Text

Description automatically generated with low confidence

Capstone Project Final Report

Group 4

Bhanoji Reddy Meka

NORTHEASTERN UNIVERSITY

MPS ANALYTICS

CLASS: ALY6140

CRN: 20532

Analytics Systems Technology

INSTRUCTOR NAME: RICHARD HE

DATE: 03/30/23

**INTRODUCTION**

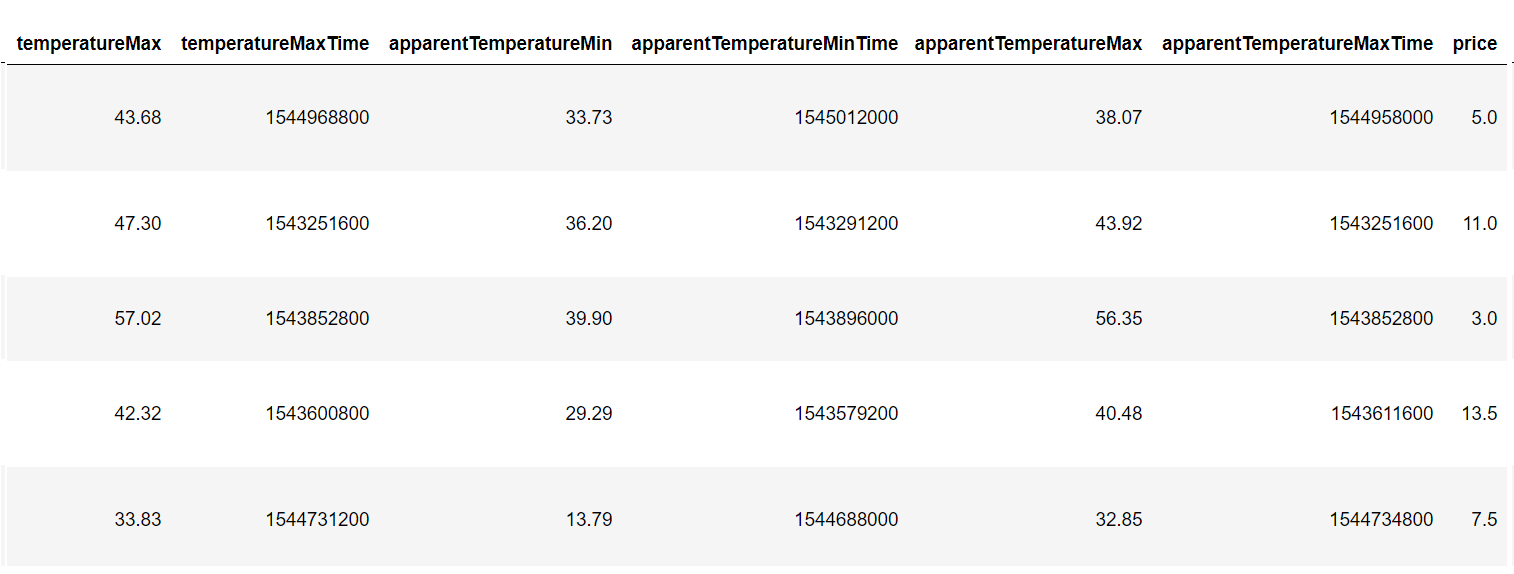
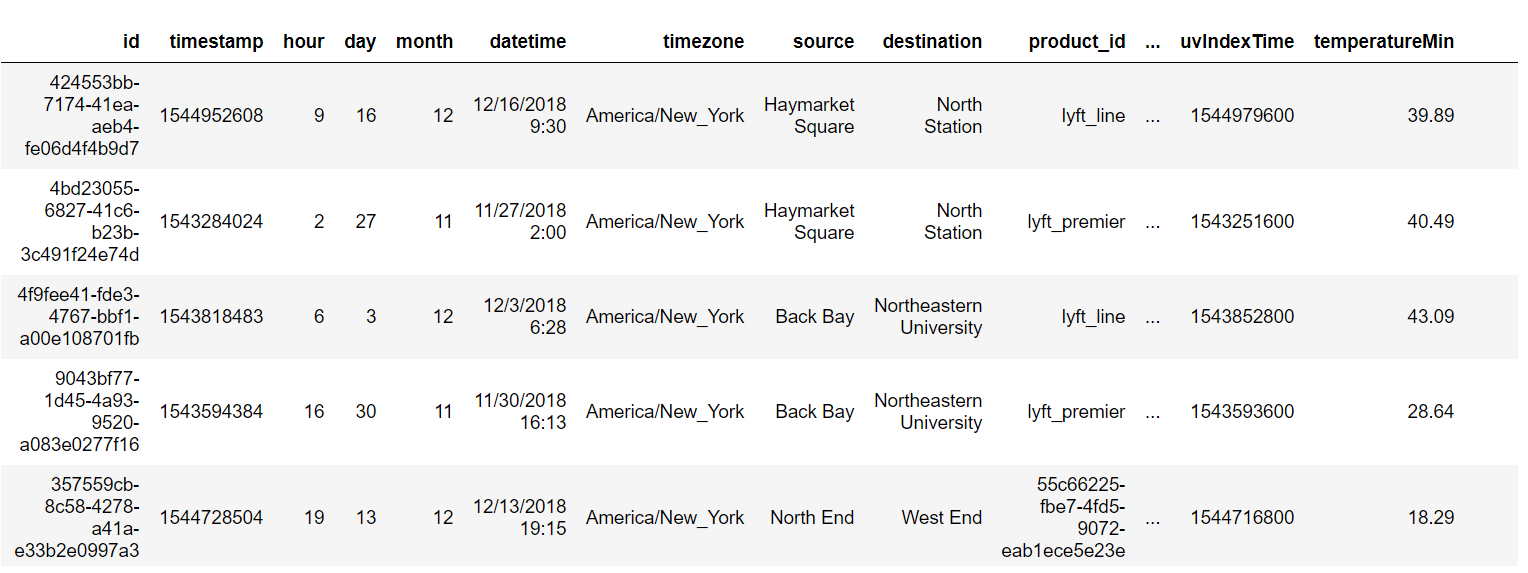
Uber is a popular company that provided ride-sharing services, as well as food and package delivery, courier services, and freight transportation. Additionally, they partnered with Lime to offer electric bicycle and motorized scooter rentals. The company was founded in 2009 by Travis Kalanick and Garrett Camp, who was a thriving technology entrepreneur. Garrett Camp came up with the idea for Uber after selling his previous startup to eBay and noticing the significant taxi issues in San Francisco.

Supervised learning involves a training set and a test set. These sets comprise examples containing input and output vectors, and the objective of the supervised learning algorithm is to deduce a function that maps the input vector to the output vector while minimizing errors. We utilized machine learning algorithms to predict the Price in the Boston Uber Dataset by selecting multiple features from 56 columns. Predictive analysis is a computational technique that aims to uncover crucial and valuable patterns in large datasets.

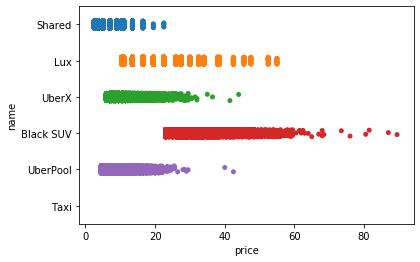
**Exploratory Data Analysis:**

Data Preparation:

Our analysis utilized data obtained from [www.kaggle.com](http://www.kaggle.com), with the initial dataset comprising 693071 rows and 57 columns, which encompassed information for both Uber and Lyft. Nevertheless, our project solely necessitated Uber's data, and we consequently focused our attention on that subset. To cater to our research objectives, we filtered the data and obtained a fresh dataset consisting of 322844 rows and 56 columns. The dataset encompasses various fields that offer valuable details concerning the time, geographic location, and weather conditions during the usage of diverse Uber cabs. Our dataset comprises three primary datatypes: integer, float, and object. Unfortunately, we encountered an issue where the dataset is incomplete, and the price column has null values that affect approximately 55095 entries.



The below chart indicated that Shared trips were the most affordable while BlackSuv rides were the priciest. UberX and UberPool had similar prices, while Lux had a moderately high cost. Interestingly, there was no chart for taxi, implying that the dataset lacked values for this category.



Upon analyzing the below chart, it became apparent that several outliers existed in the cloudy weather category, with certain entries having a significantly high price exceeding 80, while others were below 60. The plot revealed that the highest prices were observed during cloudy weather, whereas the lowest prices were recorded during foggy weather.



The bar chart displayed above depicts the value count of the icon column. Upon examining the graph, it becomes evident that the cloudy weather category has the most significant number of entries. This finding leads us to speculate that perhaps cabs were most frequently utilized during cloudy weather conditions.

Bar chart

Description automatically generated with low confidence

**Data Cleaning:**

Filling Null values:

To identify any missing values in a Pandas DataFrame, we can utilize the isnull() function. Upon inspection, we discovered that the price column in our dataset contained 55095 NaN values. To address this issue, we can use the fillna() function to fill in these null values. To address the missing values in the price column, we replaced them with the median value of the remaining dataset values. Additionally, we converted the price values from float to integer since prices cannot be represented as a float. To facilitate visualization, we created a bar chart to depict the value count of prices.

Shape, rectangle

Description automatically generated

**RFE (Recursive Feature Elimination):**

There are two important configuration options when using RFE:

* The choice in the number of features to select (k value)
* The choice of the algorithm used to choose features.

Our implementation involves utilizing scikit-learn's RFE (Recursive Feature Elimination) method via the sklearn.feature\_selection.RFE class.

Following the division of our dataset into dependent (features) and independent (target) variables, and splitting it into train and test sets, we applied RFE with the Linear Regression model. The obtained accuracies varied for different values of k (number of features), as listed below:

|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **No. of Feature (K)​** | **Accuracy​** |
| 1​ | 56​ | 0.8054834220​ |
| 2​ | 40​ | 0.8050662132​ |
| 3​ | 25​ | 0.8055355151​ |
| 4​ | 15​ | 0.8050457819​  ​ |

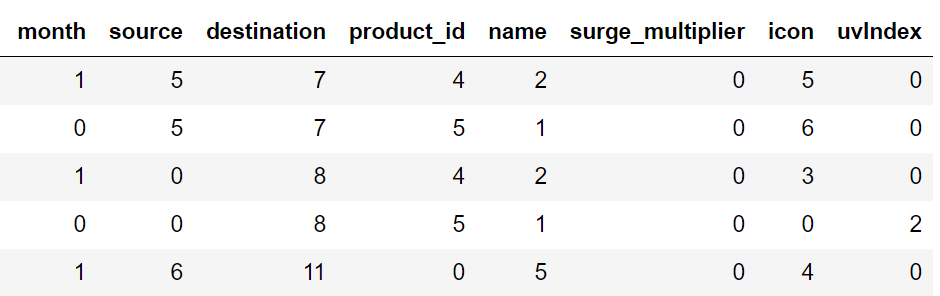
Based on the table presented above, it is evident that the accuracy is the highest for 25 features compared to other values of k. This indicates that these 25 features are the most effective ones provided by RFE. Therefore, we selected only these 25 features and discarded the remaining ones. As a result, our dataset is now reduced from 56 features to 25 features.

Once we obtained our 25 best features through RFE, we further eliminated the features that do not have a direct impact on the price. As a result, only eight features were left in our dataset. To drop these unnecessary features, we utilized the drop() method, which removes specific columns or rows based on the corresponding axis and column names.

**Binning:**

Data smoothing is a common technique used to preprocess data. Binning is one of the methods used in data smoothing. It involves grouping data into specific ranges, or bins, and any values that fall within a bin are assigned to that bin. This technique is used to handle noisy data and make it more manageable for analysis.

After eliminating unnecessary features, some of the remaining features may not be in the same range. To address this issue, we can perform binning, which involves defining a range, or "bin," and assigning any data value within that range to the bin. This technique can help smooth out the data and handle noisy data. Once we apply binning to these features, we can obtain our final dataset, which can be used for modeling.



**Modelling:**

Modeling involves training a machine-learning algorithm to make predictions on labels based on input features. The model is then fine-tuned for business needs and validated using holdout data. The accuracy of a machine learning model is crucial in assessing its performance. To determine accuracy, the model is trained using the given dataset and used to predict response values for the same dataset, and the accuracy is calculated based on the predictions.

Scikit-Learn is utilized in this project to quickly implement several models, including Linear Regression, Decision Tree, Random Forest, and Gradient Boosting.

Graphical user interface, text, application, email

Description automatically generated

**A picture containing graphical user interface

Description automatically generated**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

The modeling is done in the following steps:

* First, we split the dataset into a training set and a testing set.
* Then we train the model on the training set.
* And at last, we test the model on the testing set and evaluate how well our model performs.

So after applying these models we get the following accuracy:

|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **Models** | **Accuracy​** |
| 1​ | Linear Regression​ | 0.747545073​ |
| 2​ | Decision Tree​ | 0.961791729​​ |
| 3​ | Random Forest​ | 0.962269474 |
| 4​ | Gradient Boosting Regressor | 0.963187213​  ​ |

**Model Accuracy Table**

**Testing:**

Machine Learning involves modeling data and predicting outputs using various algorithms. However, with numerous algorithms available, selecting the most appropriate one for predicting the final data can be challenging. Hence, it is important to compare different models and select the one with the highest accuracy.

To assess the accuracy of regression models, metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) are utilized. These metrics can be computed using Scikit-Learn's mean\_absolute\_error and mean\_squared\_error methods.

Graphical user interface, text, application, email

Description automatically generated

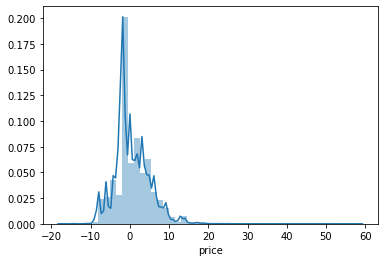
Graphical user interface, application

Description automatically generated

A scatter plot is created to compare the predicted and tested values, and then the errors such as MSE, MAE, and RMSE are calculated. The seaborn library is used to draw a distribution plot of the difference between actual and predicted values. A distribution plot, also known as a distplot, provides a visualization of the overall distribution of continuous data variables.

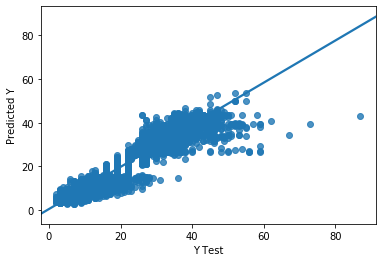
|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **Models** | **metrics** |
| 1​ | Mean Absolute Error | 3.40607721 |
| 2​ | Mean Squared Error | 20.0334370 |
| 3​ | Root Mean Absolute Error | 4.47587277 |

**Error table for Linear Regression**



**Random Forest Model Testing:**

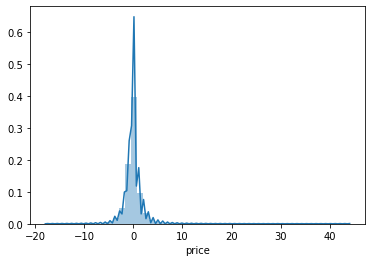
We create a scatter plot and a distribution plot to compare the predicted and tested values in random forest. Furthermore, we calculate the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) for evaluating the accuracy of the model. The implementation of these evaluation metrics can be done using Scikit-Learn's mean\_absolute\_error and mean\_squared\_error methods.



**Scatter Plot for Random Forest**

**Error table for Random Forest**

|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **Models** | **Metrics** |
| 1​ | Mean Absolute Error | 0.99813700 |
| 2​ | Mean Squared Error | 2.94465361 |
| 3​ | Root Mean Absolute Error | 1.71599930 |



**Dist Plot for Random Forest**

**Price Prediction Function:**

A function called "predict\_price" is developed to predict the price using four input parameters: cab name, source, surge multiplier, and icon (weather). As the dataset is trained on continuous values, the categorical values are also passed as integers. A user manual is created to guide users on the input format and sequence for each parameter.

The "predict\_price" function utilizes the random forest model to predict the price based on four input parameters: cab name, source, surge multiplier, and icon (weather). Firstly, the function searches for all the rows in the dataset that correspond to the input cab name and extracts their row numbers. Then, an array "x" of the same length as the new dataset is created, with initial values of zero. The input values of source, surge multiplier, and icon are assigned to their respective indices in the "x" array. Next, the function checks if the count of desired rows is greater than zero. If true, a value of 1 is assigned to the index of the "x" array, and the price is returned using the predict function with the trained random forest algorithm. Additionally, a manual is provided for users to provide instructions about input requirements and sequence.

Graphical user interface, text, application, email

Description automatically generated

**Output: 26.440184991891492**

**Conclusion:**

EDA helps us gain insights into the data before working on features. We visualize the data using various plots, which helps us understand that we don't have data for taxi prices, as well as the price variations for other cabs and different weather types. The value count plots show the type and amount of data in the dataset. Next, we convert all categorical values to continuous data types and fill the NaN values in price with the median of other values. The most crucial part of feature selection is performed using recursive feature elimination (RFE), which selects the top 25 features. After selecting those features, we drop some columns that are deemed unimportant to predict the price, leaving us with eight important columns.

Our analysis involves using four different models on the remaining dataset, out of which Decision Tree, Random Forest, and Gradient Boosting Regressor show the highest accuracy of over 96% when trained with our selected features. This indicates that these three algorithms have high predictive power for the dataset. However, we ultimately decide to use Random Forest as it is less prone to overfitting. With the help of this model, we design a function to predict the price.

**REFERENCES**

M, B. (2019, October 13). *Uber and Lyft Dataset Boston, MA*. Kaggle. Retrieved March 30, 2023, from <https://www.kaggle.com/datasets/brllrb/uber-and-lyft-dataset-boston-ma>

*Machine learning with python - algorithms*. Tutorials Point. (n.d.). Retrieved March 30, 2023, from<https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_algorithms.htm>

Smriti, S. (2023, March 29). *What is mean squared error, mean absolute error, root mean squared error and r squared?* Studytonight.com. Retrieved March 30, 2023, from <https://www.studytonight.com/post/what-is-mean-squared-error-mean-absolute-error-root-mean-squared-error-and-r-squared>

K, D. (2021, April 9). *Implementing gradient boosting regression in python*. Paperspace Blog. Retrieved March 30, 2023, from <https://blog.paperspace.com/implementing-gradient-boosting-regression-python/>

*Sklearn.tree.decisiontreeregressor*. scikit. (n.d.). Retrieved March 30, 2023, from <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>

Abel Brodeurand & Kerry Nield (2018) An empirical analysis of taxi, Lyft and Uber rides: Evidence from weather shocks in NYC.

Junfeng Jiao (2018) Investigating Uber price surges during a special event in Austin, TX

Anna Baj-Rogowska (2017) Sentiment analysis of Facebook posts: The Uber Case.

Anastasios Noulas, Cecilia Mascolo, Renaud Lambiotte, and Vsevolod Salnikov (2014) OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs.