Shrunkhala Nehete Github: snnehete

MET CS 777 – Term Project

Data: TLC Trip Record Data

The dataset used in this project was retrieved from the NYC Taxi and Limousine Commission [1]. It is the TLC Trip Record Data for the year 2019, between the months of January to June. It details all the trips completed by yellow taxi cabs in the NYC area. The attributes of the data are both continuous and categorical, and are detailed in data_dictionary_trip_records_yellow.pdf. Additionally, it should be noted that all the categorical attributes are mapped to integer values. We only convert the pickup and drop-off locations back to string values using the information in taxi_zone_lookup.csv, to increase readability of the results.

The data for each month is stored in a separate CSV file. Since the data was too large to compute within a reasonable time, 1,000,000 rows were randomly sampled from each month in order to create a smaller, more manageable dataset (dataset.csv).

Aim: Prediction

In this project, we aim to predict the duration that a particular trip would take given the pickup/dropoff locations and the time of the day/month. This would greatly help in terms of traffic planning and estimated time of arrival prediction.

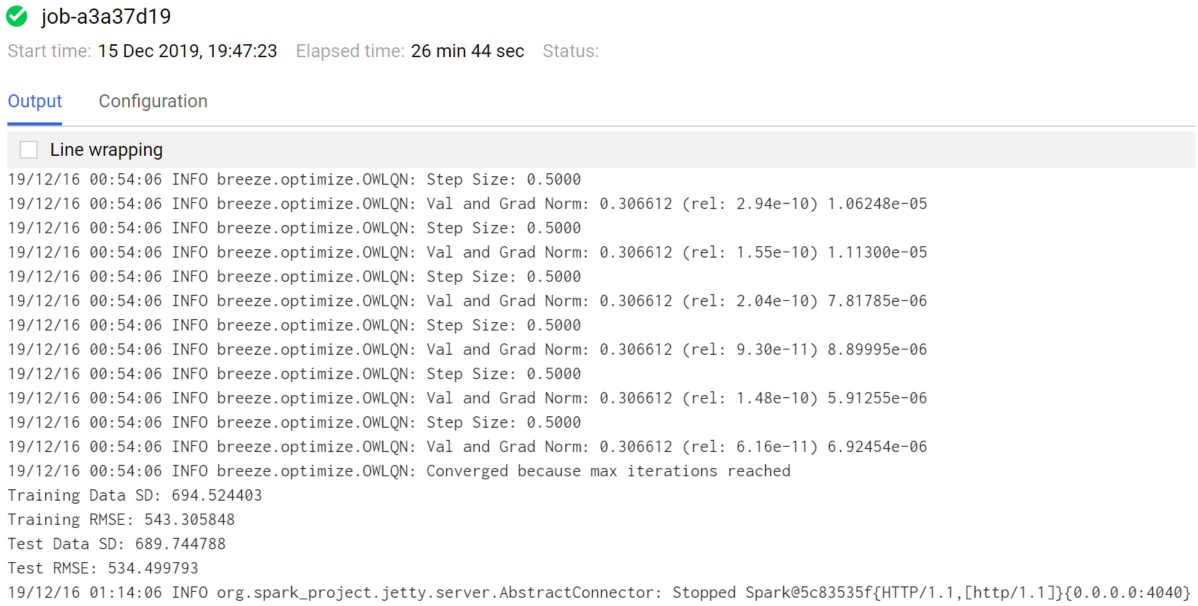
Implementation: Linear Regression

Some data cleaning was required before applying a learning model. We dropped all features concerning fare, as this is derived from the trip duration and distance. We also drop the vendor ID and passenger counts as these have no logical bearing on the trip duration. Finally, we check if all values are valid and convert the timestamp features to their individual components. The categorical features are one-hot encoded so that the information can be captured by the model.

The LInearRegression model from the pyspark ml library is used with a pyspark dataframe consisting of label (the trip duration) and a vector of all features. We use a combination of L1 and L2 normalization and initialize the model with regularization parameter of 0.07 and elastic net parameter of 0.8.

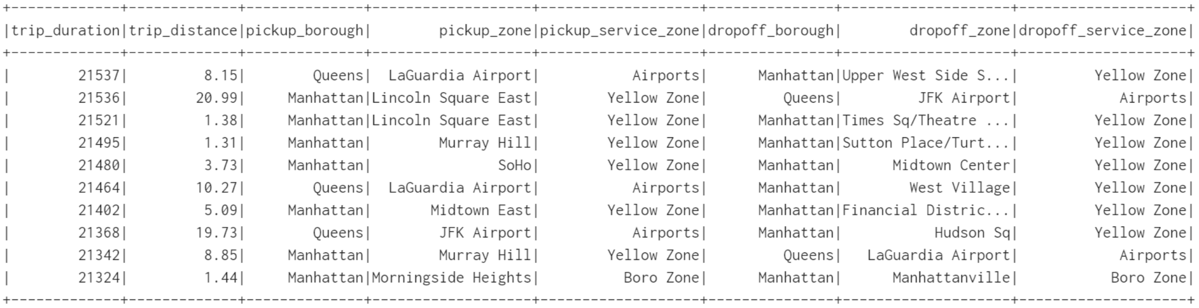
Results: Pyspark

Spark History
The program was run on a Google Cloud cluster of 4 worker nodes of 15 GB memory each.


job-a3a37d19
Start time: 15 Dec 2019, 19:47:23
Elapsed time: 26 min 44 sec
Status:
Output
Configuration
□ Line wrapping

```
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 2.94e-10) 1.06248e-05
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 1.55e-10) 1.11300e-05
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 2.04e-10) 7.81785e-06
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 9.30e-11) 8.89995e-06
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 1.48e-10) 5.91255e-06
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Step Size: 0.5000
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Val and Grad Norm: 0.306612 (rel: 6.16e-11) 6.92454e-06
19/12/16 00:54:06 INFO breeze.optimize.OWLQN: Converged because max iterations reached
Training Data SD: 694.524403
Training RMSE: 543.305848
Test Data SD: 689.744788
Test RMSE: 534.499793
19/12/16 01:14:06 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@5c83535f{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
```

Since most of the features in the dataset were categorical, we tried to manually find a correlation between the trip duration and the pickup/drop-off locations, but there appeared to be none.



```
+-------------+-------------+--------------+-------------------+-------------------+---------------+--------------------+--------------------+
|trip_duration|trip_distance|pickup_borough|        pickup_zone|pickup_service_zone|dropoff_borough|        dropoff_zone|dropoff_service_zone|
+-------------+-------------+--------------+-------------------+-------------------+---------------+--------------------+--------------------+
|        21537|         8.15|        Queens|  LaGuardia Airport|           Airports|      Manhattan|Upper West Side S...|         Yellow Zone|
|        21536|        20.99|     Manhattan|Lincoln Square East|        Yellow Zone|         Queens|         JFK Airport|            Airports|
|        21521|         1.38|     Manhattan|Lincoln Square East|        Yellow Zone|      Manhattan|Times Sq/Theatre ...|         Yellow Zone|
|        21495|         1.31|     Manhattan|        Murray Hill|        Yellow Zone|      Manhattan|Sutton Place/Turt...|         Yellow Zone|
|        21480|         3.73|     Manhattan|               SoHo|        Yellow Zone|      Manhattan|      Midtown Center|         Yellow Zone|
|        21464|        10.27|        Queens|  LaGuardia Airport|           Airports|      Manhattan|        West Village|         Yellow Zone|
|        21402|         5.09|     Manhattan|       Midtown East|        Yellow Zone|      Manhattan|Financial Distric...|         Yellow Zone|
|        21368|        19.73|        Queens|        JFK Airport|           Airports|      Manhattan|           Hudson Sq|         Yellow Zone|
|        21342|         8.85|     Manhattan|        Murray Hill|        Yellow Zone|         Queens|   LaGuardia Airport|            Airports|
|        21324|         1.44|     Manhattan|Morningside Heights|          Boro Zone|      Manhattan|      Manhattanville|           Boro Zone|
+-------------+-------------+--------------+-------------------+-------------------+---------------+--------------------+--------------------+
```

The results of training and testing the data on an 80/20 train-test split were as follows:

```
Training Data SD: 694.524403
Training RMSE: 543.305848

Test Data SD: 689.744788
Test RMSE: 534.499793
```

The standard deviation of the data describes the error if we predict all the values as the mean of all trip durations. We see that the model performs better than this baseline assumption. It also performs better on the test data than the training data, proving that there is no overfitting.

Conclusion

While linear regression performs reasonably well over randomly predicting values, it may be worth testing other types of regressors. This is because the trip duration is not linearly correlated to any of the features and thus, cannot be effectively predicted by a linear model.

References

[1] TLC Trip Record Data. (n.d.). Retrieved from https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.