**CS6923, Machine Learning**

**Assignment #5**

**Submitted By :**

Rakshit Sareen (rs5606)

**Problem Definition**

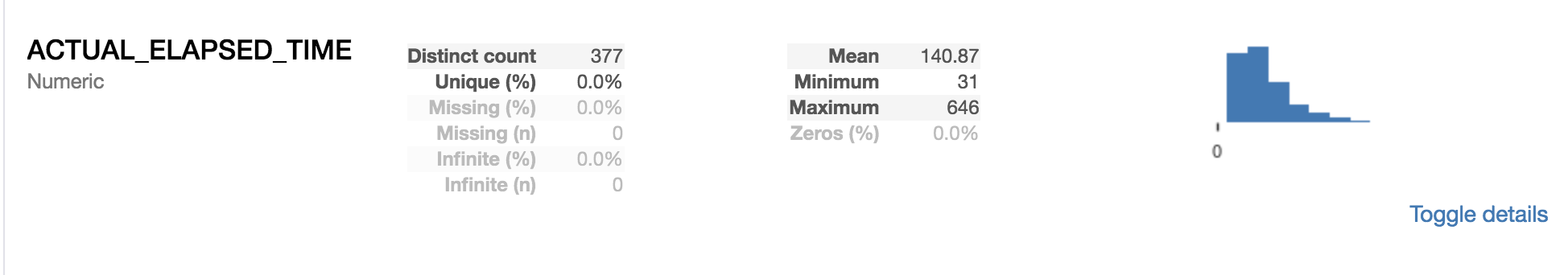
The project is about predicting the delay of a flight. This is a regression problem and we can use many regression algorithms to predict the delay in the flight.

There are many choices that can be used for this purpose, like K-NN, Decision Trees, Random Forest, Gradient Boosted Trees (Ensemble Methods), Linear Regression, Neural Networks etc.

I decided to work with four algorithms: Linear Regression, Decision Trees, Random Forest Regressor and Gradient Boosting Regressor.

**Data Exploration**

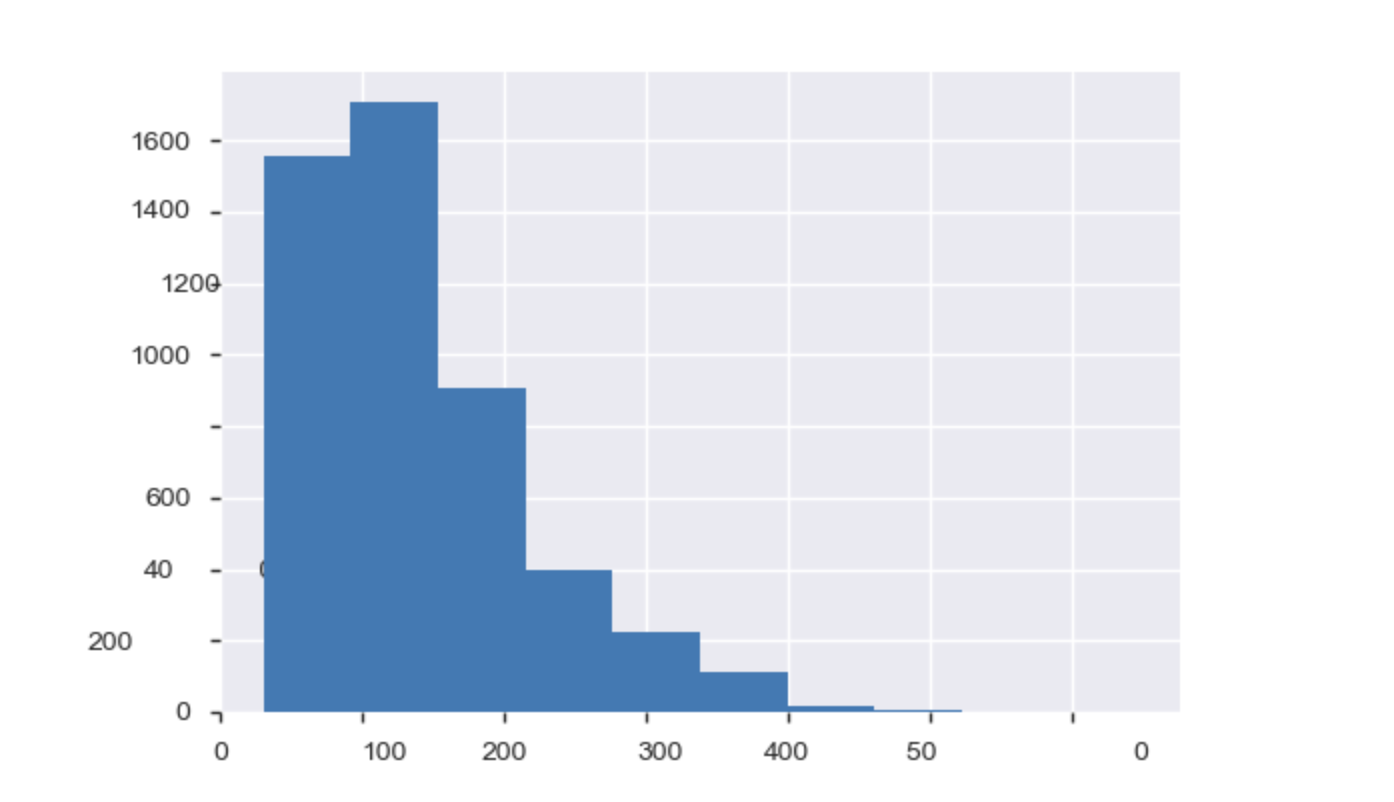
This section presents some of the many data explorations performed in the assignment, as can be seen in the jupyter notebook.



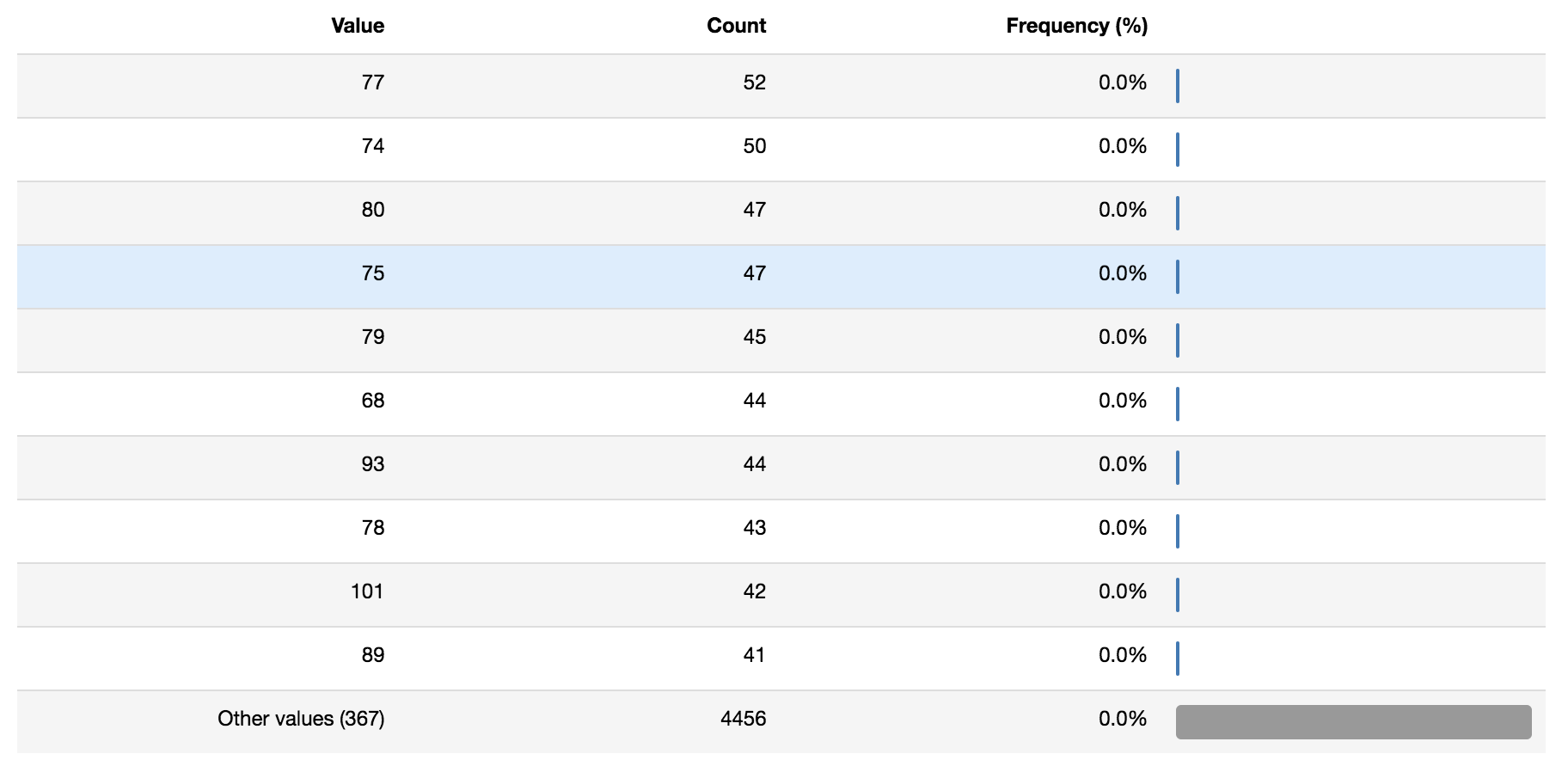
Statistics



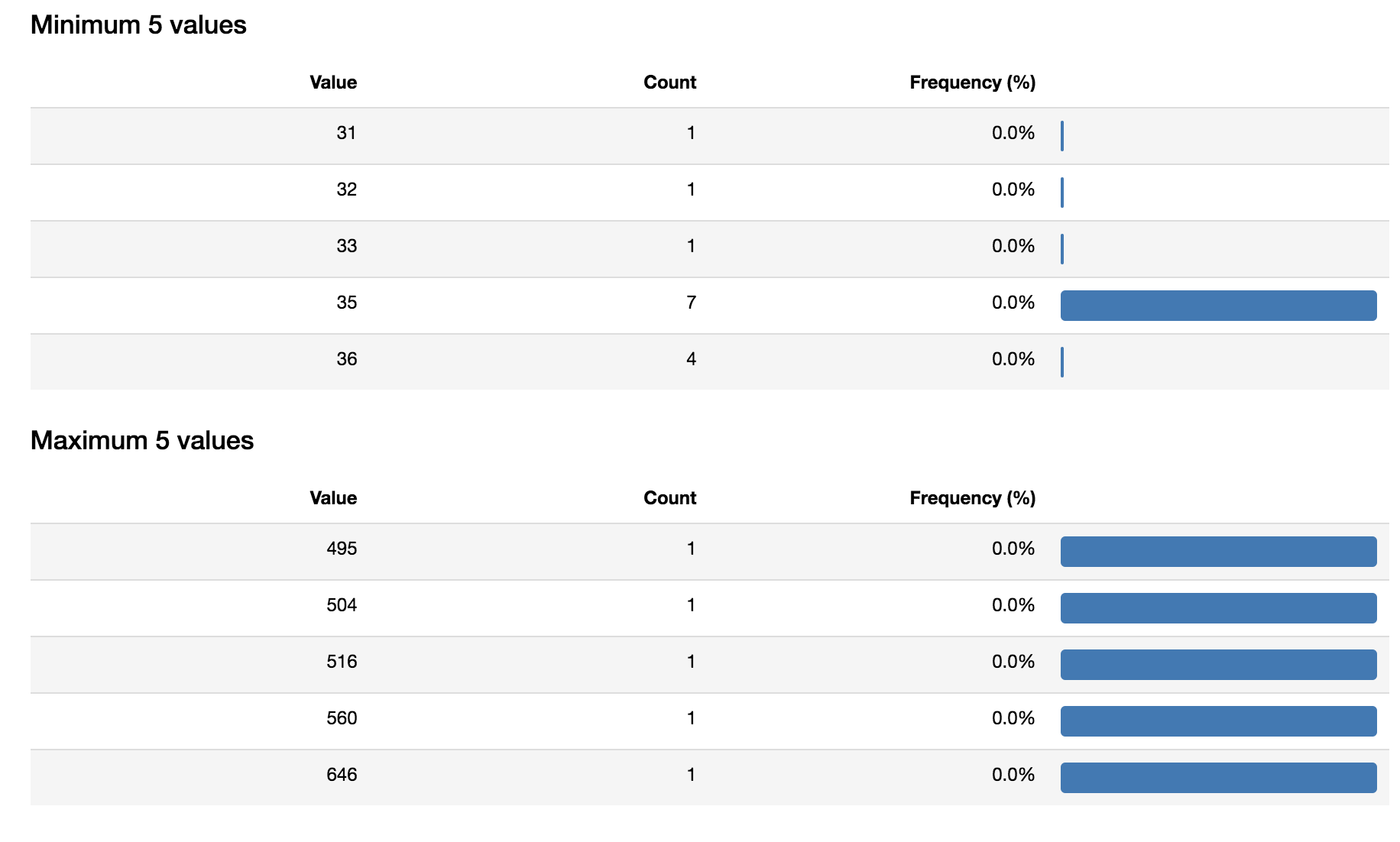
Histogram



Common Values



Extreme Values



**Feature Engineering**

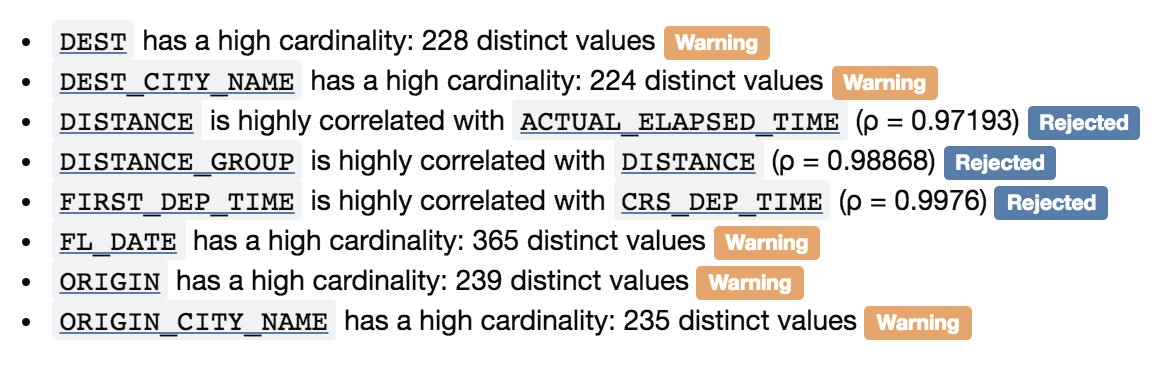
Feature Dropping

I decided to explore the features and get to know more about the data. I found out various features that are not relevant to the task at hand. Some of the features were highly correlated and some had too much cardinality. I decided to drop those.

Here is a table of the features I dropped and the reason for it.

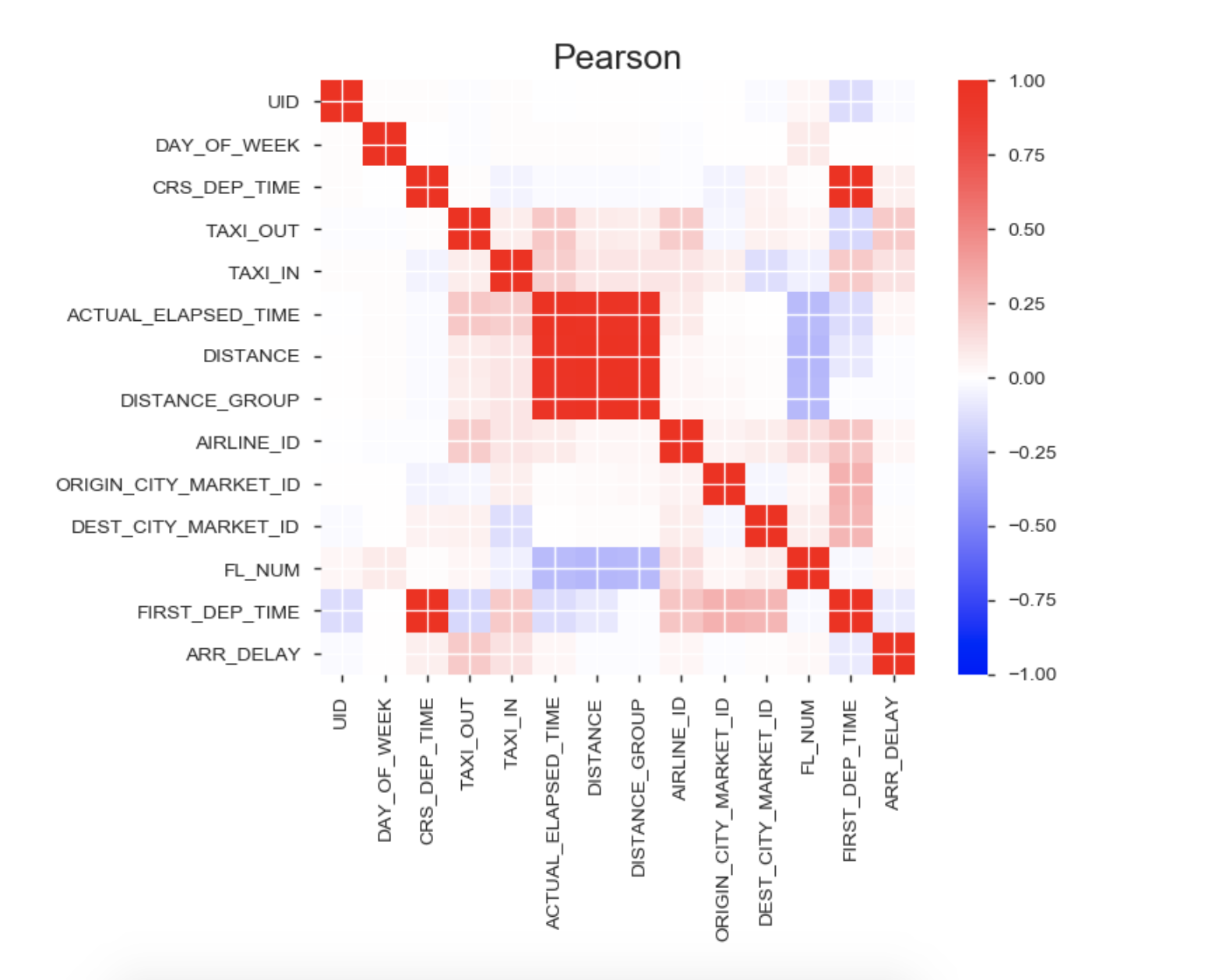
**Feature Reason of Dropping**

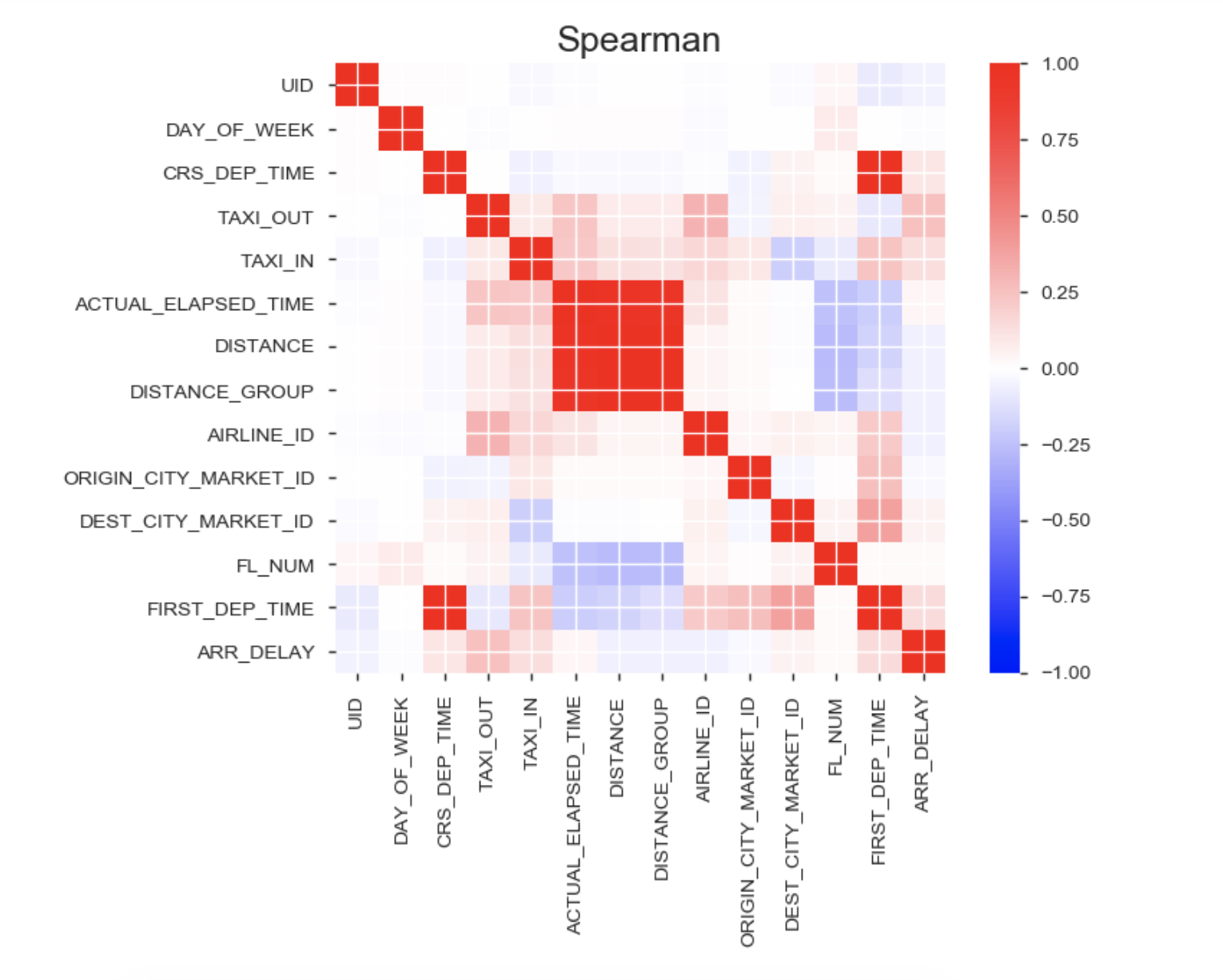
|  |  |
| --- | --- |
| 'ORIGIN\_STATE\_ABR' | High Cardinality |
| 'UNIQUE\_CARRIER' | Same as AIRLINE\_ID - High Correlation |
| 'DEST\_STATE\_ABR' | High Cardinality |
| 'UID' | Just a unique value, Not important to analysis |
| 'DEST' | High Cardinality |
| 'DEST\_CITY\_NAME' | High Cardinality |
| 'DISTANCE\_GROUP' | Highly correlated with DISTANCE |
| 'FIRST\_DEP\_TIME' | Missing Values for almost all of the data |
| 'FL\_DATE' | High Cardinality |
| 'ORIGIN' | High Cardinality |
| 'ORIGIN\_CITY\_NAME' | High Cardinality |
| 'ACTUAL\_ELAPSED\_TIME' | Highly correlated with DISTANCE |
| 'CRS\_DEP\_TIME' | Highly correlated with FIRST\_DEP\_TIME |
| 'FL\_NUM' | High Cardinality |



I tried to find out the correlation of features among themselves.

Here I present two correlation matrices, Pearson and Spearman.





Feature Creation

Creation of new features is an important part of data analysis and exploration. We can add missing data, augment data with new columns , extract meaning from different features and combine them into a new feature.

Below table presents what features have been created and their description.

**Feature Description**

|  |  |
| --- | --- |
| Day | The day of the month |
| Month | The month in the |
| SPEED | The speed of the airplane in the journey |
| IS\_HOLIDAY | 1,0 based on whether FL\_DATE was holiday |
| IS\_WEEKEND | 1,0 based on whether FL\_DATE was a weekend |

Feature Encoding

I decided to encode the AIRLINE\_ID. I used one-hot encoding provided by the pandas library.

The reason to use one-hot encoding instead of integer encoding is because AIRLINE\_ID does not have any ordered relationship with each other.

Also, since AIRLINE\_ID did not have much cardinality, I decided to use one-hot encoding to it.

**GRID SEARCH CV (CROSS VALIDATION = 10)**

I performed grid search over the hyperparameters for each of the chosen models. The parameters chosen were adopted by Cross Validation with cv = 10.

Here are the parameters used for all the algorithms and their values:

**Linear Regression**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Values** |
| fit\_intercept | Whether to calculate the intercept for this model | True, False |
| normalize | X will be normalized before regression | True, False |

Grid Search CV provides the best score and the best parameters it found.

Result :

best\_score\_ : 1940.90185776

best\_params\_ :{'normalize': False, 'fit\_intercept': True}

**Decision Trees**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Values** |
| splitter | Strategy used at each split of the node | best ,random |
| max\_features | The number of features to consider when looking for the best split | auto, sqrt, log2 |
| min\_samples\_split | The minimum number of samples required to split an internal node | 2,4,8 |
| max\_depth | The maximum depth of the tree | 2,4,6,8,10 |
| presort | Presort the data to speed up the finding of best splits in fitting | True |

Result :

best\_score\_ : 1999.59277975

best\_params\_ : {'max\_features': 'auto', 'min\_samples\_split': 2, 'presort': True, 'max\_depth': 2, 'splitter': 'random'}

**Random Forest**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Values** |
| n\_estimators | The number of trees in the forest. | 10,20,30,40,50,60 |
| max\_features | The number of features to consider when looking for the best split | auto, sqrt, log2 |
| min\_samples\_split | The minimum number of samples required to split an internal node | 2,4,8 |
| bootstrap | Whether bootstrap samples are used when building trees. | True, False |
| warm\_start | When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest. | True,False |

Result :

best\_score\_ : 1985.79949748

best\_params\_ : {'max\_features': 'log2', 'min\_samples\_split': 8, 'bootstrap': True, 'n\_estimators': 60, 'warm\_start': True}

**Gradient Boosting**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Values** |
| n\_estimators | The number of trees in the forest. | 10,20,30,40,50,60 |
| max\_features | The number of features to consider when looking for the best split | auto, sqrt, log2 |
| min\_samples\_split | The minimum number of samples required to split an internal node | 2,4,8 |
| learning\_rate | learning rate shrinks the contribution of each tree by learning\_rate. | True, False |
| warm\_start | When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest. | True,False |

Result :

best\_score\_ : 1943.52304248

best\_params\_ : {'warm\_start': False, 'loss': 'ls', 'learning\_rate': 0.2, 'n\_estimators': 40, 'min\_samples\_split': 2, 'max\_features': 'auto'}