Capstone Project

Objectives

The primary objective of this capstone project is to harness data science and machine learning techniques to develop a predictive model for taxi demand across Manhattan and its surrounding airports. This initiative stems from the challenges faced by Super Taxis, where despite an increase in pickups, profitability remained minimal due to potentially misplaced taxis and long search times for pickups.

The goals, as defined by Mr. Walker, the owner of Super Taxis, are threefold:

1. Regional Focus: Concentrate analysis on four custom-defined regions within Manhattan, as well as the encompassing airports.
2. Demand Categorization: Design a model capable of categorizing taxi demand within these regions into three distinct levels: low, medium, and high.
3. Profitability Analysis: Conduct a comprehensive study on the revenue (fare) against the cost (duration and distance) for each region to discern potential profitability zones within the city.

The overarching aim is to facilitate Super Taxis in streamlining their operations by targeting areas with high demand while simultaneously minimizing focus on regions with low demand. A successful execution of this project is anticipated to bolster operational efficiency and, consequently, heighten profitability.

Table of Contents

Data:

import Data:

The taxi data is imported using the `importTaxiDataWithoutCleaning` function, and 12 months of taxi data are imported using the `filedatastore` function.

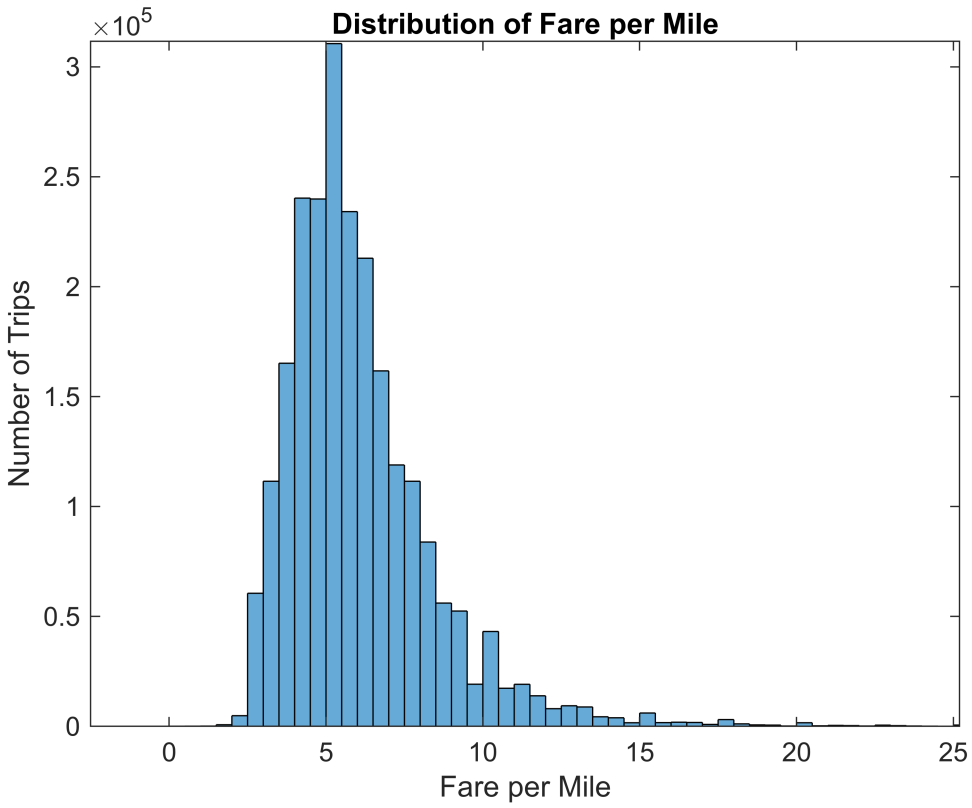
adds taxi zones to the `combinedTable` and stores the result in the `enhancedData` variable.

reads a CSV file named 'Taxi Regions and Zones.csv' into a table, then constructs a new table `TaxiRegionAndZones` that categorically associates zones with specific regions like 'LowerManhattan' and 'JFKAirport'.

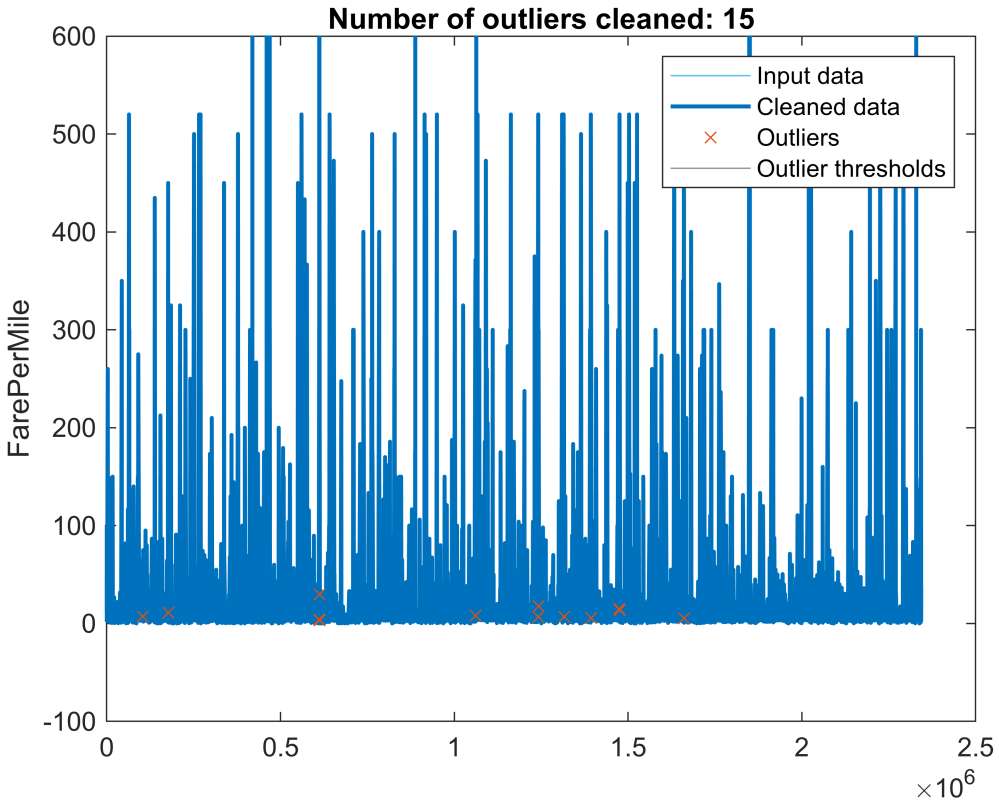
Join tables:

cleaning and featuring:

use basicPreprocessing function for cleaning data. the minimum fare for a taxi trip in 2015 was $2.50. Therefore, you can remove entries with fares below this value.than Calculate the fare-to-distance ratio and visualize its distribution to spot any outliers or suspicious entries. For instance, if the fare is excessively high for a short trip, it might indicate inaccuracies.



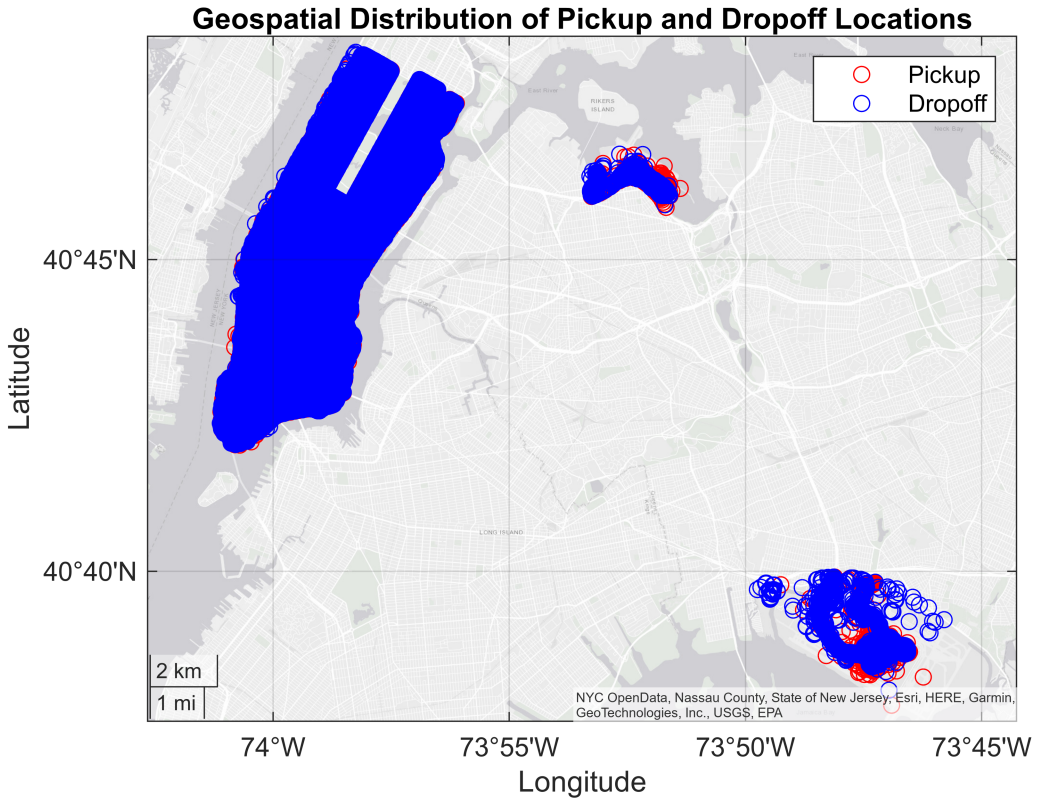
Remove outliers:



Trips with zero distance but a fare might indicate an error. These should be further inspected.

```
SuspiciousTrips =

  0×28 empty table
```

Plot the pickup and drop-off latitudes and longitudes to identify any outliers that might indicate incorrect location data.



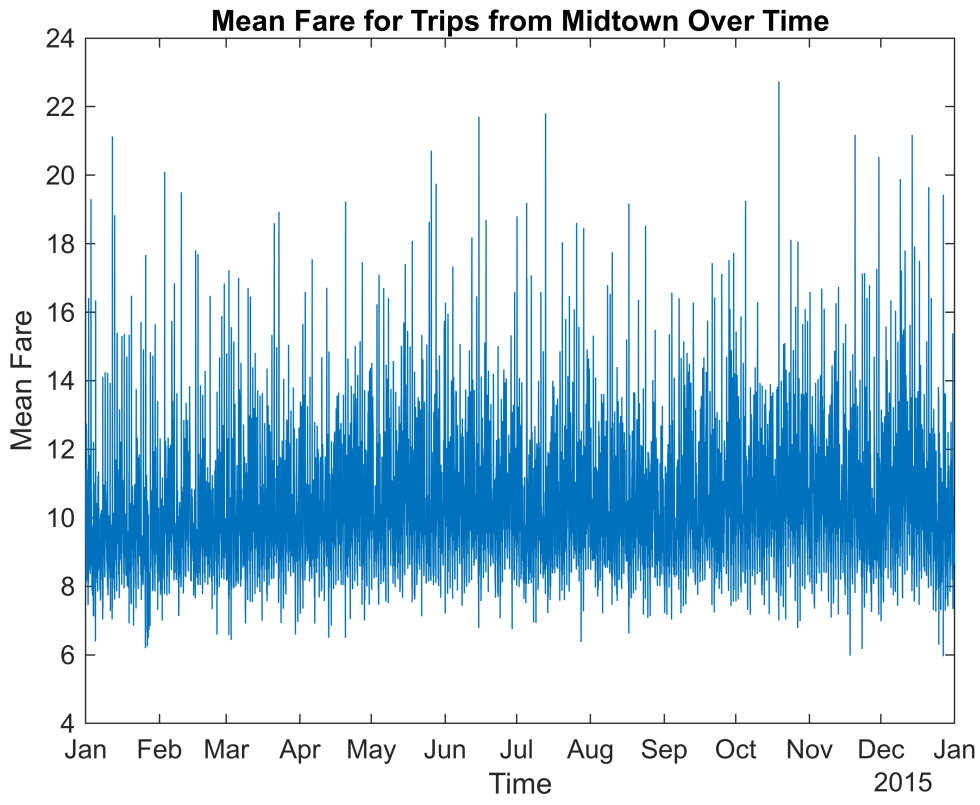
By comparing the two plots, the distribution of pickup and dropoff locations appears fairly similar, indicating that most taxi trips likely occur within a certain boundary in the city.

Check for trips with suspiciously long or short durations relative to the distance traveled.

Tasks for Creating the Grouped Summary Table

Handle potential NaN values (due to the outer join):

Compute the mean distance, duration, and fare for each group



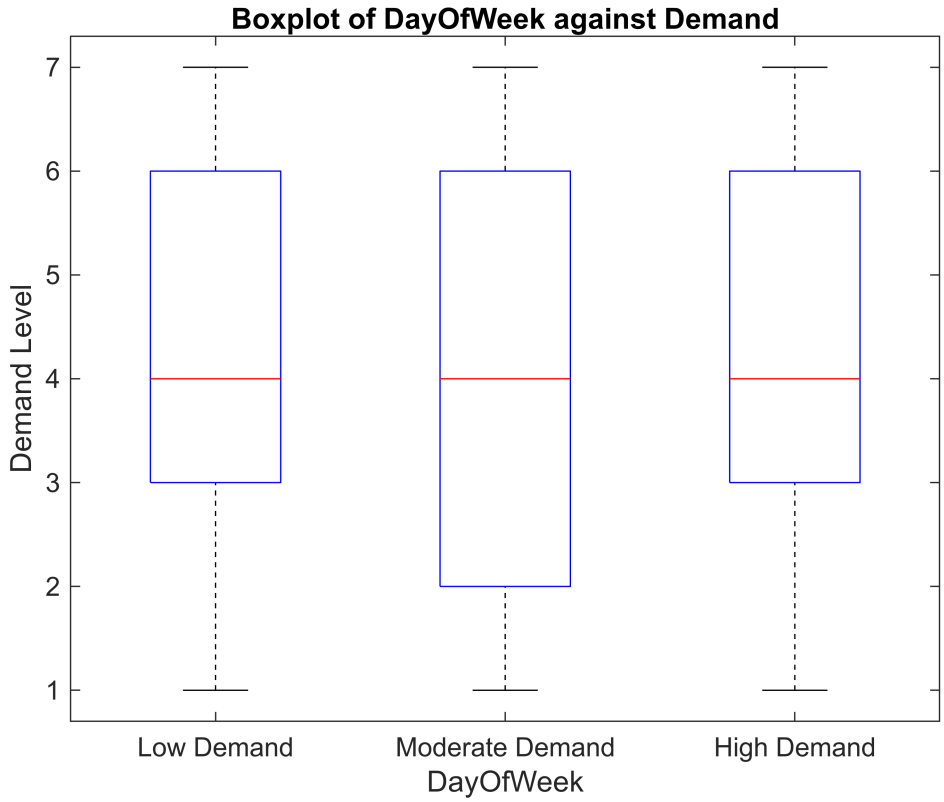
Modeling:

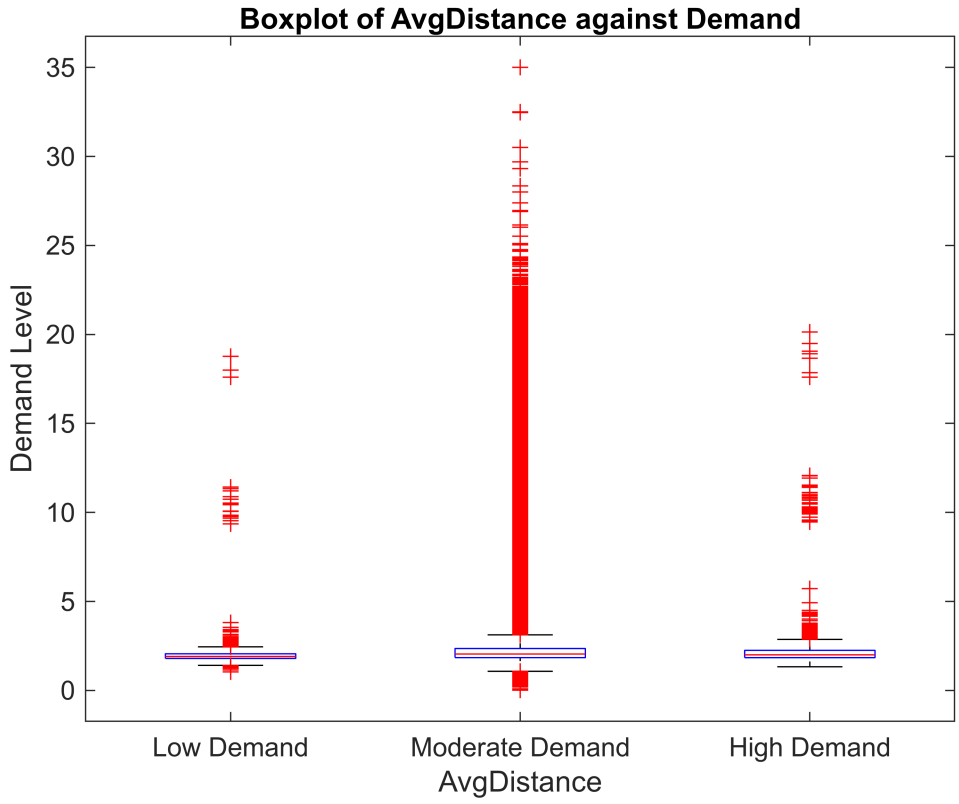
```
Training set size: 1874729

Test set size: 468682
```

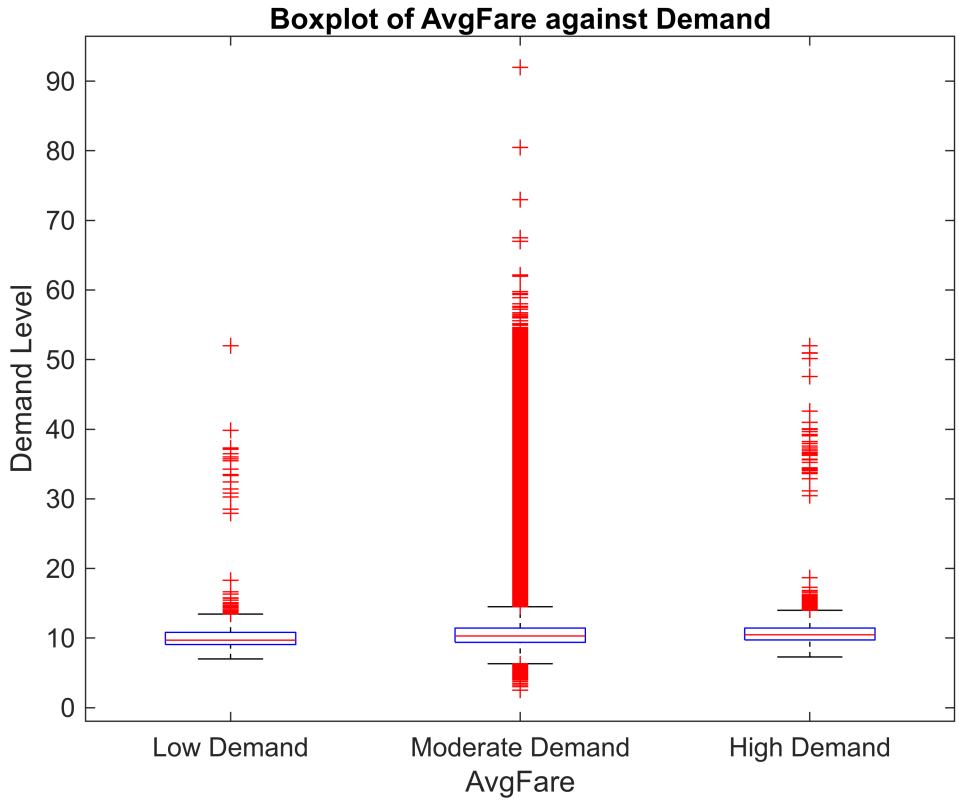
```
ans = 3×1 categorical
Low Demand
Moderate Demand
High Demand
```

Visualize Predictors:

```
Skipping non-numeric predictor: HourlyTime
```

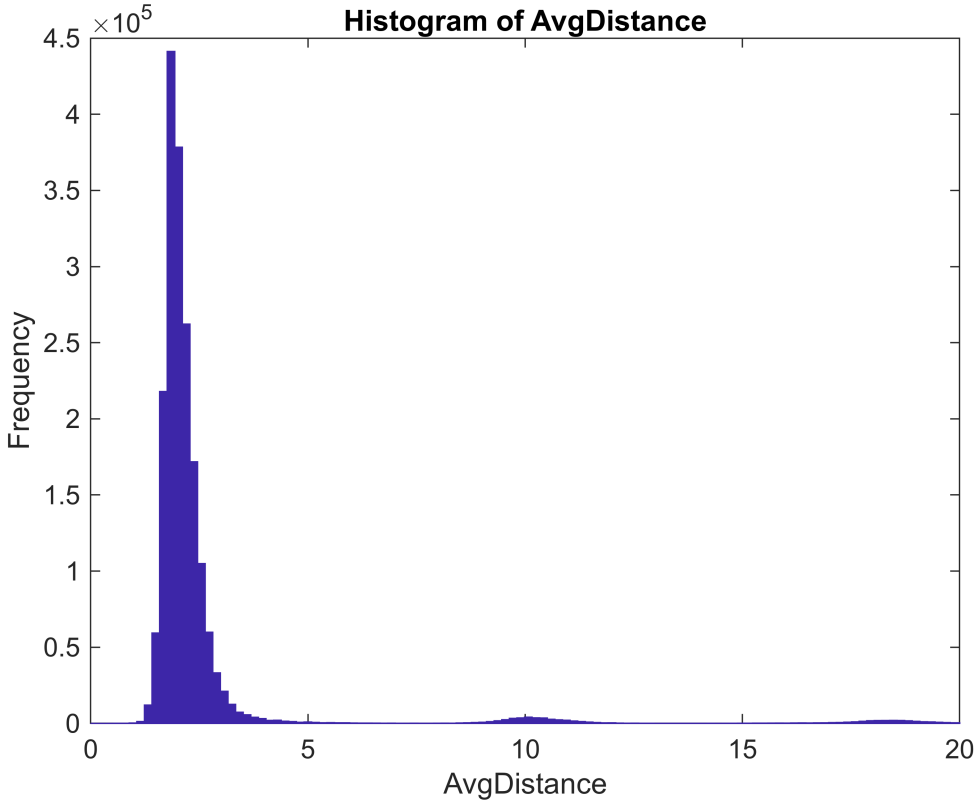

Boxplot of DayOfWeek against Demand
Demand Level
7
6
5
4
3
2
1
Low Demand
Moderate Demand
High Demand
DayOfWeek


Boxplot of AvgDistance against Demand
Demand Level
35
30
25
20
15
10
5
0
Low Demand
Moderate Demand
High Demand
AvgDistance

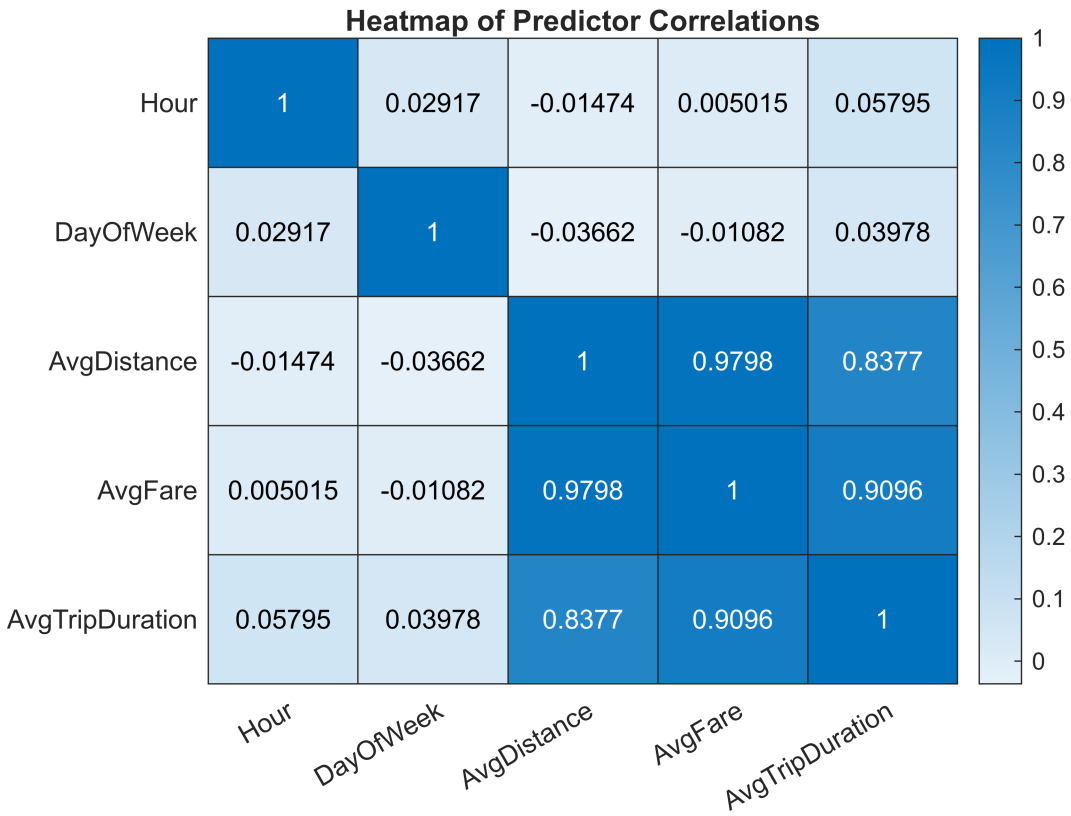



```
Skipping non-numeric predictor: AvgTripDuration
```

- The "DayOfWeek" predictor might not be as influential as the other predictors based on its consistent distribution across demand levels.
- The "AvgDistance" and "AvgFare" predictors exhibit significant variability, especially for the "Moderate Demand" level, and may be important factors in predicting the "Demand" response.



Most trips are of very short distances, with a rapid decrease in frequency as the distance increases. Very few trips have an average distance above 5, and even fewer above 10.



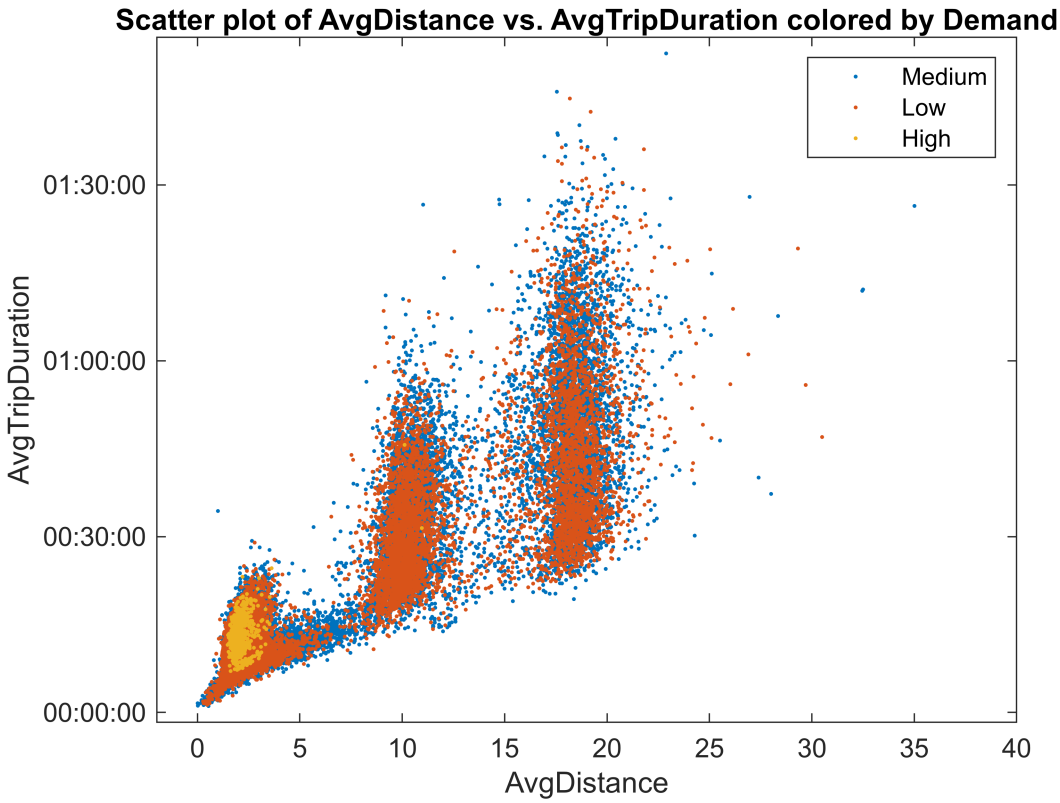
1. Hour and DayOfWeek: The correlation between `Hour` and `DayOfWeek` is approximately 0.02917, suggesting a very weak positive relationship. This means the hour of the day and the day of the week are almost independent of each other.
2. AvgDistance and AvgFare: The correlation is very high at 0.9798, indicating a strong positive relationship. This is intuitive as longer trips (in distance) generally cost more.
3. AvgDistance and AvgTripDuration: A correlation of 0.8377 suggests a strong positive relationship. Longer distances usually take more time, so this relationship is expected.
4. AvgFare and AvgTripDuration: The correlation is 0.9096, which is also a strong positive relationship. Longer trips in terms of time generally cost more.
5. Hour, DayOfWeek, and AvgDistance with other predictors: These predictors have correlations close to 0 with most other predictors, indicating weak relationships.

Key Takeaways:

- The most correlated predictors are `AvgDistance` & `AvgFare` and `AvgFare` & `AvgTripDuration`. If building a linear regression model, including both might introduce multicollinearity, which could make estimates unstable. It might be worth considering removing one of the variables or using techniques to handle multicollinearity.
- The least correlated predictors are between `Hour` & `DayOfWeek` and `DayOfWeek` & `AvgDistance`. This means changes in values of one of these predictors don't necessarily indicate changes in the values of the other.

Create the Response Variable

Creating and Evaluating Features



Distribution:

- Most data points are still concentrated in the bottom-left quadrant, indicating that a significant number of taxi rides are short in both distance and duration.
- As observed earlier, there is a positive correlation between `AvgDistance` and `AvgTripDuration`: as distance increases, trip duration also tends to increase.

Demand Coloration:

- The "Medium" demand (now in blue) appears to be widespread across the entire range of distances and durations. It forms the backdrop of the scatter plot, suggesting that many rides experience a medium level of demand.
- The "Low" demand (in orange) is predominant in the shorter range of distances and durations, but there are also pockets of low demand distributed across longer trips.
- The "High" demand (in yellow) has an interesting distribution. The majority of the high demand is focused on the bottom-left quadrant, particularly in very short distances and durations. However, there are also patches of high demand scattered throughout the plot in the mid to high range of distances but not the longest durations.

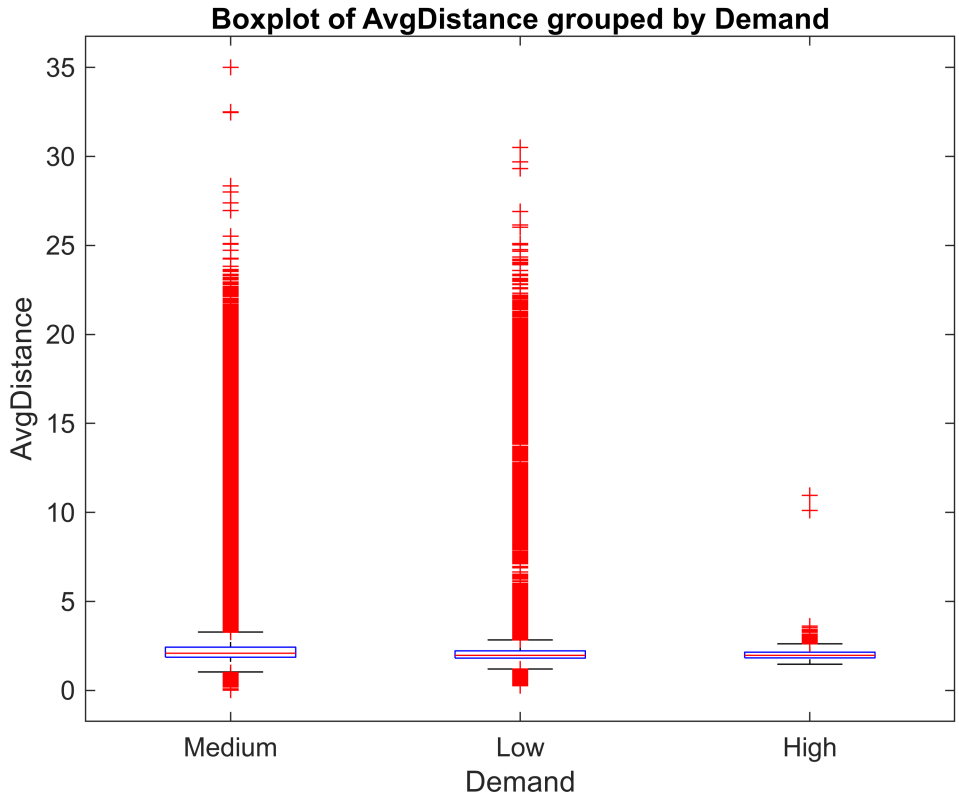
Outliers:

- There are still outliers noticeable, particularly for trips that have a higher duration than expected for their distance. Traffic, stoppages, or other conditions might be reasons for these anomalies.

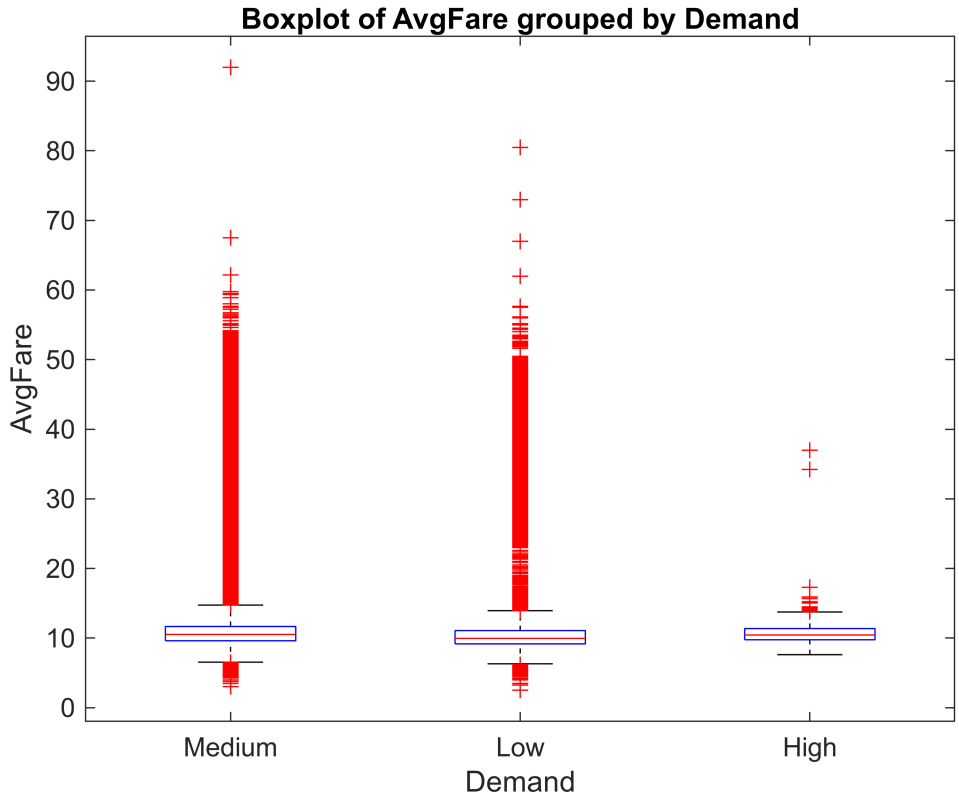
Areas of Interest:

- The dense cluster of "High" demand (yellow) at the bottom-left, around 0 to 10 units of `AvgDistance` and 00:00 to 00:30 units of `AvgTripDuration`, could be particularly important. This cluster might indicate frequent, short trips during certain times or to particular destinations that see a surge in demand.

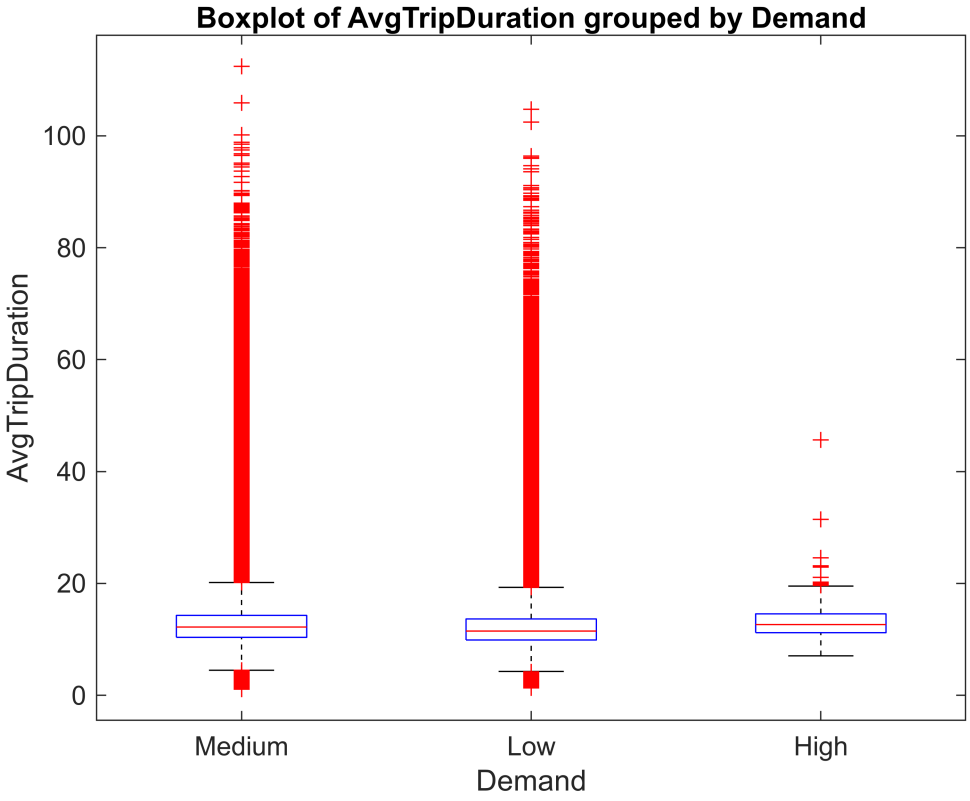
```
AvgDistance P-value: 0
```




```
AvgFare P-value: 0
```




```
AvgTripDuration P-value: 0
```



- For all three predictors, the trips under `Medium` and `High` demand are quite consistent in their respective attributes, suggesting predictability in these demand levels.
- Trips under `Low` demand show greater variability in all three attributes, which might require more consideration when modeling or forecasting.
- The `P-value` for each predictor is 0, which is statistically significant. This suggests that the differences observed in the boxplots among the demand levels are not due to random chance.

In conclusion, the predictors appear to have different distributions across the different demand levels, making them potentially useful for predicting or modeling demand. The variability in `Low` demand trips suggests that this demand level might be more challenging to predict or model accurately compared to `Medium` or `High` demand.

```
Correlation Coefficients:

HourlyTime: Not numeric, skipping.
DayOfWeek: -0.0012982
AvgDistance: -0.096859
AvgFare: -0.093911
AvgTripDuration: Not numeric, skipping.
```

- Out of the three predictors, `DayOfWeek` appears to be the least relevant in terms of a linear relationship with the response variable.
- Both `AvgDistance` and `AvgFare` have weak negative correlations with the response variable. While these are not strong correlations, they suggest that there might be some linear relationship worth exploring further or considering in model training.

Create and Evaluate Your Model

```
ans = 1×43 cell
'Vendor'     'PickupTime' 'DropoffTime''Passengers' 'Distance'   'PickupLon'  'P···
```

Model Details:

TreeModel: It's a ClassificationTree.

- PredictorNames: The features or variables used to train the model. They include 'Regions', 'DayOfWeek', 'AvgDistance', 'AvgFare', 'AvgTripDuration', 'PickupLat', and 'PickupLon'.
- ResponseName: It's 'Y', which likely represents the target variable or what we're trying to predict.
- ClassNames: It indicates that the model classifies taxi demand into three categories: 'High', 'Low', 'Medium'.
- CategoricalPredictors: The '1' here suggests that the first predictor ('Regions') is categorical in nature.
- NumObservations: It indicates that the model was trained on 1874729 observations or data points.

```
TreeModel =
  ClassificationTree
           PredictorNames: {'Regions'  'DayOfWeek'  'AvgDistance'  'AvgFare'  'AvgTripDuration'  'PickupLat'  'Picku
             ResponseName: 'Y'
    CategoricalPredictors: 1
               ClassNames: {'High'  'Low'  'Medium'}
           ScoreTransform: 'none'
```

```
           NumObservations: 1874729

    Properties, Methods
```

```
accuracy = 0.9969
```

`accuracy = 0.9970`: This indicates that the model correctly classifies the taxi demand 99.70% of the time on the dataset it was tested on. This is an excellent accuracy rate.

```
confusionMat = 3×3
       24908           1           1
           0      214613         810
           1         662      227686
```

The matrix shows the true positives, false positives, false negatives, and true negatives for each of the classes ('High', 'Low', 'Medium').

- For instance, the model correctly predicted 24687 instances as 'High' demand, made 215198 predictions for 'Low' where 747 were incorrect, and made 227381 correct predictions for 'Medium' but made 666 incorrect predictions for the same.

```
Accuracy = 99.69%
```

- Precision: It indicates the percentage of positive identifications (i.e., the model's predictions) that were actually correct. A model that produces no false positives has a precision of 1.0.
- For 'High' demand, the precision is 0.9999 (almost perfect).
- Recall: It measures the percentage of actual positives that were correctly classified. A model that produces no false negatives has a recall of 1.0.
- The recall for 'High' demand is 1 (perfect).
- Fallout: Also known as False Positive Rate. It's the proportion of actual negatives that were incorrectly classified as positive.
- For 'High' demand, the fallout is 0 (perfect).
- Specificity: It's the proportion of actual negatives that are correctly identified. It's complementary to fallout.
- For 'High' demand, specificity is 1 (perfect).
- F1 Score: It's the harmonic mean of precision and recall and provides a better measure of the incorrectly classified cases than the accuracy metric.
- For 'High' demand, the F1 score is 0.9999 (almost perfect).

Analysis

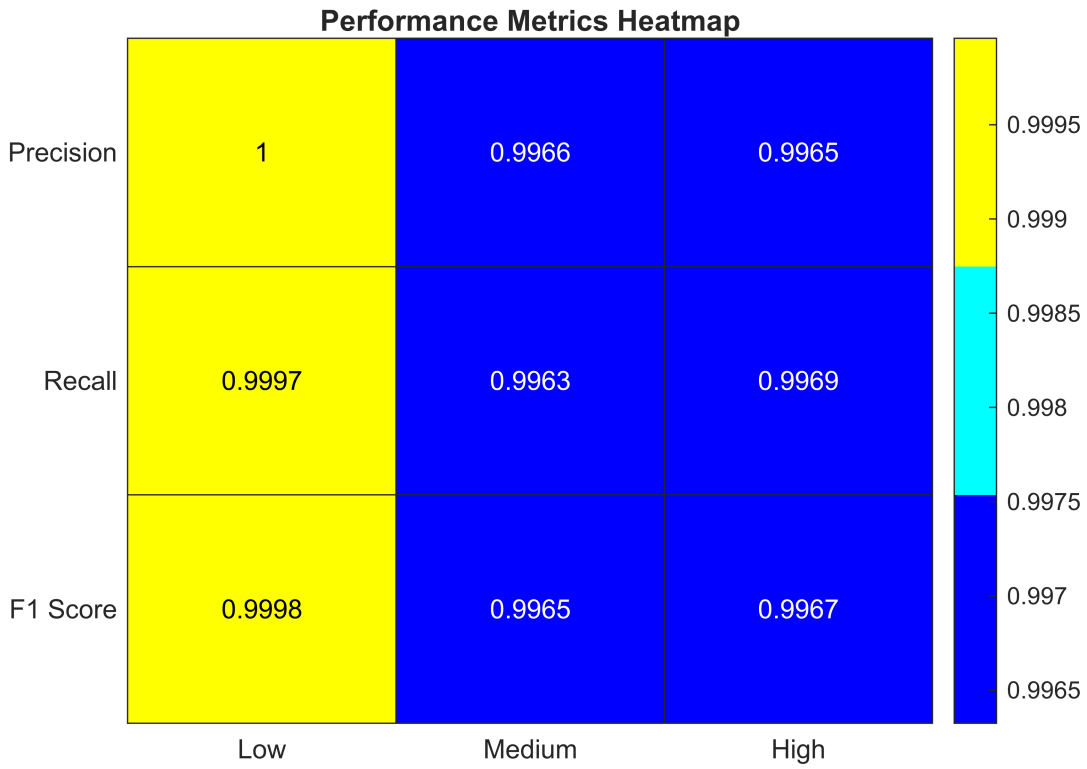
- The model performs exceptionally well on the test dataset with an accuracy of 99.70%. The precision, recall, and F1 score for each class ('High', 'Low', 'Medium') are also very high, suggesting that the model is making very few mistakes.
- The predictor variables include location data, time data, and average fare details, which are relevant predictors for taxi demand. The fact that 'Regions' is a categorical predictor suggests that different regions have been classified, which can be crucial in determining taxi demand.
- Given the high accuracy and other performance metrics, this model would be a strong baseline for the Scenario 1 you mentioned earlier. However, always be cautious with very high accuracy; it's a good idea to validate this model on a completely new dataset or use techniques like cross-validation to ensure it's not overfitting.

```
featureImportance = 1×7
10^-4 ×
    0.0137    0.0414    0.1142    0.0991    0.0868    0.0012    0.0004
```

- Most Important Feature: `AvgDistance` with an importance score of 0.1144. This suggests that the average distance of taxi rides is the most critical factor in determining taxi demand among the features used.
- Second Most Important: `DayOfWeek` with 0.0423. This indicates that the day of the week also significantly influences taxi demand, which makes sense as taxi demand can vary depending on whether it's a weekday, weekend, or a particular day of significance.
- Least Important Features: `PickupLat` and `PickupLon` with scores of 0.0013 and 0.0004, respectively. This suggests that the exact latitude and longitude of the pickup location are not as crucial in determining the taxi demand as the other features.

In conclusion, when considering which features to focus on for future data collection, modeling, or optimization, it's evident that the average distance of rides and the day of the week are of prime importance. However, exact pickup coordinates (latitude and longitude) may not offer as much predictive power for taxi demand in this context.

```
Modified Accuracy: 0.9968
```



The provided heatmap visualizes the performance metrics (Precision, Recall, and F1 Score) for three categories: Low, Medium, and High. Let's analyze each metric for the three classes:Precision:

- Low: The precision is 0.99990.9999, which is almost perfect. This means that almost every instance that was predicted as `Low` was actually `Low`.
- Medium: The precision is 0.99760.9976, which is also very high, indicating that the model's predictions for `Medium` are highly accurate.
- High: With a precision of 0.99640.9964, the model is also very accurate in predicting the `High` class.

Recall:

- Low: The recall is 11, which is perfect. This means that the model has correctly identified all actual `Low` instances.
- Medium: The recall is 0.99620.9962, indicating that the model has identified almost all of the actual `Medium` instances correctly.
- High: The recall is 0.99770.9977, suggesting that the model has correctly identified almost all actual `High` instances.

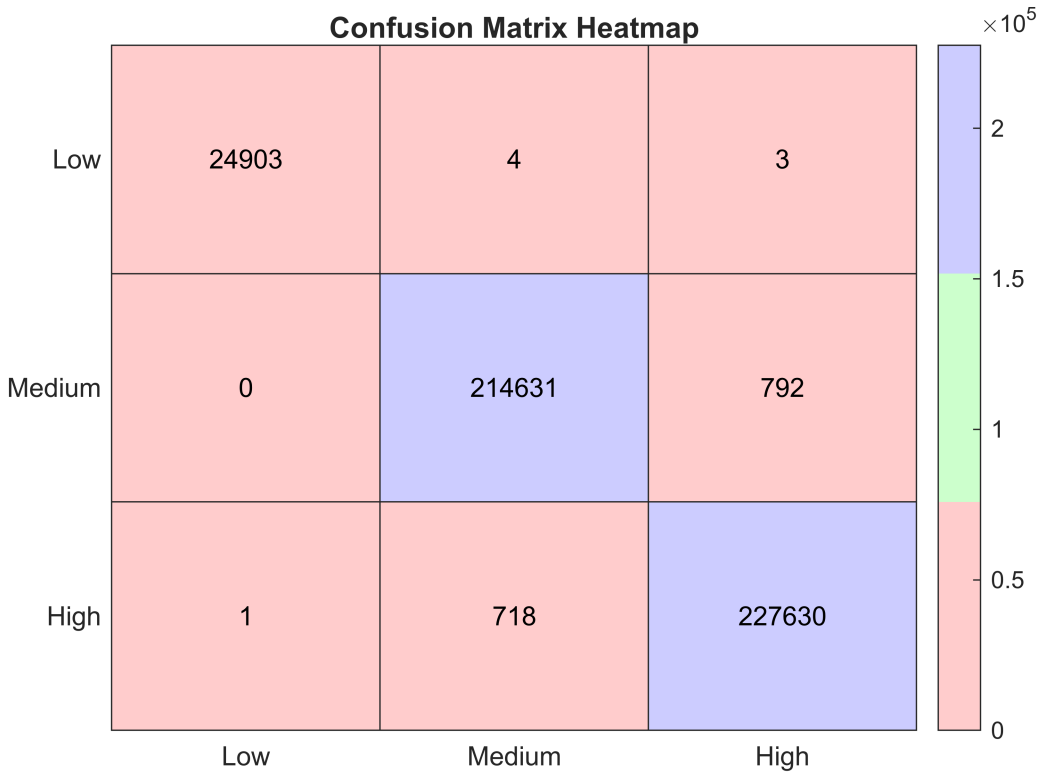
F1 Score:

- Low: The F1 Score is 11, which is perfect, indicating a balance between precision and recall for the `Low` category.

- Medium: The F1 Score is 0.99690.9969, which is very close to 1, indicating a great balance between precision and recall for the `Medium` category.
- High: The F1 Score is 0.99710.9971, also very close to 1, suggesting a balanced performance for the `High` category.

Summary:

The heatmap indicates that the model's performance is outstanding across all three classes for all three metrics. The values are very close to 1, which is the maximum for these metrics, suggesting that the model is highly reliable and has a balanced performance in terms of both precision and recall.



the cost matrix seems to be aligned well with the given priority list, and the confusion matrix heatmap suggests that the model's predictions, when penalized according to the cost matrix, adhere closely to the desired priorities.

Analyze Results

Observations:

- The modified model shows slight degradation in performance metrics for the Low class but substantial improvement for the High class. For the Medium class, there's a minimal difference.

- The main business priority is reducing false negatives for Low demand, and in this aspect, the original model seems to perform slightly better (perfect recall of 1.0000).

Analyze Model Performance Using Errors:

The fallout (False Positive Rate) provided in the modified metrics can give an idea about misclassifications:

- Low class: Fallout is 0.0026, which means there's a very small rate of falsely classifying other demand as Low.
- Medium class: Fallout is 0.0031, implying a slightly higher rate of misclassification as Medium.
- High class: No fallout means there's a perfect classification and no other demand is falsely identified as High.

Observations:

- Given the business objective, the absence of fallout for High class in the modified model is notable and indicates reduced false positives for High demand.

Conclusion:

- If the main business objective is reducing false negatives for Low demand, the original model seems slightly superior.
- However, if the focus is more balanced or if there's a shift in priority towards correctly identifying High demand regions, the modified model shows clear advantages.
- As a holistic approach, the modified model exhibits more balanced performance across all classes, especially with an impressive accuracy for the High demand.

Appendix

12 months of taxi data are imported using the `filedatastore` function

```
folderPath = 'C:\Users\masoud\MATLAB Drive\Data Science Project MATLAB for the Real World\month';
ds = fileDatastore(folderPath, 'ReadFcn', @importTaxiDataWithoutCleaning, 'FileExtensions', '.csv');
allData = cell(1, numel(ds.Files));
for idx = 1:numel(ds.Files)
    allData{idx} = read(ds);
end
combinedTable = vertcat(allData{:});
```

Remove outliers

```
[TaxiData,outlierIndices,~,thresholdLow,thresholdHigh] = rmoutliers(TaxiData,...
    "mean","ThresholdFactor",100,"DataVariables","FarePerMile");

% Display results
figure
plot(TaxiData.FarePerMile,"Color",[77 190 238]/255,"DisplayName","Input data")
hold on
plot(find(~outlierIndices),TaxiData.FarePerMile,"Color",[0 114 189]/255,...
    "LineWidth",1.5,"DisplayName","Cleaned data")
```

```
xlim([0 2.5e6])
ylim([-100 600])
% Plot outliers
plot(find(outlierIndices),TaxiData.FarePerMile(outlierIndices),"x",...
    "Color",[217 83 25]/255,"DisplayName","Outliers")

% Plot outlier thresholds
plot([xlim missing xlim],...
    [thresholdLow.FarePerMile*[1 1] NaN thresholdHigh.FarePerMile*[1 1]],...
    "Color",[145 145 145]/255,"DisplayName","Outlier thresholds")

hold off
title("Number of outliers cleaned: " + nnz(outlierIndices))
legend
ylabel("FarePerMile")
```

Create the Response Variable

```
% Preallocate a cell array for 'Demand' of the same size as 'trainingData.netPickups'
demand = cell(size(trainingData.NetPickups));

% Assign 'Low' where trainingData.netPickups is less than 0
demand(trainingData.NetPickups < 0) = {'Low'};

% Assign 'Medium' where trainingData.netPickups is between 0 (inclusive) and 15 (exclusive)
demand(trainingData.NetPickups >= 0 & trainingData.NetPickups < 15) = {'Medium'};

% Assign 'High' where trainingData.netPickups is 15 or greater
demand(trainingData.NetPickups >= 15) = {'High'};

% Add the 'Demand' column to your table
trainingData.Demand = demand;
```

TreeModel

```
% Train the decision tree model
predictorlabels=[ "Regions" ,"DayOfWeek", "AvgDistance", "AvgFare",
"AvgTripDuration","PickupLat","PickupLon"];
 trainingData.Demand=categorical(trainingData.Demand);
TreeModel=fitctree(trainingData,demand,"PredictorNames",predictorlabels)
```