for Newark and negotiated fare and with improvement_surcharge.

The conclusions from the observations are:

- A lot of variables are correlated with each other due to their nature: total_amount is in fact computed using tolls_amount, tip_amount, mta_tax and fate_amount. These values actually measure different fractions of the same thing – amount of money paid to the driver by the client. The same may be said about tip_amount and GoodTip – it is intuitive that high tip_amount results in True GoodTip by definition. In same fashion – by definition – payment_type_2 equals Cash;
- A shorter summary of the Cash predictors would be that it depends on amount of money paid and trip_distance representing non-monetary feature of the trip. However, trip_distance is strongly correlated with the money related variables

which is also intuitive. Correlation coefficient sign for Cash-money correlation and Cash-distance correlation is the same;

- Collinearity does not itself influence accuracy of the classification model negatively, so we may first fit a model using all the variables, but there will be a room to enhance it for sure;
- We remain conscious that the correlation value is different from statistical significance of one value as a predictor for another. This is why low absolute values of correlation should not *per se* bar any predictors from being included into the model.

To sum up, the heatmap confirmed the assumptions that could be made even before it was plotted.

4.3. Plotting the Cash Variable against the Variables Correlated with It

4.3.1. Payment_type as put onto the Cash variable barplot may be seen at fig. 5. From the figure it is clear that the payment_type value clearly divides Cash into True and False. As payment_type values ‘1’ were dropped while getting dummies, the correlation is shown with payment_type_2 only. Here we also see luckily that True and False values are distributed in Cash variable approximately at 1:2 rate;

4.3.2. GoodTip as put onto the Cash variable barplot may be seen at fig. 6. Here by definition of GoodTip its True values indicate that Cash value will be False;

4.3.3. Categorical plot of tip_amount distributed by Cash values is presented at fig. 10. It is easy to spot that tip_amount always equals to 0 while Cash is True;

Outcome from the first three plots: the three variables in fact represent cheat paraphrase of the Cash variable. They will make a model 100% accurate which means pointless. A model based a given direct indication to the target variable value is worth nothing. So the payment_type_2, GoodTip and tip_amount are very likely to be excluded from any model to be designed.

4.3.4. Barplot for the Cash being True or False for each drop-off hour may be seen at fig. 7;

4.3.5. Barplot for the Cash being True or False for each pickup hour may be seen at fig. 8;

4.3.6. The same barplot for each value of passenger_count was deemed useless due to extremely low correlation coefficient between Cash and this numeric value. This means that passenger_count increase does not imply increase of probability of Cash value being True.

Outcome from the two barplots:

- The overall trends of client's desire to pay in cash across both drop-off hour and pickup hour is the same. This means: people almost equally tend to pay in cash or not at the certain hour regardless whether they are picked up or dropped at this hour;
- The above trend is sometimes smoothed between hours (e. g. notice the similarity between 15-17 range for pickup and 16-18 range for drop-off). This may be explained by the fact that the same people are counted for different columns in these plots because they spend some time in the trip;
- Proportion of True Cash observations during any hour is roughly identical and corresponds with overall 1:2 value proportion.

4.3.7. Barplot for the Cash being True or False for each pickup weekday be seen at fig. 9. No separate plot was made for drop-off weekdays as in overwhelming majority of observations pickupweekday equals to dropoffweekday. The insight from this plot is that people pay in cash more frequently at the weekend but cash payment number never actually exceeds 60% of non-cash payment number;

4.3.8. Categorical box plot of trip_distance distributed by Cash values is presented at fig. 11. Its scaled version to find out more about short-distance trips is shown at fig. 11a These plots tell that a greater trip distance corresponds with less desire to pay in cash, however, the difference is so tiny that it is impossible to base prediction of a unique case on trip distance only. This may mean that trip_distance might help the model to be designed identify complicated cases correctly but will not be sufficient to make good predictions.

General result of data exploration with regard to the project goal may be summarized as follows:

- There are variables that predict the target variable value overconfidently;
- Usage of these variables is pointless as their values not just predict but predetermine the target variable value;
- Amongst others money-related and trip_distance variables seem to be mighty predictors for the target variable.

These hypotheses should be tested practically while making up a model.

## 5. Modelling

The model to be implemented in this dataset is chosen to be random forest.

Logistic regression model would also be a good choice, but decision tree models have lesser bias. Higher variance of decision tree models will be combated by bagging. Finally, random forest models have a plenty of hyperparameters to be tuned. These three considerations lead to random forest model.

Some general points about modelling performed:

- Random forest models are based on bagging. Moreover, the interface for designing them provides for validation on out-of-bag observations. Being conscious of that, we split the dataset into training and test parts and assess the model performance for both of them. Test sample proportion was chosen equal to 0.25;
- Despite the results of data exploration have already provided for some clues for model implementation, the first models will include all the predictors that are duly preprocessed and will not be sophisticated in terms of hyperparameters. The first model is rather an illustration than a working system;
- Model tuning will be performed gradually to see the progress caused by a single hyperparameter changed;
- While tuning the model, the target score will be recall. The logic base for the research suggest that the thing the taxi companies wish to face the least is expecting card payment while the payment will be in cash. This approach may be different for other countries where cash payment is more common than card payment, however, the overall proportion of Cash values proves that card payment should be considered a default option. In other words, a taxi company normally expects card payment but is interested in accurate prediction of cash payments. This means we should minimize the false negative rate that is reflected in recall value.

### 5.1. Initial Model

The model includes all the preprocessed predictors and, as predicted, lacks usefulness. Results of its performance are presented at figures 12 and 12a. The predictors directly determine the target value, so the model performs perfectly on both training and test samples.

It is hard even to blame high variance of the models and suspect it is overfit because the nature of its perfect performance is literally cheating. We may see that the most influential features were those that tell the target value otherwise. This means that the tree constructor was eager to split the nodes by payment_type_2, tip_amount and GoodTip values because this separation led to almost pure nodes or even directly to the leaves.

Outcome of the initial model is: we should exclude the three cheat predictors from the model.

### 5.2. The First Meaningful Model

The model designed without these three predictors is the first one that actually *predicts* rather than *computes* the target value. Results of its performance are presented at figures 13 and 13a.

We see that the cheat predictors were way more important than those we have at hand now, but all the scores except for recall are satisfactory. Moreover, one should notice that the top-2 important features are strongly correlated with each other (as fare_amount forms a large part of total_amount).

Geographic variables have roughly the same importance. The importance rate among time-related variables seems illogical. Passenger_count, that is attractive due to its low correlation with any other variable, is closer to the bottom of the importance rate.

The outcome of the first model:

- The cheat predictors are not the only ones that let us make good predictions;
- We should proceed with the most important features and then add some other that are not correlated with them (i. e. contain some new information about the observation);
- We should seek for a better recall value.

### 5.3. The First Successful Model

The next model includes the top-3 important features of the previous one only. Results of its performance are presented at figures 14 and 14a.

The greatest achievement of the model is having a satisfactory recall. Alongside with this, we notice that relative importance of trip_distance variable has become much higher (5x versus 3x difference with the most important feature). This may be deemed as confirmation of intuitive hypothesis that the money-related variables are not alone that help predict the Cash value – trip_distance is a distinct value that may also be useful.

At the same time we remember that total_amount and fare_amount are mutually correlated and may be computed using each other. So introducing a new feature that combines these two variables would be beneficial. For the sake of interpretability, we should probably use the difference between total_amount and fare_amount because it stands for difference between purely time-distance related payment and total payment amount, i. e. how costly the trip is apart from the time it takes and the distance it lasts.

Yet another thing that might be useful is to add some unused predictor like passenger_count to see the effect.

## 5.4. The First Artificial Feature Based Model

This model is based on a value of (total_amount - fare_amount), trip_distance and passenger count. Results of its performance are presented at figures 15 and 15a.

As we can see, the first artificial feature importance is times more important than other two features. We may also try to combine trip_distance and the existing artificial value as trip_distance correlated with both features used to calculate the existing artificial value just as strongly as they correlated with each other. To avoid NAs due to division by zero we should multiply the trip_distance and amount_minus_fare variables to get a ‘amount_minus_fare above trip_distance’ variable.

Passenger_count feature appeared to be futile (which was predictable since it showed no correlation with the Cash value).

The overall performance of the model was improved. The ideas for further development are dropping the passenger_count predictor and making an ultimate artificial feature combining both money-related and distance-related variables.

## 5.5. The Rollback

The performance of the one artificial feature based model is shown at fig. 16. No feature importance graph is provided as there was the only predictor feature in the forest. All the performance core decreased that shows that combining the features confused the random forest.

However, the idea that including trip_distance and money-related variables to the model simultaneously is repetitive still seems based. As seen by correlation coefficients, dropping the pure trip_distance predictor should not be worse than dropping the pure fare_amount predictor.

Thus we come to the idea of making a model based on amount_minus_fare only.

## 5.6. Final Model

The model based on amount_minus_fare and its performance may be contemplated at fig. 17. The recall is almost perfect as the model almost perfectly predicts True values of Cash variable. This was achieved at cost of precision that has decreased dramatically but remains reasonable (over 0.8).

The model permits to know if the payment will be performed in cash for 93% of cases. It may sometimes err on the conservative side, but it is acceptable under assumptions stated before para. 5.1. above.

### 5.7. Further Recommendations

Since the recall score achieved is almost perfect, future tuning will not be performed. However, it is clear that the model is overfit despite all the methods used. This could have included increasing the number of trees, their depth etc., but the main issue remains to be lack of observations. Greater dataset could give a bit of confidence that the tendencies caught by the model are not random.

Put it otherwise, amount of money to be paid on top of time-distance tariff is not the thing that may predetermine cash or card payment with 93% probability. So if more data is supplied, some strategies to improve the model may be useful.

Among them we should mention:

- Tuning the maximum number of features in each tree. This number should be limited to avoid growing correlating trees. Functions that compute the best parameters usually assure that the best number is the largest one (the trees become more complex), but in fact a model with no limits is too computationally expensive and extinguishes limitation of features as one of the key ideas of random forest;
- Tuning the maximum number of trees (default setting used was 100);
- Tuning the maximum tree depth. Short trees are better interpretable and less computationally expensive (default setting used was to grow the tree till its leaves become pure, that is obviously more than one or two levels).

Plots proving that the performance of the model on the given data was close to perfect with default settings may be seen at figures 18-20.

## 6. Conclusion

As a result of the research:

- The Cash variable is surprisingly independent from geographic variables, days of week, pickup and drop-off hours and number of passengers;
- Random Forest is a suitable model to distinguish Cash values;
- To make the model meaningful, some of the variables that are presented in the dataset should be ignored;

- The strongest predictor for Cash variable is the difference between total_amount and fare_amount;
- Despite the effort applied, the model is subject to further enhancement to deal with new data;
- If new data is not added, the model performance cannot be enhanced to a significant extent within the framework of random forest classifier;
- Intuitive insight tells that using cash or not is a matter of personal attitude and random life circumstances, so a perfect model should somehow take personality of the passenger into account.

## 7. Bibliography

Gareth James, Daniela Witten , Trevor Hastie , Robert Tibshirani:  An Introduction to Statistical Learning with Applications in R.  Springer, 2013

Scikit Learn User Guide: https://scikit-learn.org/stable/user_guide.html

# ANNEX 1

Fig. 1 Description of the Numeric Variables

| | passenger_count | trip_distance | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude |
|---:|---:|---:|---:|---:|---:|---:|
| count | 28454.00000 | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 |
| mean | 1.64360 | 13.374979 | -72.754727 | 40.079531 | -72.837611 | 40.126010 |
| std | 1.30007 | 1772.032130 | 9.418807 | 5.188718 | 9.096966 | 5.011552 |
| min | 0.00000 | 0.000000 | -74.177383 | 0.000000 | -74.344337 | 0.000000 |
| 25% | 1.00000 | 1.000000 | -73.991722 | 40.736568 | -73.991234 | 40.734502 |
| 50% | 1.00000 | 1.680000 | -73.981335 | 40.753445 | -73.979668 | 40.753859 |
| 75% | 2.00000 | 3.060000 | -73.966524 | 40.767941 | -73.962257 | 40.769294 |
| max | 6.00000 | 298914.200000 | 0.000000 | 40.955898 | 0.000000 | 41.143021 |

|  | **fare_amount** | **extra** | **mta_tax** | **tip_amount** | **tolls_amount** | **improvement_surcharge** | **total_amount** |
|---|---|---|---|---|---|---|---|
| count | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 | 28454.000000 |
| mean | 17.879753 | 0.326986 | 0.498002 | 1.762671 | 0.285483 | 0.299726 | 21.052621 |
| std | 917.746560 | 0.450626 | 0.034350 | 2.348040 | 1.369447 | 0.011794 | 917.887853 |
| min | -52.000000 | -1.000000 | -0.500000 | 0.000000 | 0.000000 | -0.300000 | -52.800000 |
| 25% | 6.500000 | 0.000000 | 0.500000 | 0.000000 | 0.000000 | 0.300000 | 8.300000 |
| 50% | 9.000000 | 0.000000 | 0.500000 | 1.350000 | 0.000000 | 0.300000 | 11.750000 |
| 75% | 14.000000 | 0.500000 | 0.500000 | 2.350000 | 0.000000 | 0.300000 | 17.160000 |
| max | 154810.430000 | 20.550000 | 1.160000 | 54.000000 | 61.050000 | 0.300000 | 154832.140000 |

Fig. 2 Pair Plot of the Dataset

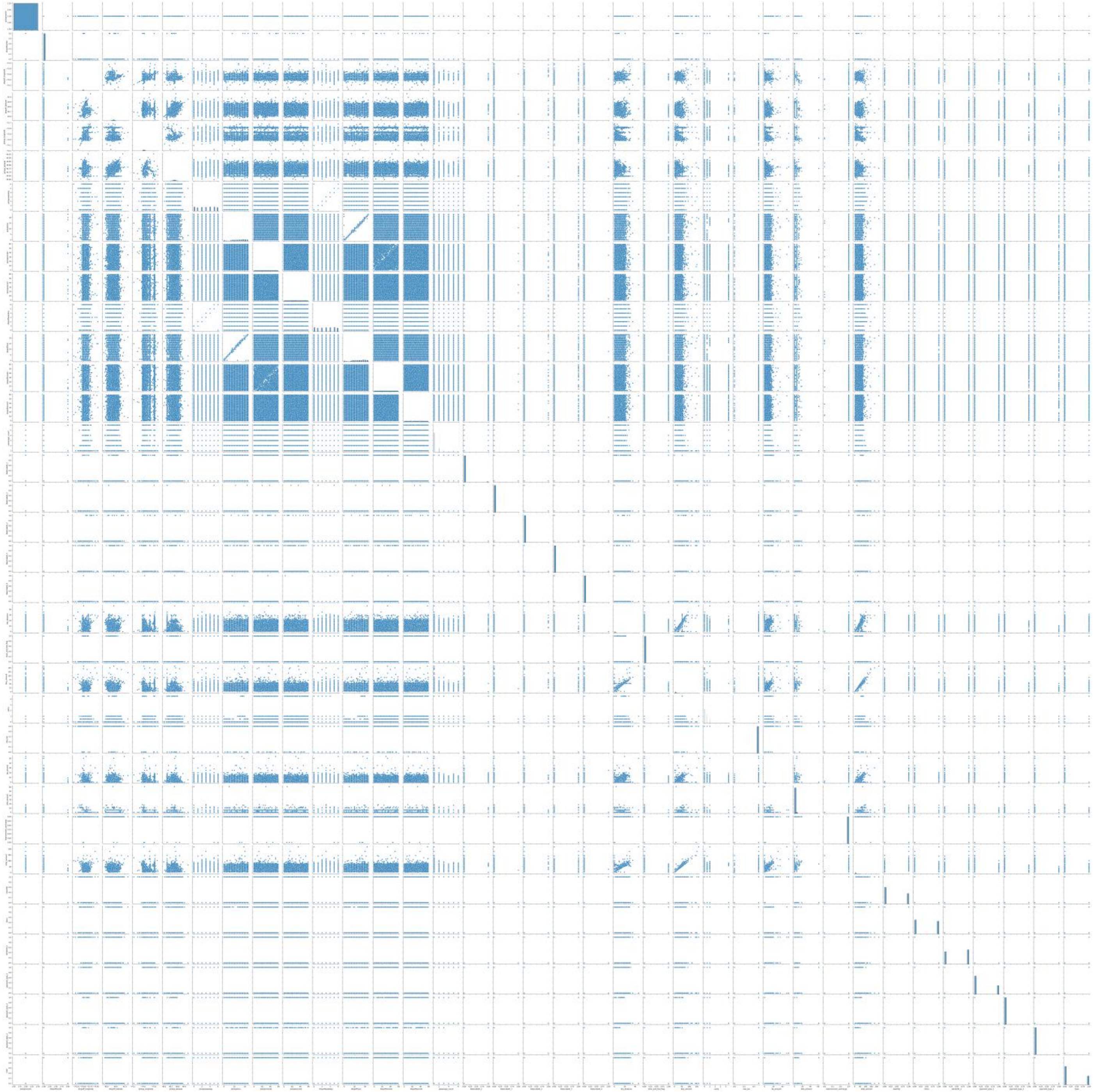


Fig. 3 Correlation Heatmap

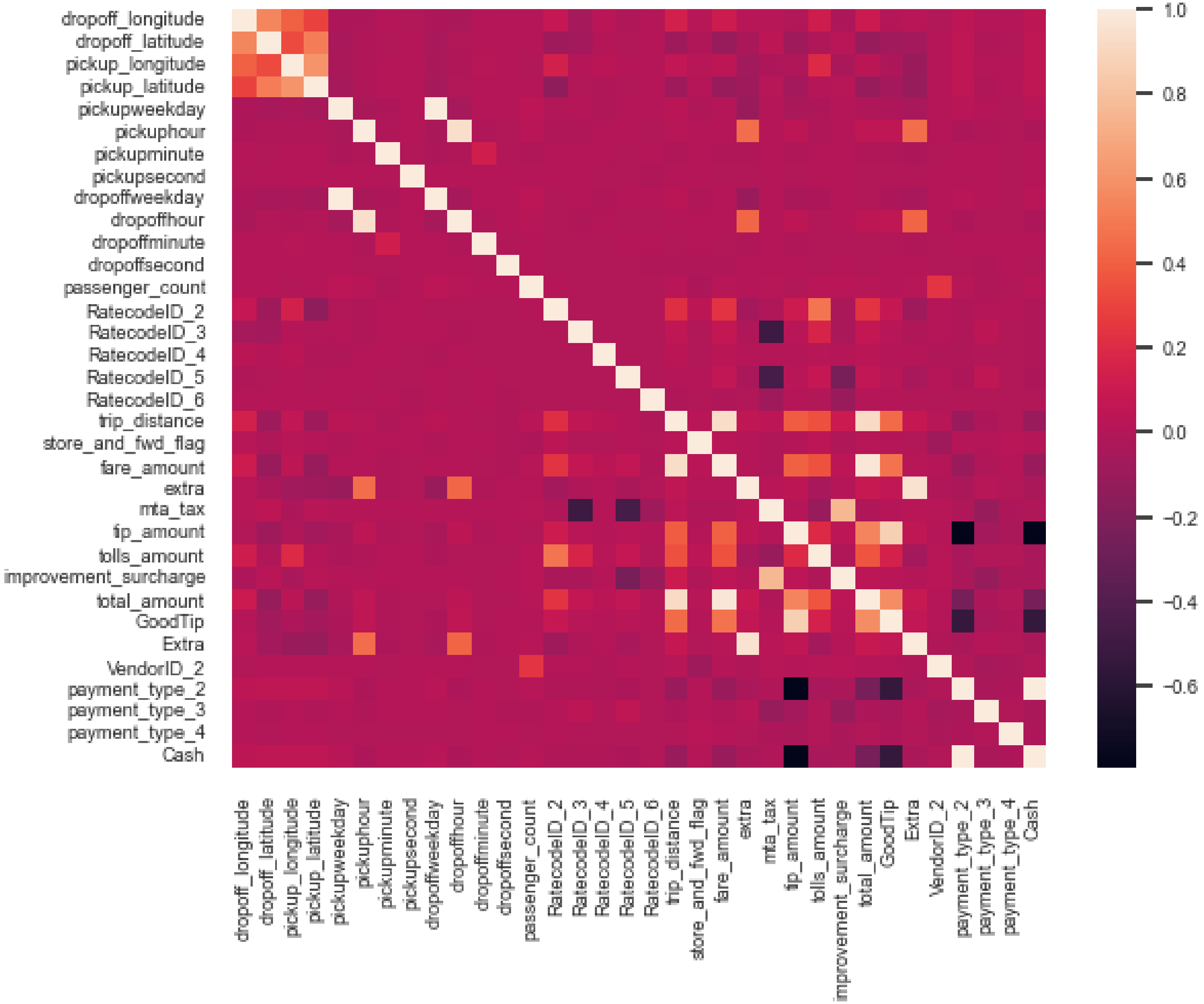
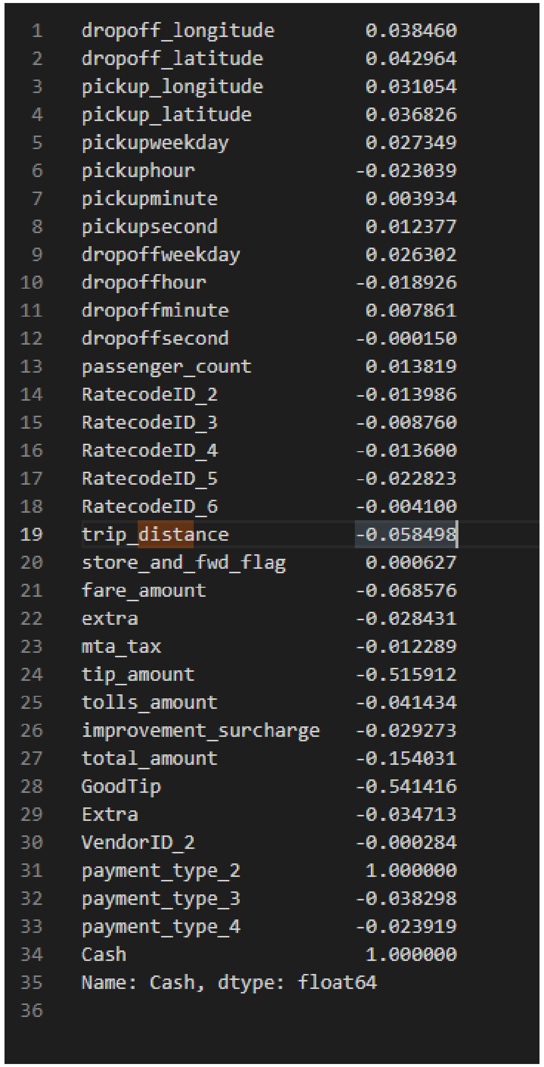


Fig. 4 Correlation Values for Cash Variable



```
dropoff_longitude        0.038460
dropoff_latitude         0.042964
pickup_longitude         0.031054
pickup_latitude          0.036826
pickupweekday            0.027349
pickuphour              -0.023039
pickupminute             0.003934
pickupsecond             0.012377
dropoffweekday           0.026302
dropoffhour             -0.018926
dropoffminute            0.007861
dropoffsecond           -0.000150
passenger_count          0.013819
RatecodeID_2            -0.013986
RatecodeID_3            -0.008760
RatecodeID_4            -0.013600
RatecodeID_5            -0.022823
RatecodeID_6            -0.004100
trip_distance           -0.058498
store_and_fwd_flag       0.000627
fare_amount             -0.068576
extra                   -0.028431
mta_tax                 -0.012289
tip_amount              -0.515912
tolls_amount            -0.041434
improvement_surcharge   -0.029273
total_amount            -0.154031
GoodTip                 -0.541416
Extra                   -0.034713
VendorID_2              -0.000284
payment_type_2           1.000000
payment_type_3          -0.038298
payment_type_4          -0.023919
Cash                     1.000000
Name: Cash, dtype: float64
```

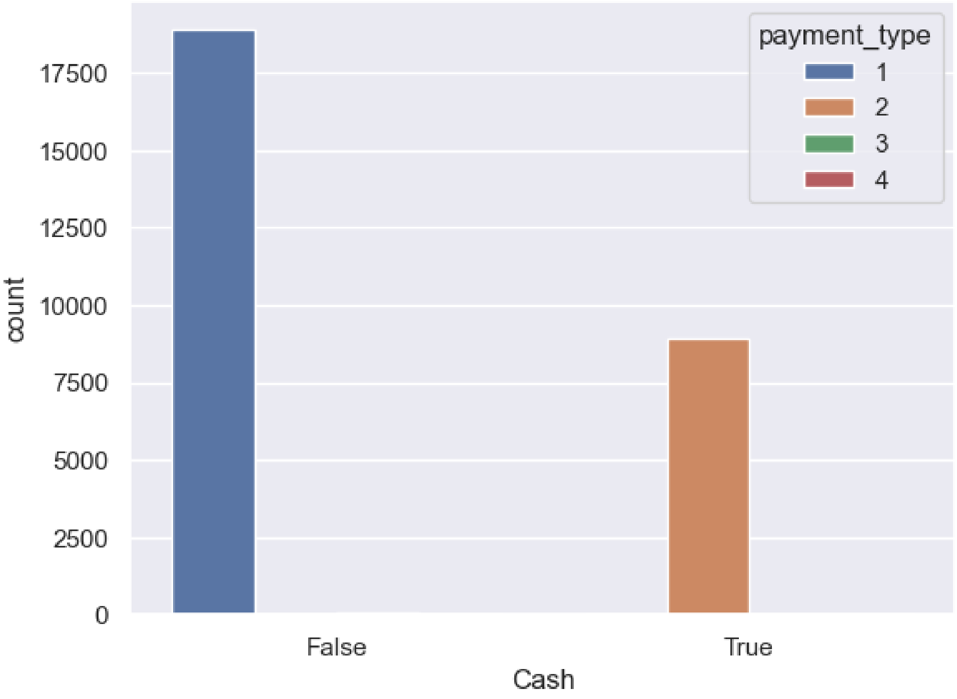


Fig. 5 Count Plot across Payment Types

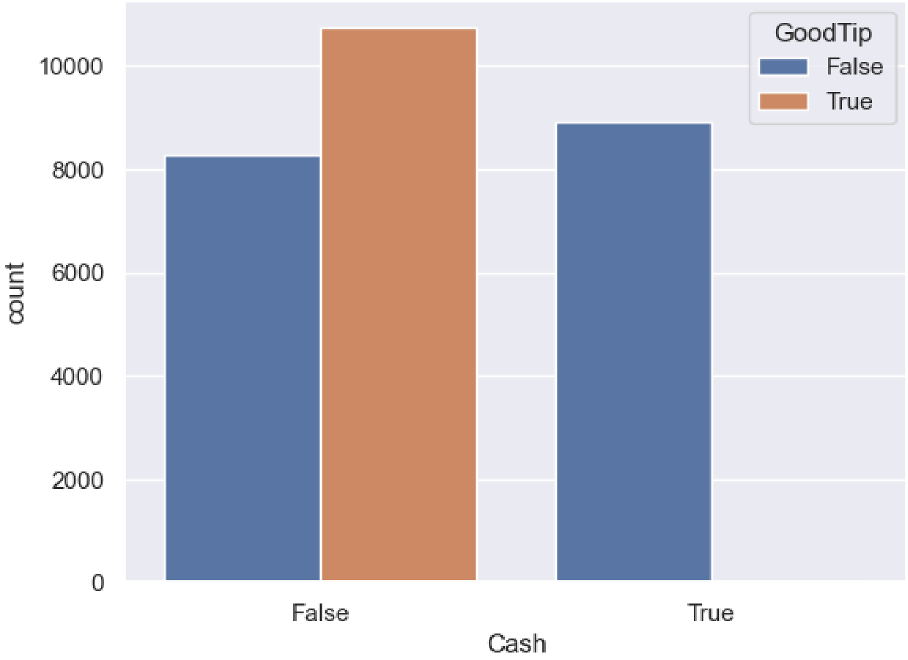


Fig. 6 Count Plot across GoodTip Values

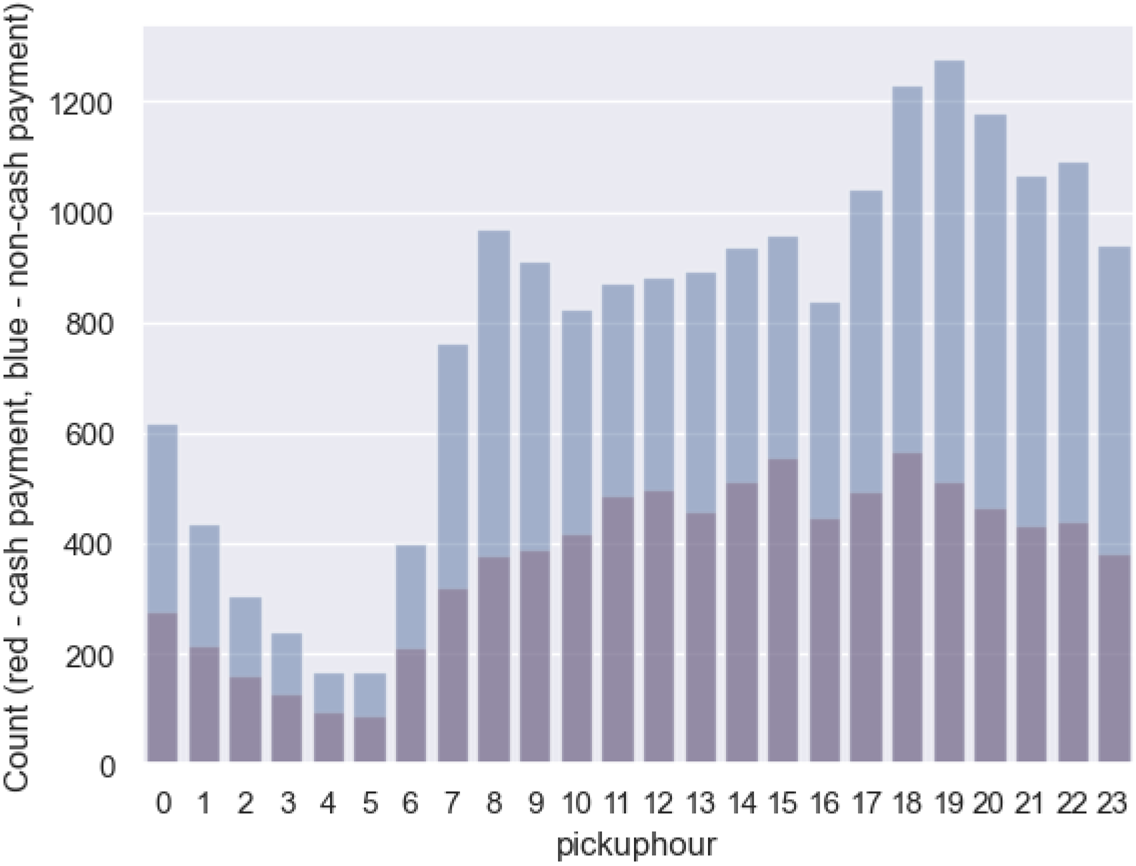


Fig. 7 Cash Values Proportion across Pickup Hours

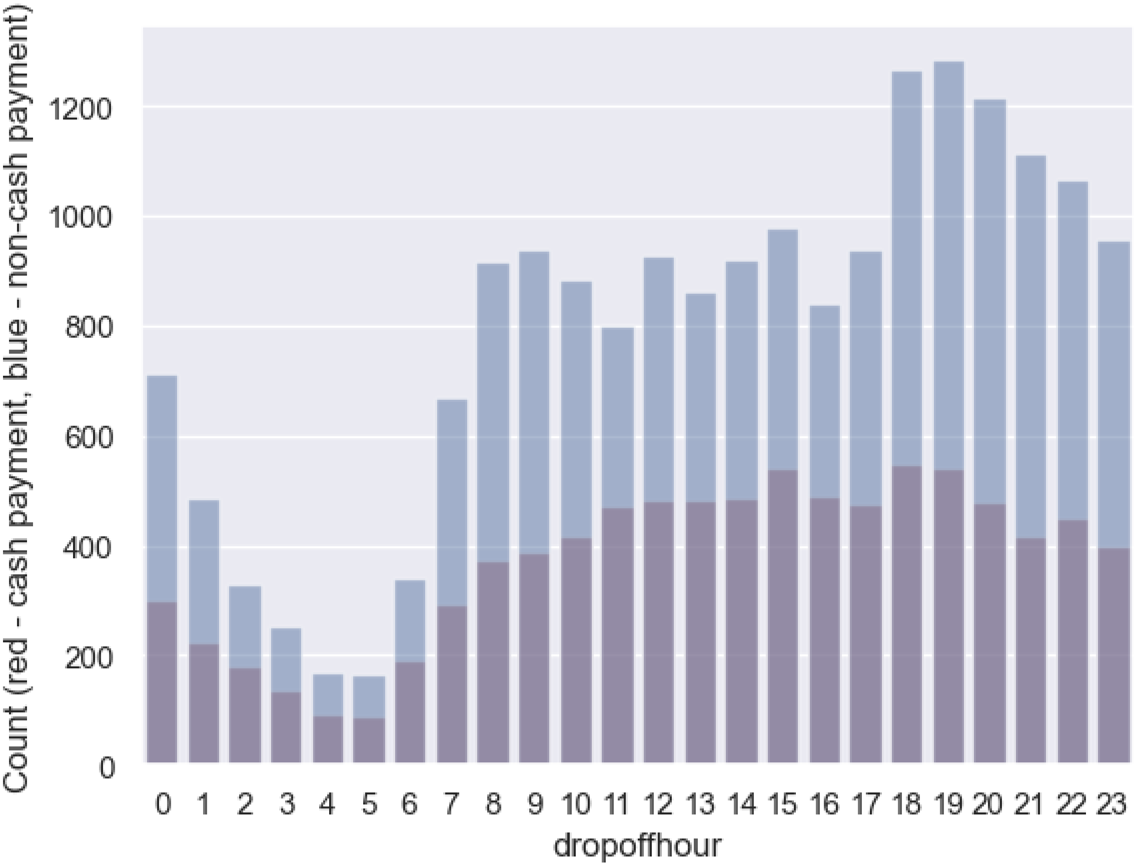


Fig. 8 Cash Values Proportion across Drop-off Hours

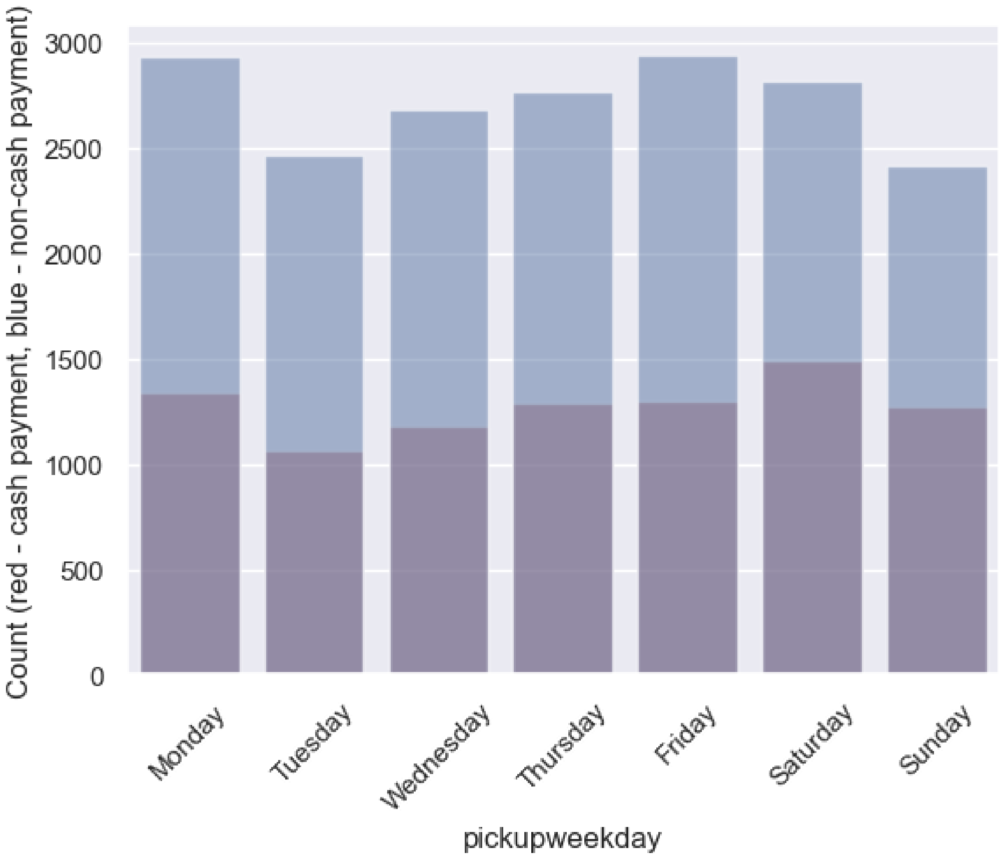


Fig. 9 Cash Values Proportion across Pickup Weekdays

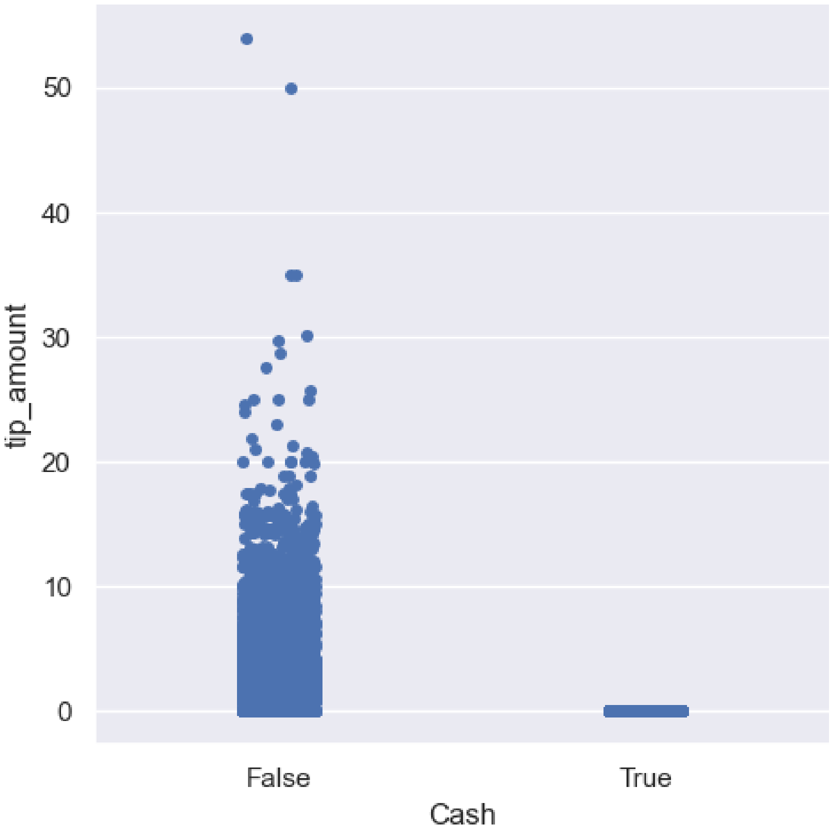


Fig. 10 Cash Values Distributed across tip_amount Values

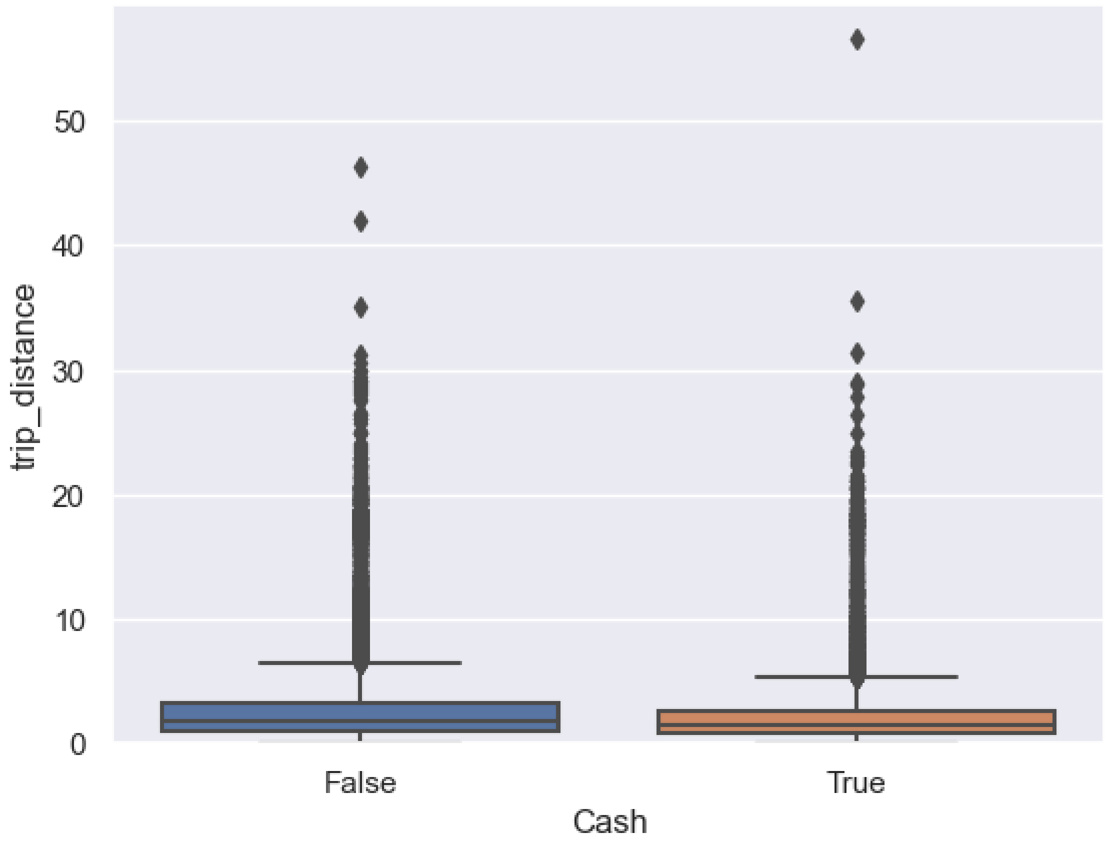


Fig. 11 Cash Values Distributed across
trip_distance Values

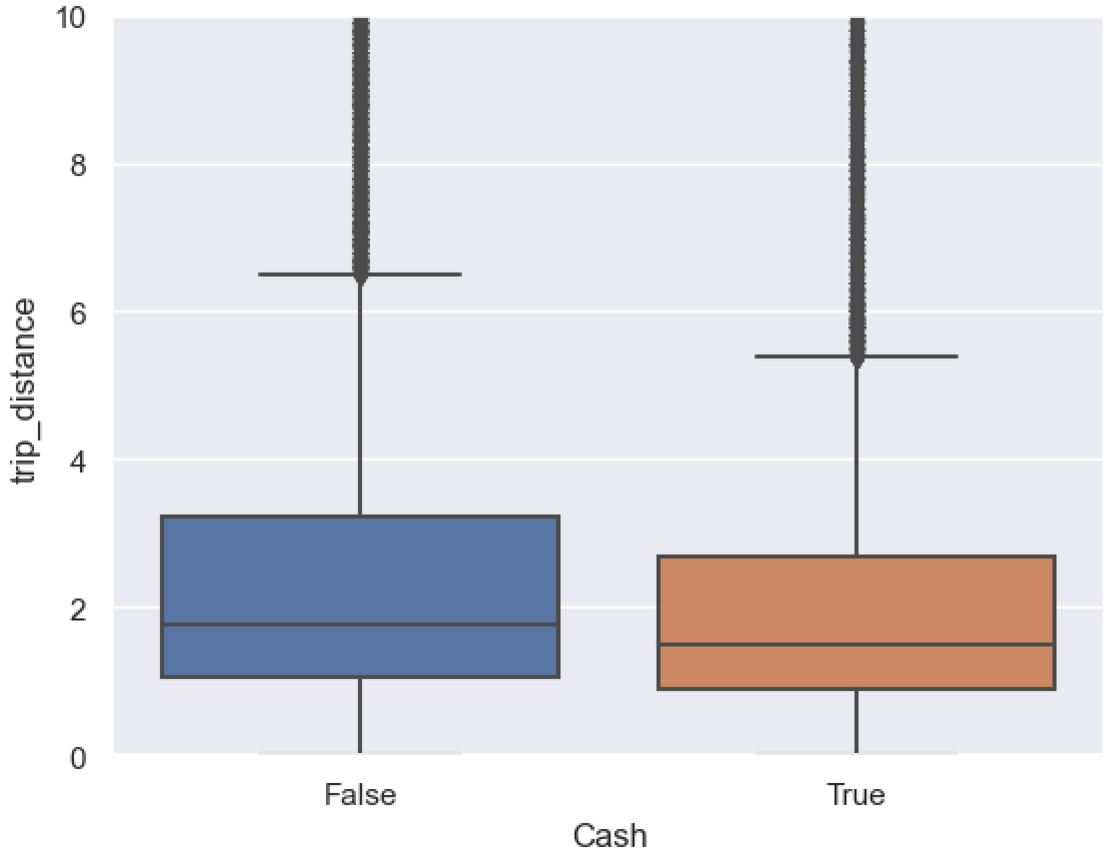


Fig. 11a Cash Values Distributed across
trip_distance Values
(Focused on Low Values of trip_distance)

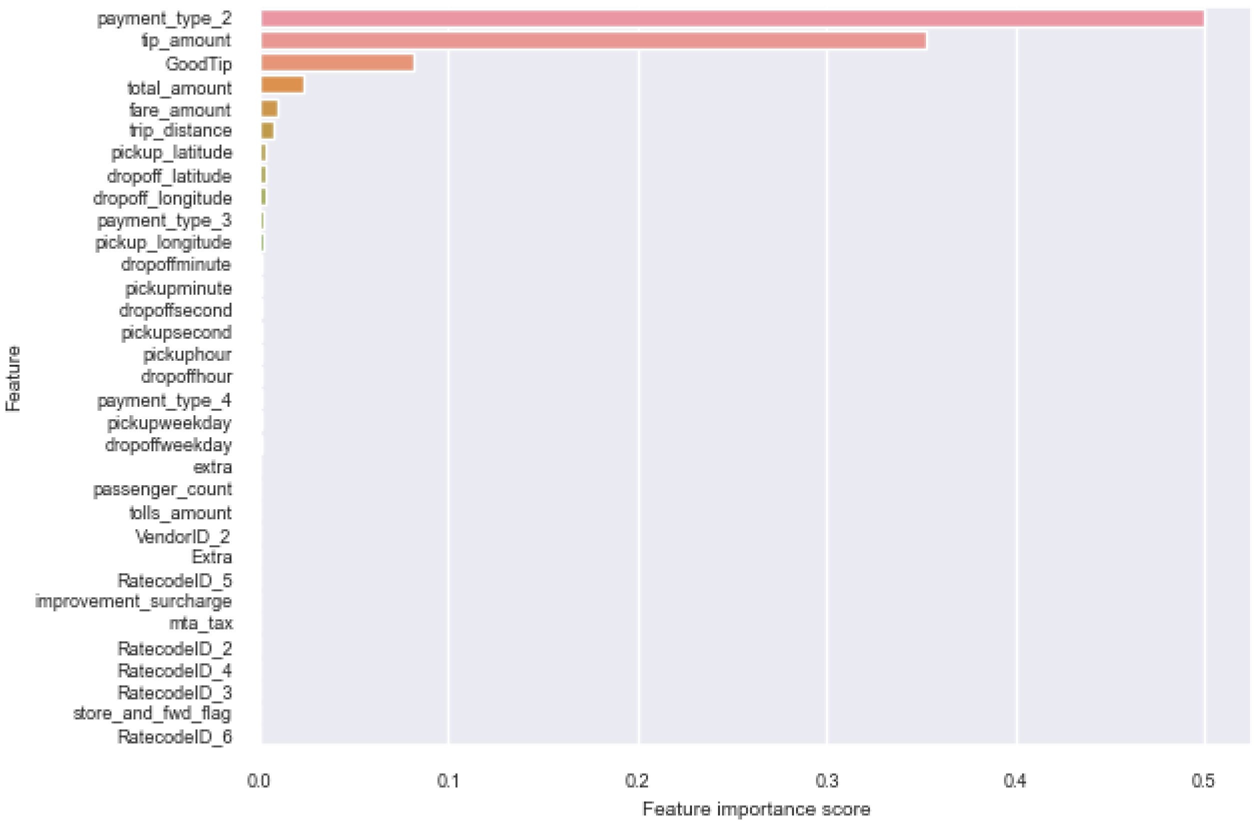
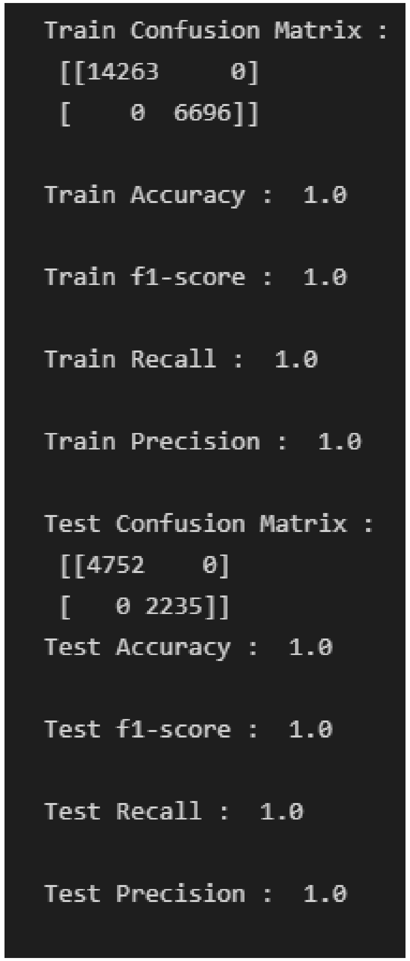


Fig. 12



```
Train Confusion Matrix :
 [[14263     0]
 [    0  6696]]

Train Accuracy :  1.0

Train f1-score :  1.0

Train Recall :  1.0

Train Precision :  1.0

Test Confusion Matrix :
 [[4752    0]
 [   0 2235]]
Test Accuracy :  1.0

Test f1-score :  1.0

Test Recall :  1.0

Test Precision :  1.0
```

Fig. 12a

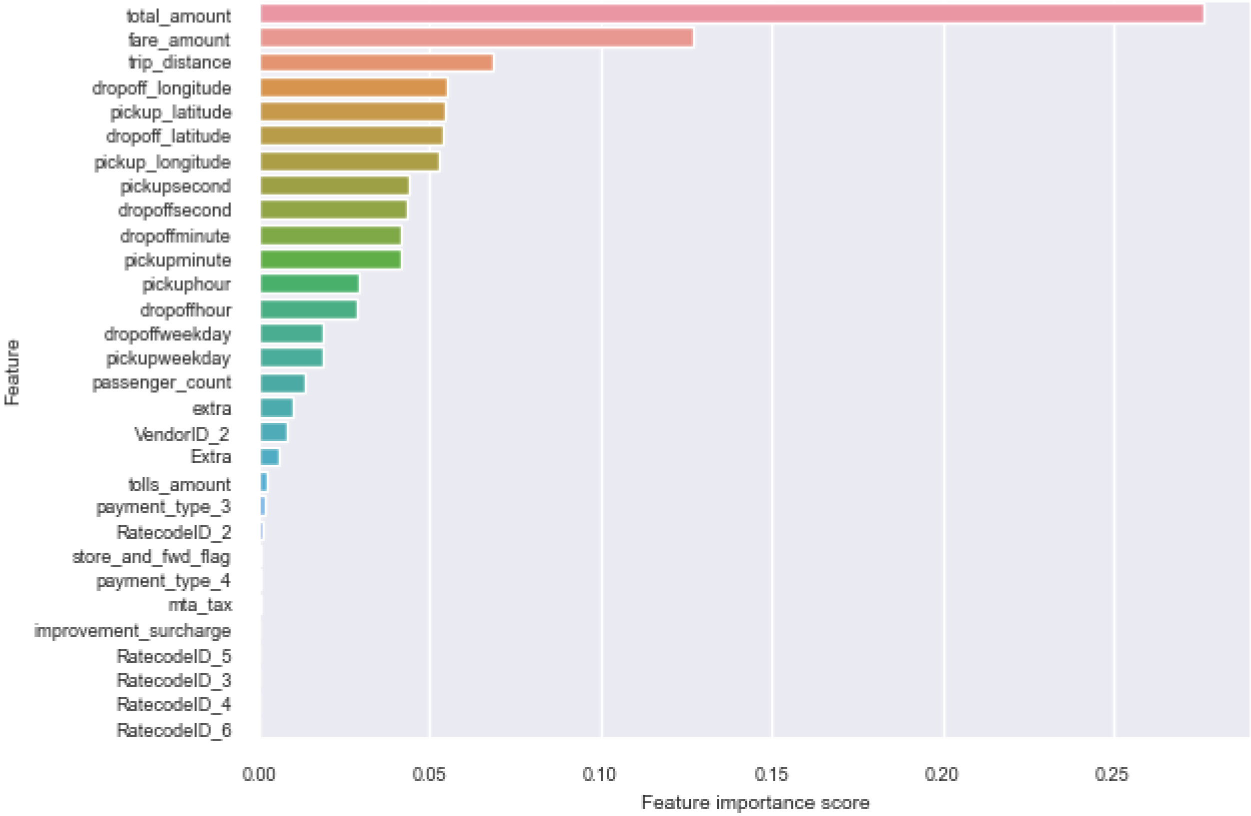
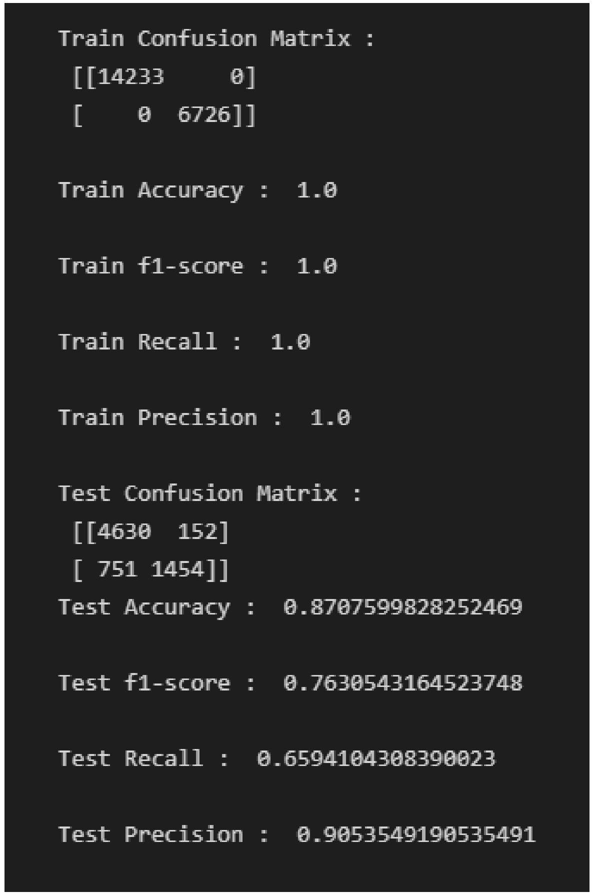


Fig. 13



```
Train Confusion Matrix :
 [[14233     0]
 [    0  6726]]

Train Accuracy :  1.0

Train f1-score :  1.0

Train Recall :  1.0

Train Precision :  1.0

Test Confusion Matrix :
 [[4630  152]
 [ 751 1454]]
Test Accuracy :  0.8707599828252469

Test f1-score :  0.7630543164523748

Test Recall :  0.6594104308390023

Test Precision :  0.9053549190535491
```

Fig. 13a

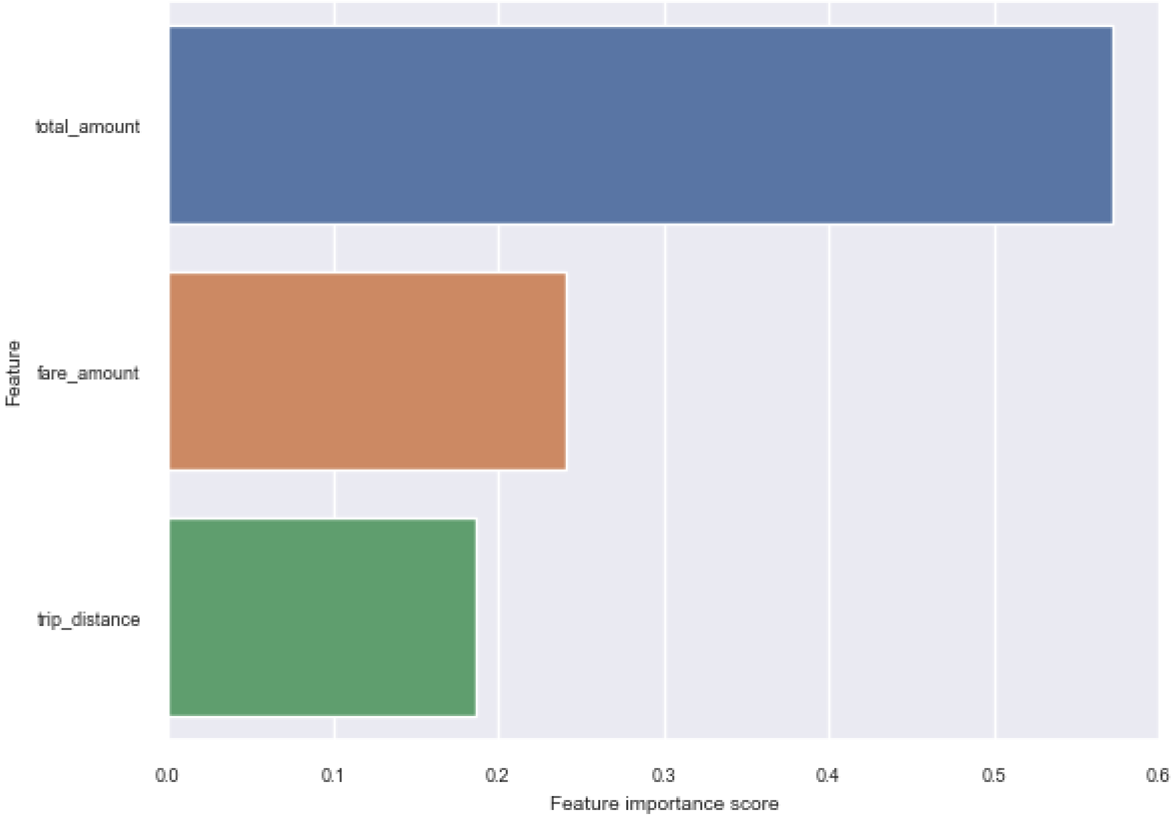
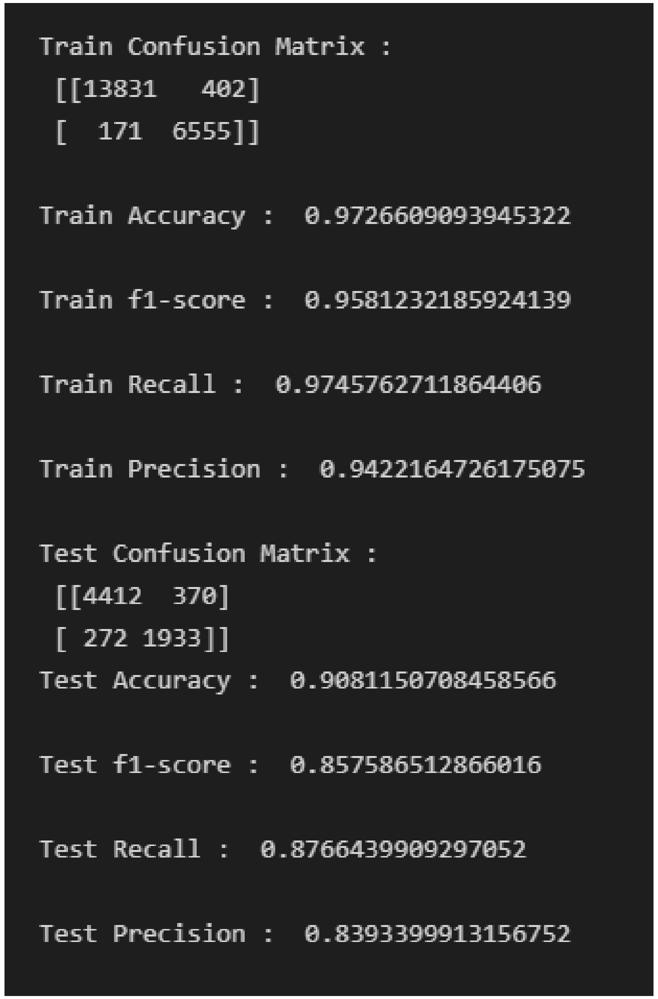


Fig. 14



```
Train Confusion Matrix :
 [[13831   402]
 [  171  6555]]

Train Accuracy :  0.9726609093945322

Train f1-score :  0.9581232185924139

Train Recall :  0.9745762711864406

Train Precision :  0.9422164726175075

Test Confusion Matrix :
 [[4412  370]
 [ 272 1933]]
Test Accuracy :  0.9081150708458566

Test f1-score :  0.857586512866016

Test Recall :  0.8766439909297052

Test Precision :  0.8393399913156752
```

Fig. 14a

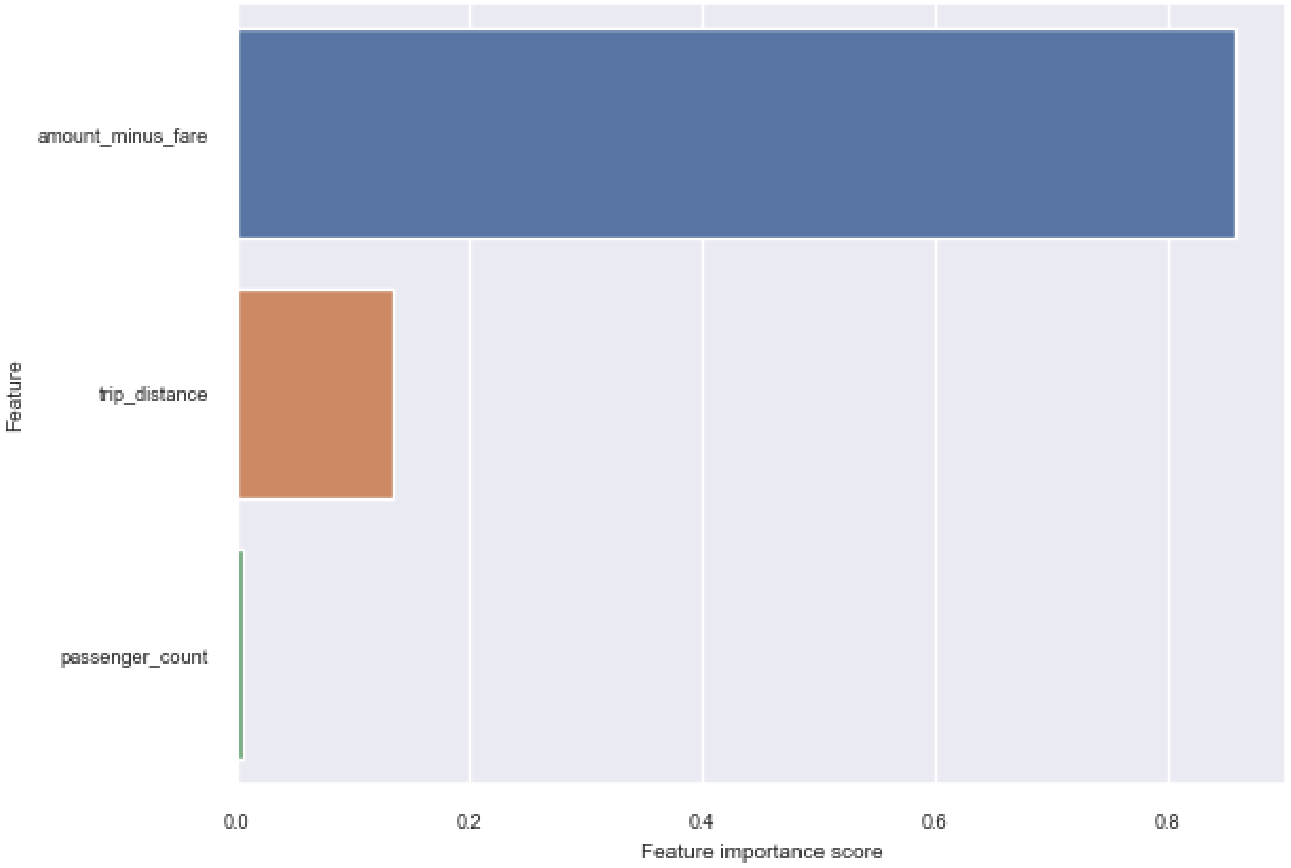
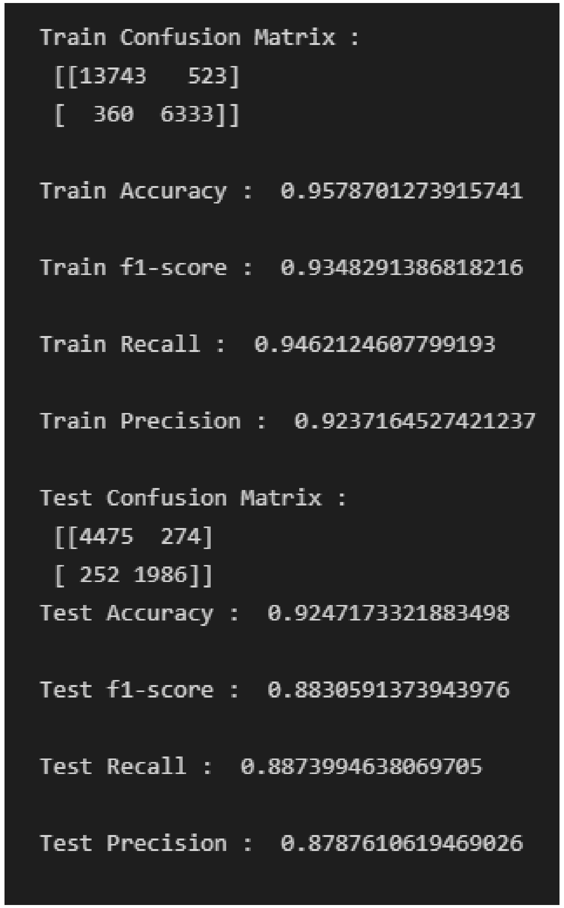
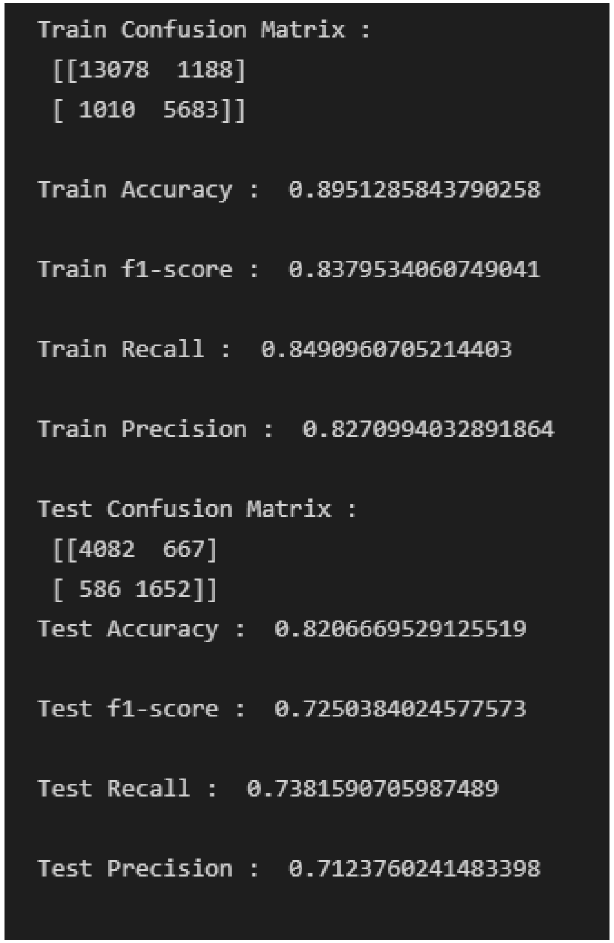


Fig. 15



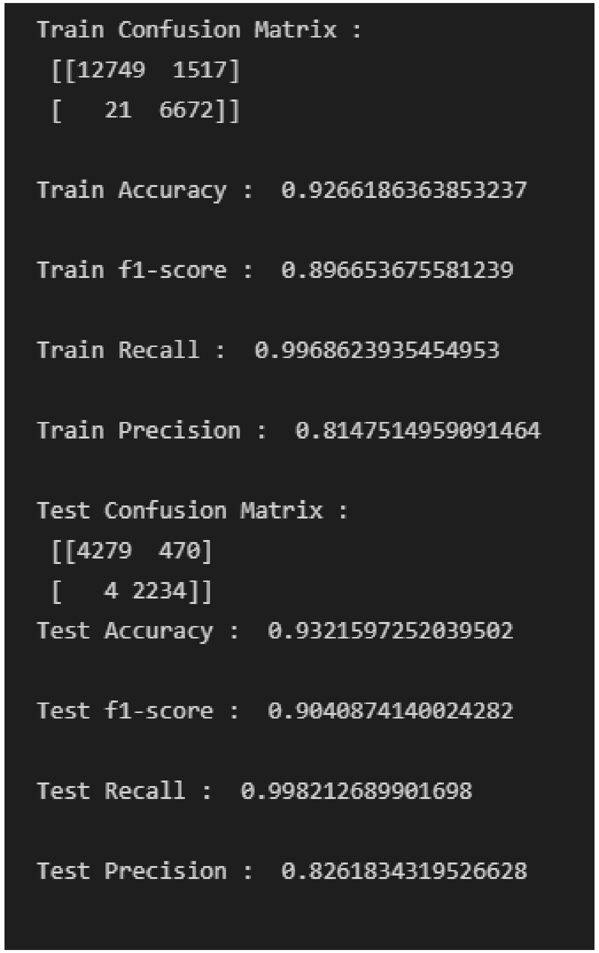
```
Train Confusion Matrix :
 [[13743   523]
 [  360  6333]]

Train Accuracy :  0.9578701273915741

Train f1-score :  0.9348291386818216

Train Recall :  0.9462124607799193

Train Precision :  0.9237164527421237

Test Confusion Matrix :
 [[4475  274]
 [ 252 1986]]
Test Accuracy :  0.9247173321883498

Test f1-score :  0.8830591373943976

Test Recall :  0.8873994638069705

Test Precision :  0.8787610619469026
```

Fig. 15a



```
Train Confusion Matrix :
 [[13078  1188]
 [ 1010  5683]]

Train Accuracy :  0.8951285843790258

Train f1-score :  0.8379534060749041

Train Recall :  0.8490960705214403

Train Precision :  0.8270994032891864

Test Confusion Matrix :
 [[4082  667]
 [ 586 1652]]
Test Accuracy :  0.8206669529125519

Test f1-score :  0.7250384024577573

Test Recall :  0.7381590705987489

Test Precision :  0.7123760241483398
```

Fig. 16



```
Train Confusion Matrix :
 [[12749  1517]
 [   21  6672]]

Train Accuracy :  0.9266186363853237

Train f1-score :  0.896653675581239

Train Recall :  0.996862393545495З

Train Precision :  0.8147514959091464

Test Confusion Matrix :
 [[4279  470]
 [   4 2234]]
Test Accuracy :  0.9321597252039502

Test f1-score :  0.9040874140024282

Test Recall :  0.998212689901698

Test Precision :  0.8261834319526628
```

Fig. 17

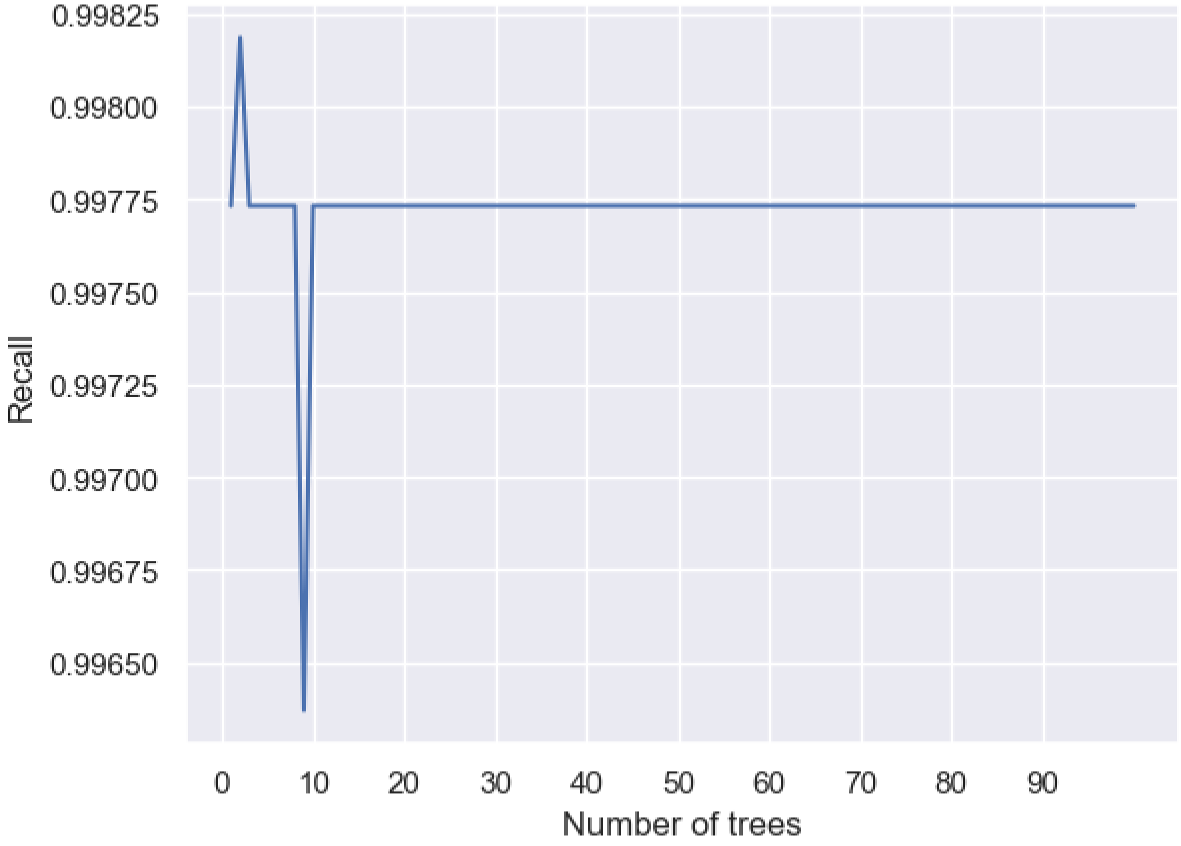

Recall
0.99825
0.99800
0.99775
0.99750
0.99725
0.99700
0.99675
0.99650
0
10
20
30
40
50
60
70
80
90
Number of trees

Fig. 18

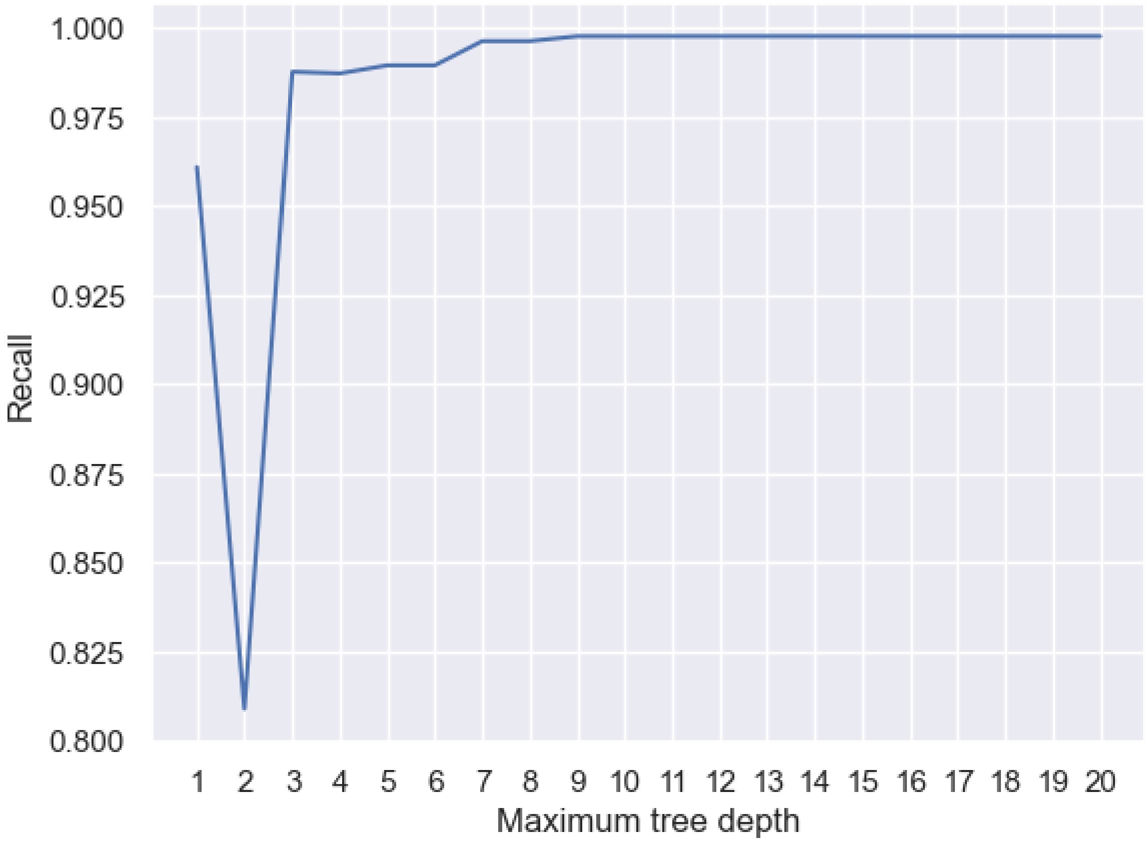

Recall
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Maximum tree depth

Fig. 19

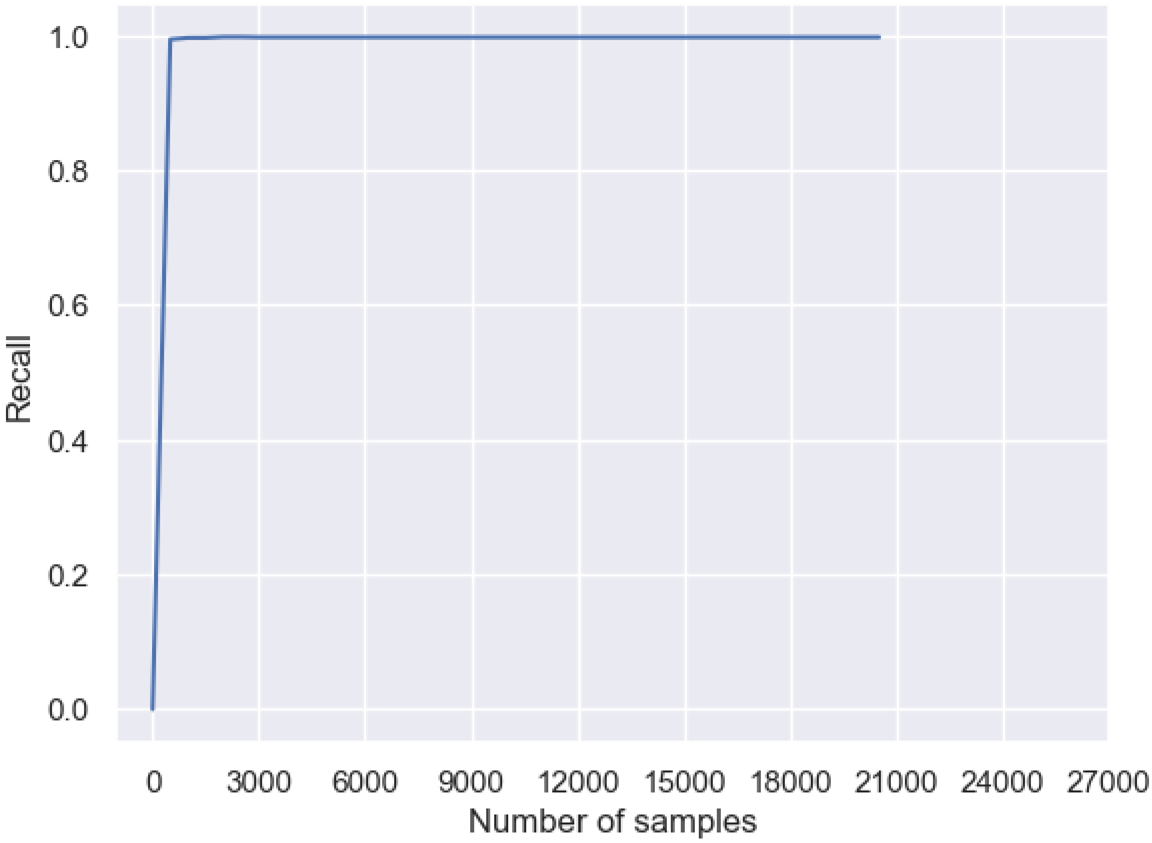

1.0
0.8
0.6
0.4
0.2
0.0
Recall
0
3000
6000
9000
12000
15000
18000
21000
24000
27000
Number of samples

Fig. 20