Project Report

AI Project Part 1

Muhammad Rohaan Atique (20I-0410)
Ahmed Moiz (20I-2603)

26th April, 2023

Introduction

In this project, we were given data of multiple features from an uber type application, and we had to apply pre-processing to the data and derive actual and predicted gaps (i.e. demand-supply) of different regions.

I used two different models to gauge the prediction capabilities of the models, and both had different results.

Model 1: Linear Regression

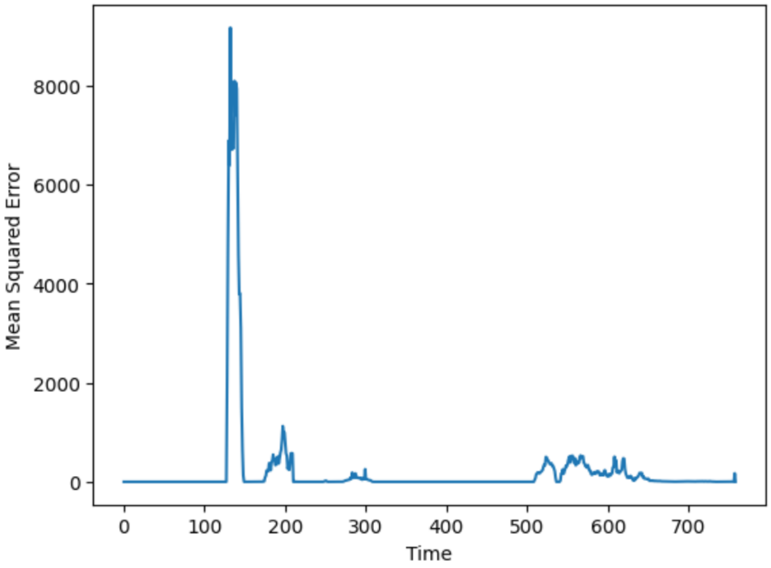
In this model, we used the feature set

```
[['start_district_hash', 'dest_district_hash', 'Price', 'weather',
'temperature', 'PM2.5']]
```

With the independent variable being the Gap column.

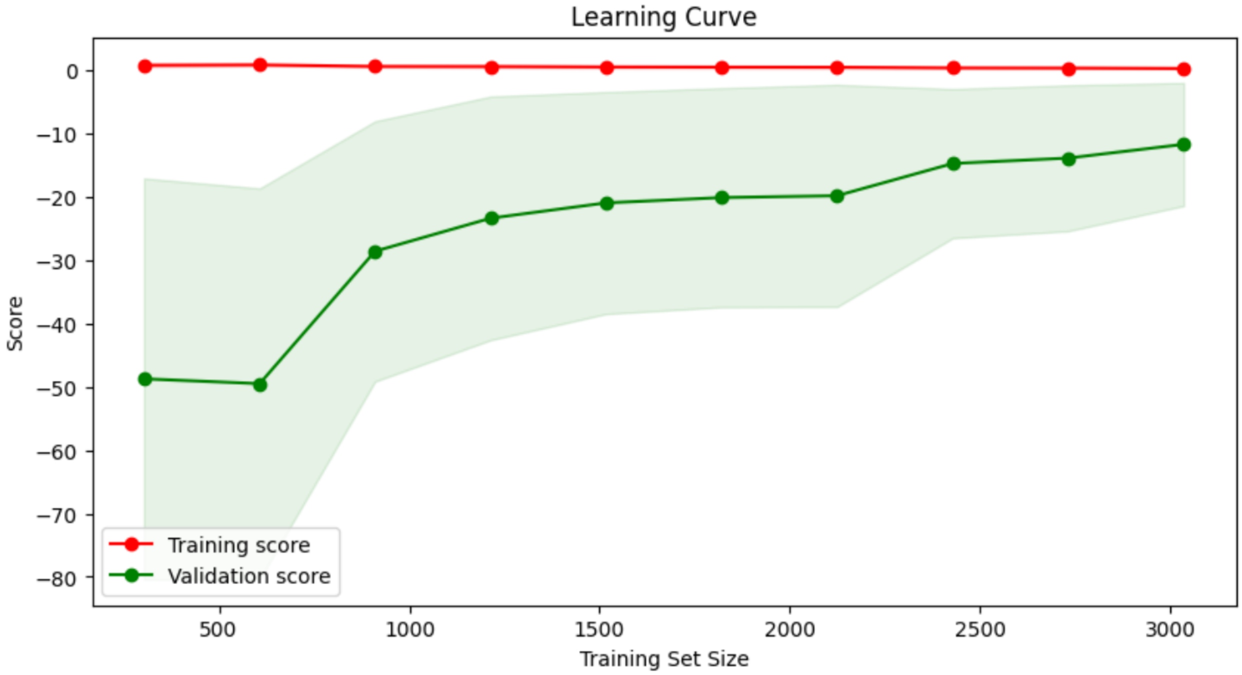
Using the iloc feature of pandas, we made a rolling window of size 200 with a shift of 2 rows every iteration.

On each iteration of the rolling window, the error of the model was noted using sklearn's MSE function, where the plot looked like this. (findings on next page)



It is clear that the error is fluctuating as the window rolls on, which may be due to the fact that the data points in each window can be quite different from each other, leading to variability in the model performance. There was an extreme outlier between t = 100 and 200 which might be due to an anomaly in the data.

Similarly, using the learning_curve function of sklearn library, and dividing our input data into training and testing data, we are able to plot the learning curve of the model as well.

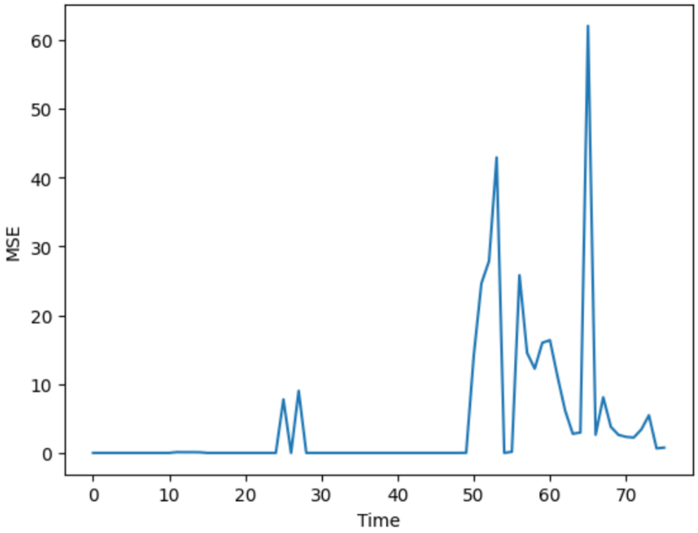


The training score shows how well the model fits the training data as the training set size increases, and the validation score shows how well the model generalizes to new data (i.e., the testing data) as the training set size increases.

From the graph, we can see that the model fits well with the training data, and initially performs badly for new data but also gradually improves in that domain.

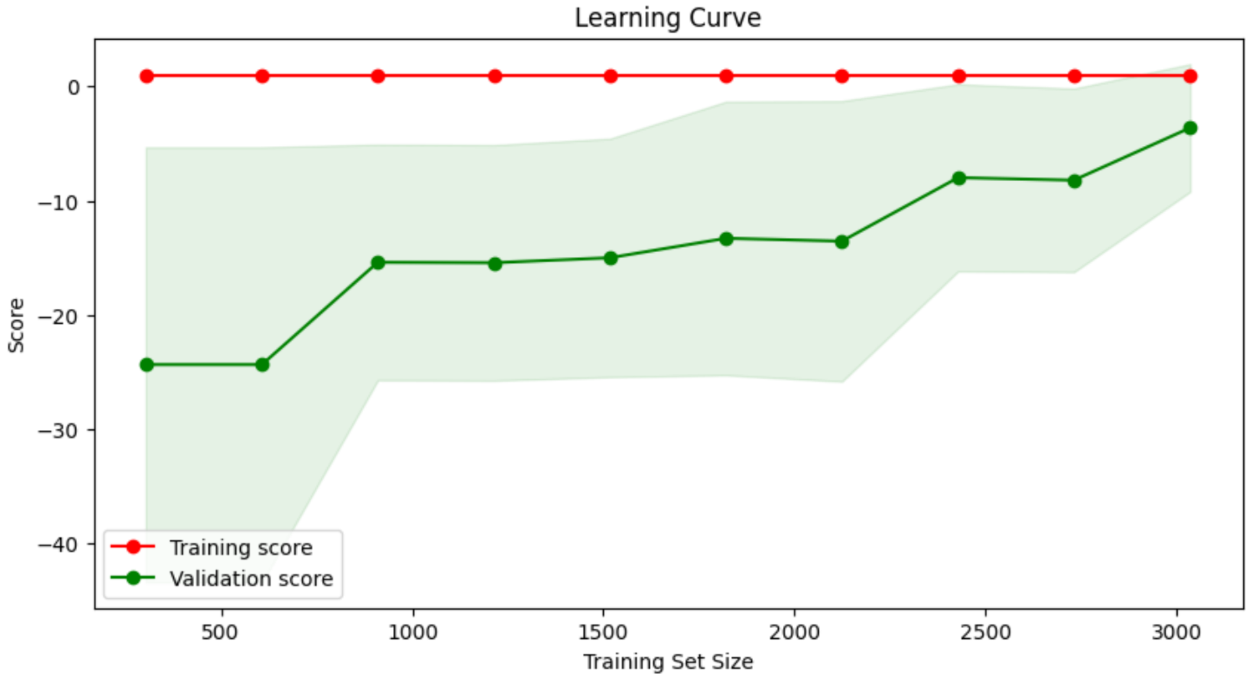
Model 2: Decision Tree Regressor

In the case of decision tree, a similar graph was drawn using the same feature set so that the model comparison would be fair.



Here too we can see that the model seems to fluctuate as the window rolls, however, the average MSE score is much lower than what we had with linear regression.

As for the learning curve,



Here too we can see that the validation score (which reminder, represents how well our model adapts to new data) is closer to the training score as compared to the linear regression learning curve.

Challenges Faced

While pre-processing the data, we found out that multiple regions which were present in the orders file were not present in the cluster map. We had to ensure that the rest of the regions are also mapped to numbers so the model can use them as part of the feature set. (Only 66 regions were provided in the cluster map, however, there were more than 700 regions in the order files).

Additionally, loading all the order files was a challenging process as well as the files had a significant amount of data, and it was very often for the RAM to be a bottleneck for the model training.

Other than that, we noticed that the models were often overfitting, which was not apparent until the data was tested on the test data. (as the learning curves also show)

Finally, since the data was relatively large, ensuring that it all was being used (and not running out of RAM) required the usage of optimized pandas methods such as `pd.DataFrame.from_records`

Conclusion

We conclude that different models can act differently on the given set of data, and some models can predict better than others in some specific cases. Depending on the problem type (classification vs regression). In this instance, the **decision tree regressor** was found to be a better fit for the data, as it has a lower average MSE (Mean Square Error), as well as a better learning curve.