Member 1: 20I-0621 Farquleet Farhat

Member 2: 20I-0441 Shehreyar Ramzan

Section: B

Assignment: Project 1

REPORT

Code Summary:

The code reads weather data and extracts timestamp, weather, and temperature. This data is then appended to a list called weather_list. Similarly, the code reads data related to clusters and extracts information such as regoin_hash and regoin_id, and appends it to a list called cluster_list.

The code then reads data related to orders and extracts order_id, driver_id, start_location, end_location, price, and timestamp. It then rounds off the timestamp to the nearest 5-minute interval and uses this rounded-off time to match with the data from the weather_list. The weather and temperature information from the weather_list is then appended to the order details and stored in a list called order_list.

It then proceeds to count the number of unfilled orders at every 10-minute interval of the day and stores this information in a list called count_list. Finally, the code sorts the unfilled orders according to their start_location_id and end_location_id and stores them in a list called region_sort.

Next, the code extracts the predictor and response variables from the dataset and sets the window size for the time series analysis. It then initializes two lists to store the performance metrics for each window: R-squared and mean squared error.

The code then loops through the data in overlapping windows of the size (200) and fits a regression model to each window. The code includes three regression models that can be used: linear regression, decision tree regression, and random forest. The chosen regression model is fit to the predictor and response variables in the current window, and the model is then used to

make predictions on the data in the window. The R-squared and MSE are calculated for the current window, and the performance metrics are appended to the lists.

Finally, the code prints the summary of the fitted model for each window, including the intercept, coefficients, and R-squared. It also prints the second predictor variable and the regression coefficients. Additionally, it prints the mean squared error for the current window.

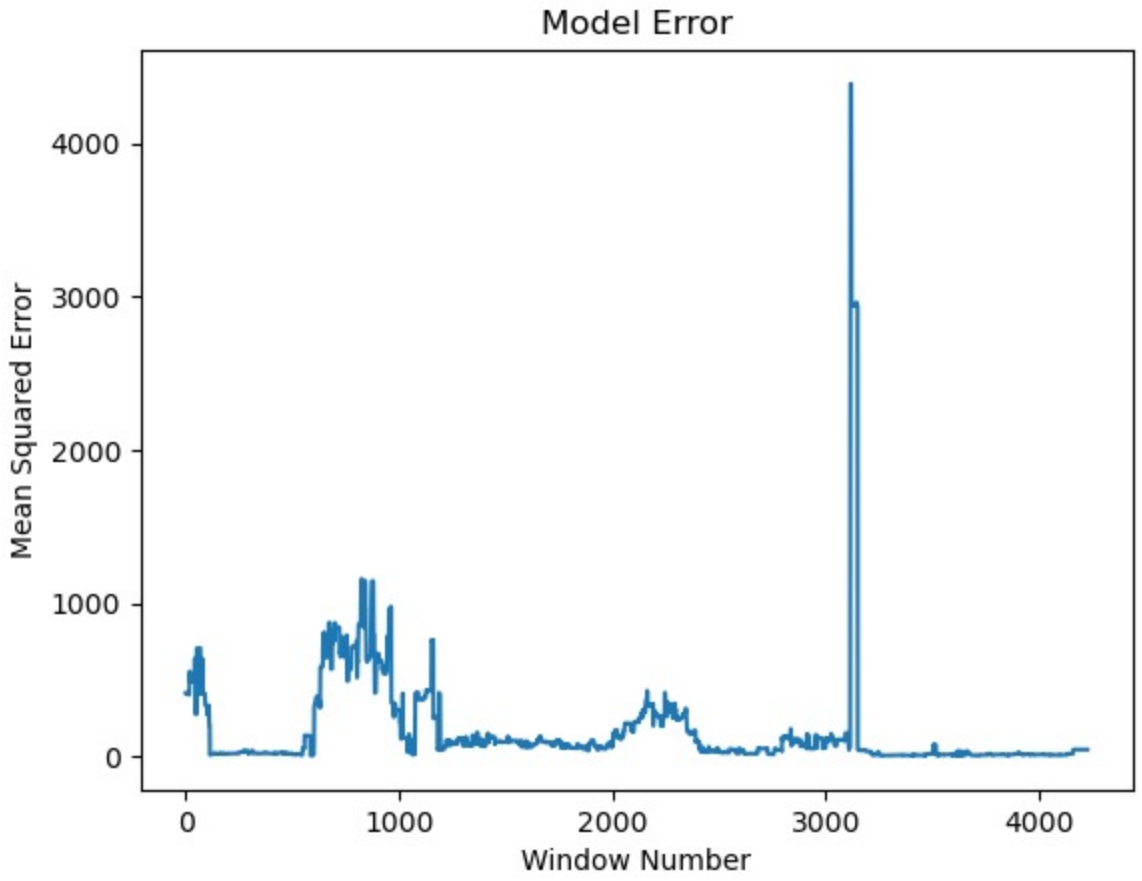
Additionally, other regression models could be tested to find the best-fit model for the data. Finally, the code could be modified to predict the gap variable for future time periods.

Experience:

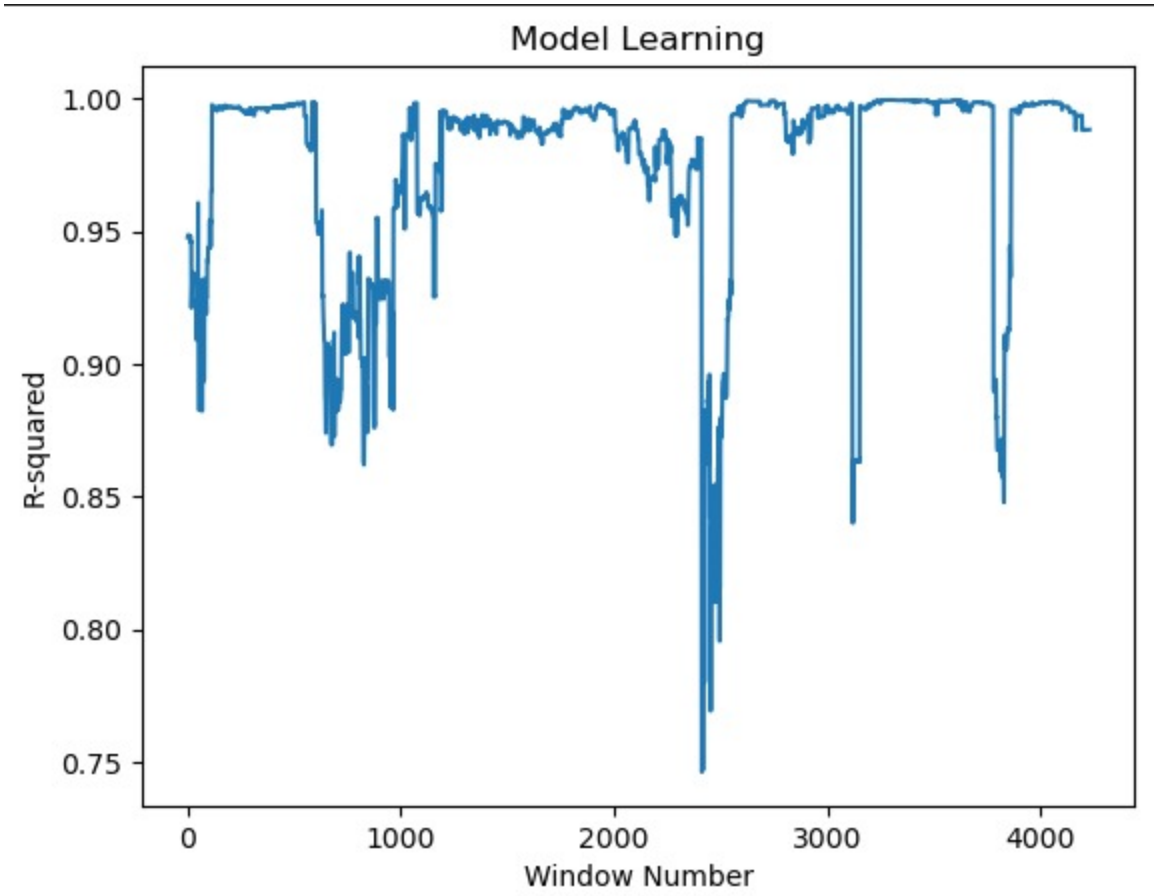
During the project, I thoroughly enjoyed working with machine learning and found it to be a lot of fun. However, I did face some challenges along the way. One of the biggest issues I faced was when I had to merge the poi class data. The instructions provided were insufficient, and I had to spend a lot of time trying to figure out the correct approach but no result. Additionally, the size of the data was quite large, and it was almost impossible to train the model using all the training data sets. I then tried to calculate the error using the given mean absolute error, but found it to be quite time-consuming. Eventually, I decided to use the inbuilt function for error calculation, which made things a lot easier. Despite these challenges, I learned a lot during this project and had a great time working on it.

Graphs:

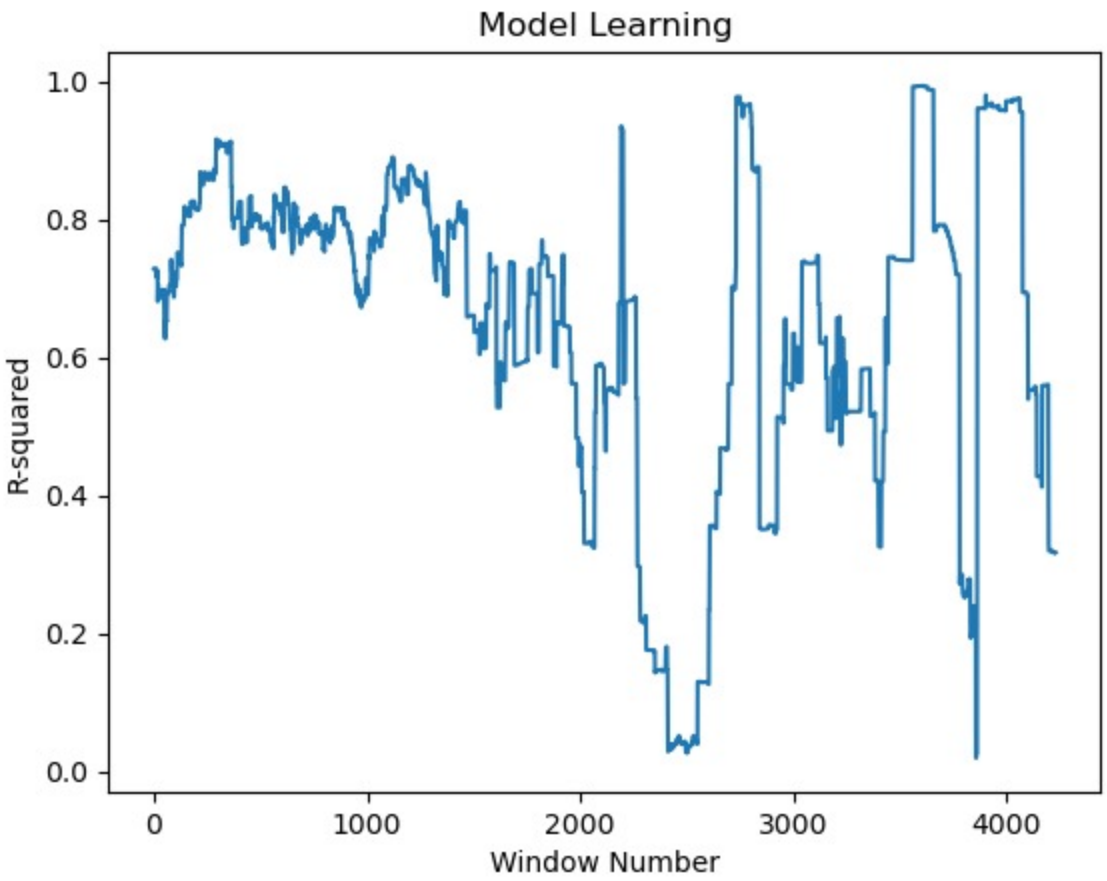
Model == Decision Tree

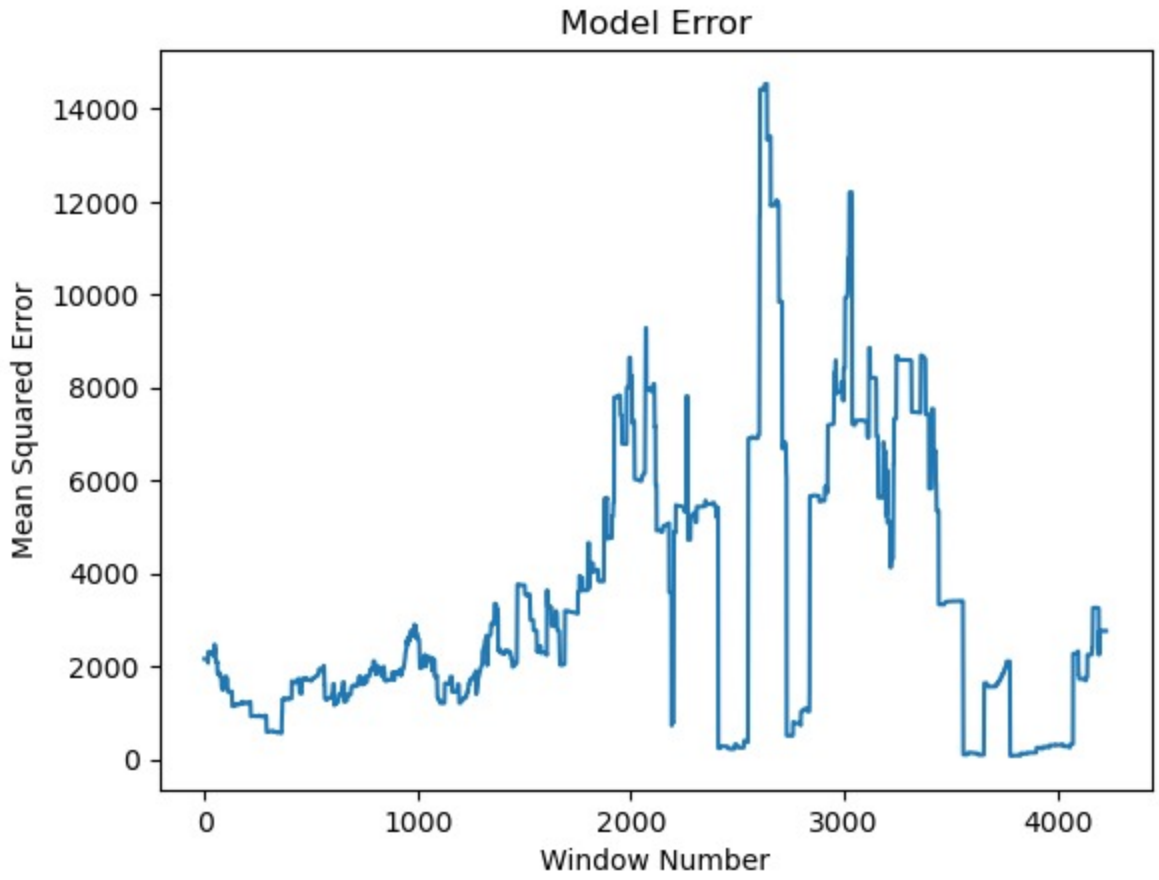


One of the major drawbacks of using Decision Tree as a machine learning algorithm is its tendency to overfit the data. This means that the model fits too closely to the training data and may not generalize well to new, unseen data. This issue can be clearly observed in the figure where the error is consistently zero for most of the data points. As a result, the model may not be able to accurately predict outcomes for new data points outside the training set.

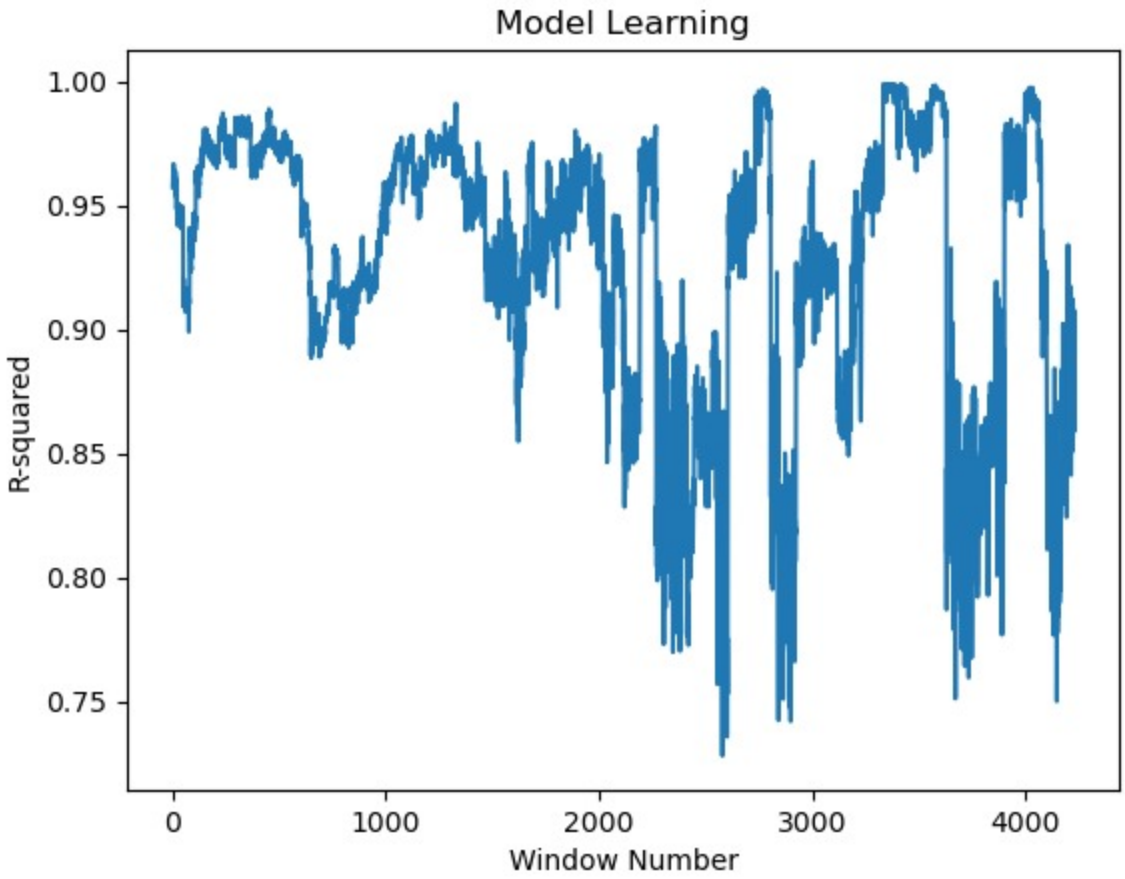


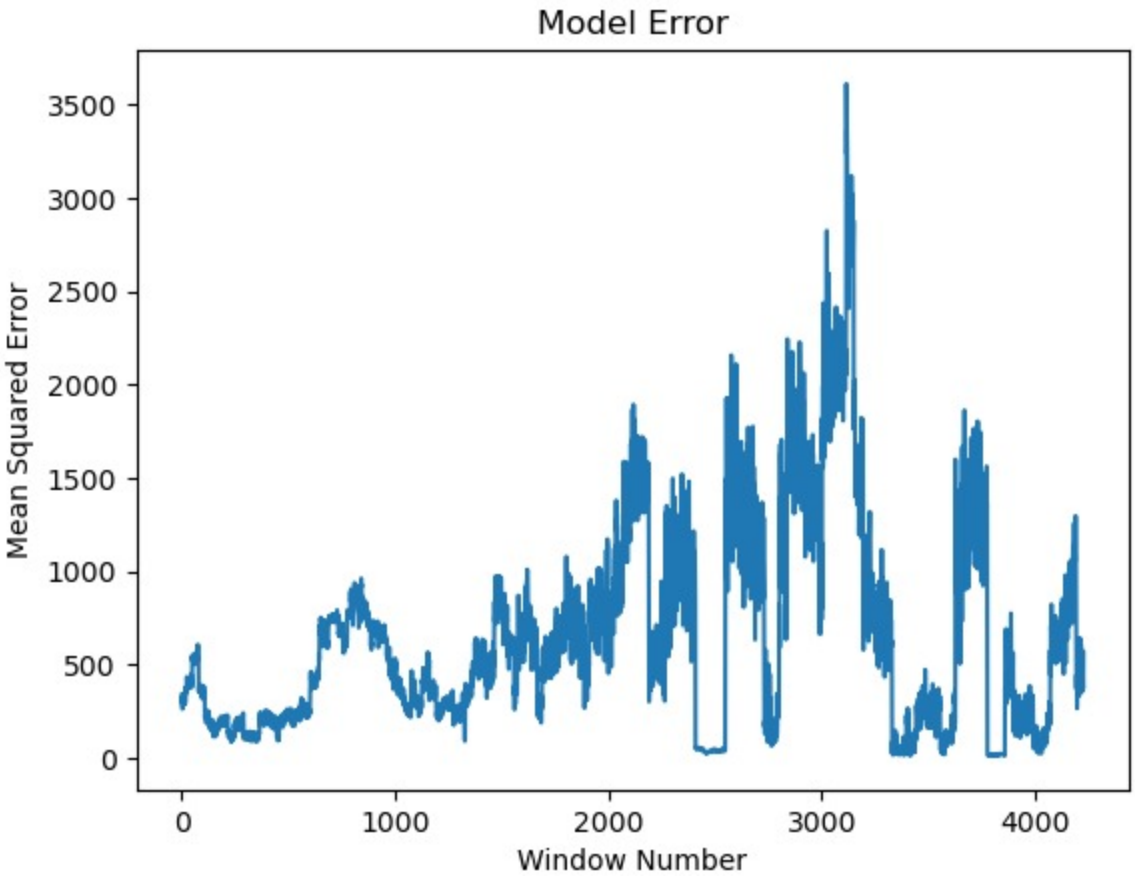
Model == Linear Regression


Model Learning
R-squared
1.0
0.8
0.6
0.4
0.2
0.0
0
1000
2000
3000
4000
Window Number



Model == Random Forest


Model Learning
1.00
0.95
0.90
0.85
0.80
0.75
R-squared
0
1000
2000
3000
4000
Window Number


Model Error
Mean Squared Error
Window Number
0
500
1000
1500
2000
2500
3000
3500
0
1000
2000
3000
4000