Evan Joyce

Intro

For my project, I will be doing a prediction on the cost of a yellow taxi trip in New York City. One of the main modes of transportation in New York City is taxis. Even in the current age of Uber and Lyft, many people still choose to ride taxis and thus in one of the biggest cities in the world, we can obtain millions of points of data to analyze. For this project I plan to do a multiple linear regression model using spark in order to predict the price of the taxi trip. A linear regression model calculates the relationship between predicting features and a final y variable. For my project, the y variable I am predicting would be the 'fare amount' of the ride.

Why could this be useful and interesting?

As someone who is an avid user of many forms of transportation including ridesharing, metro electric scooters and more, I am interested to see if Taxis could be a more efficient and cost-effective way of getting around. It may provide some insight as to why Taxis continue to be a large part of New York City culture given how many other new transportation methods there are in the city.

It will be useful to see what features have the highest correlation to price for taxi rides. I am not sure if it exists, but it could provide an opportunity to create an app that will estimate prices for taxi rides. Currently Uber and Lyft probably create a model that uses all the features you input such as distance and time and gives you a price. Taxi's generally do not do this up front, so if there was a way to apply the model to an app to understand beforehand what the price will be, that would help customers and be useful.

Doing this model could also be useful to taxi drivers and owners to understand how and where they make the most money and which trips/times of day they should be looking to drive to maximize profit.

Dataset

The dataset I used for this project is from the New York City Taxi & Limousine Commission website. On this website, they have a multitude of datasets dating back to 2009. Each year has many datasets. For this project I will be using a dataset from January of 2019. I decided not to use a dataset from 2020 or 2021 because I wanted to test a model on data from before Covid-19. The dataset I am using had over 7.5 million entries with 18 columns. But, due to memory constraints and run time issues, I will be using a subset of this data and using 1.25 million entries.

An interesting element from this site is that they have separate datasets for yellow taxis and green taxis. Before this. I did not know there was a difference between the two, but I learned that yellow taxis are authorized to attract street hails anywhere and green cabs can only do that in very designated areas around the city. There are significantly fewer green taxis in New York, so I decided to only do the yellow taxis dataset.

Data Exploration and Manipulation

Once I had my data loaded into python, I decided to explore the data using pandas to get a better understanding of it. One of the first things I recognized was that some of the columns were dates, which can cause problems when trying to do a linear regression. As a result I decided to omit these columns when conducting my regression model. I also came across a unique column name called, “store_and_fwd_flag”. I was immediately curious as to what this meant, and after some research on the website I obtained the data from, I found that it meant the following:

“This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server (Y=store and forward; N=not a store and forward trip)”

To me, this did not really matter in my prediction so I decided to omit this as well.

First Steps- Linear Regression

I decided to test my linear regression model on my data using Sklearn first before I went on to doing it with Spark. As with all regression models, I went through the process of choosing my features, my predicting variable, splitting the data and fitting it. At first I just did this linear regression on a couple hundred rows of data just to see if it was working. To my astonishment I was getting perfect results, every single cost prediction was accurate. I initially was excited until I realized it is very unlikely for things to be this perfect, so I decided to take a step back and do some more data exploration.

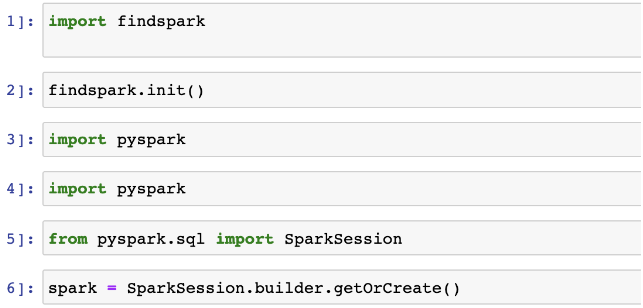
After lots of analysis I found a large problem with my model and the data. The first being that the data consisted of a tip_amount, fare_amount and total_amount. My initial prediction variable was going to be about the “total_amount”, but then I realized if I was feeding the tip amount and fare amount into the features, then the model would just add those two together and get the prediction right every time. In order to remedy this, I decided to omit the total amount from the equation and then have my prediction be “fare_amount”. Once I did this, I reran the model in sklearn and was able to obtain acceptable results with an r squared of around .83.

My next step was to increase the amount of data to a more substantial size of around 100,000 rows of data. This is where I started to run into more trouble. I ran the same model again just with more data and still in sklearn and got an r squared of .18. I thought something must be wrong so I ran it a couple more times resetting the kernel and the notebook and still got poor results.

I decided to go back a step again and do some more data exploration to see if I could find any logical explanation for why this model is not working. I wanted to make sure that the model would work before I even attempted to do it in spark. After getting summary statistics of all of the columns in the dataset, I realized that there were some major outliers in the fare amount and total amount columns. First, I was coming across negative values which doesn't make sense because how can someone be paying negative amounts for a taxi ride. Second, I was seeing fare_amounts as high as 3,000 dollars which also seemed egregious and that something was going wrong. I actually consulted some family who lives in New York and asked them what the average range of taxi costs would be. I took this advice and decided to set a minimum amount for the fare at 0$ to get rid of the negative trip values, and then set a max amount at 300. One approach that I could do next time is to scale the data in between 0 and 1 so that outliers do not impact the data as much. But, I figured that the data had to be incorrect so I decided to omit those outliers. Once I made these changes I reran the data in sklearn and got an r squared of .88. My next step was to apply this data and the linear regression model in Spark.

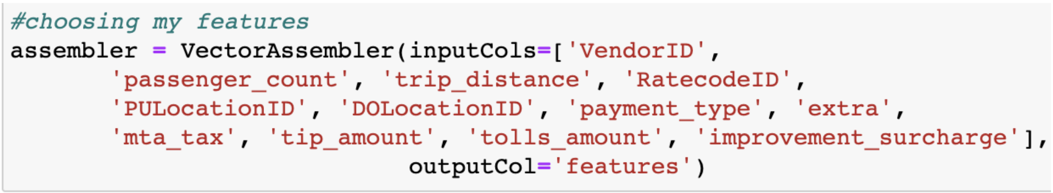
Model In Spark

To effectively utilize spark in python, I was able to install and use ‘pyspark’ in jupyter notebook. After many hours of trial and error I was finally able to get spark running. Below is the process of loading and starting a spark session for my model.



```
1]: import findspark

2]: findspark.init()

3]: import pyspark

4]: import pyspark

5]: from pyspark.sql import SparkSession

6]: spark = SparkSession.builder.getOrCreate()
```

It is important to note that I took my dataset from pandas and turned it into a spark dataframe. The next step was to initialize my features and my prediction variables. I did not end up using all of the columns in the dataset as features. Below one can see which variables I choose:



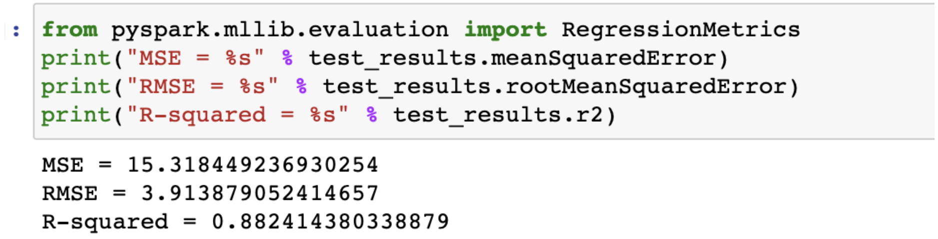
```
#choosing my features
assembler = VectorAssembler(inputCols=['VendorID',
       'passenger_count', 'trip_distance', 'RatecodeID',
       'PULocationID', 'DOLocationID', 'payment_type', 'extra',
       'mta_tax', 'tip_amount', 'tolls_amount', 'improvement_surcharge'],
                            outputCol='features')
```

Next, splitting the data into testing and training was necessary and I eventually settled on using a 70/30 split, where 70% of the data was for training and 30% was for testing.

Results

The main part of my project was using the month of January data to both train and test the model. To evaluate my model, I decided to use a spark mllib library to get the mean squared error, the root mean squared error and the r2. To me, I mainly looked at the r2 value because this represents the goodness of fit and the "percentage of the variance in the dependent variable that the independent variable explains".

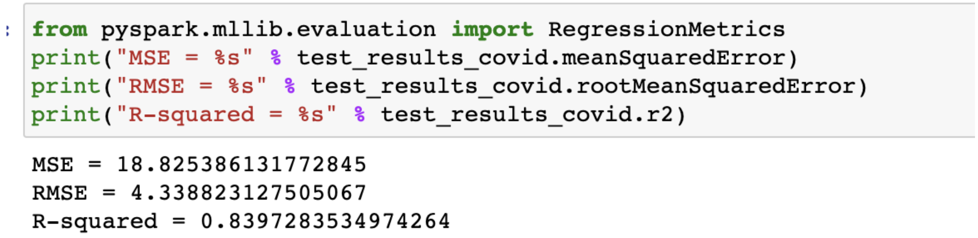
The first results below are from when I trained and tested on the month of January:



```
from pyspark.mllib.evaluation import RegressionMetrics
print("MSE = %s" % test_results.meanSquaredError)
print("RMSE = %s" % test_results.rootMeanSquaredError)
print("R-squared = %s" % test_results.r2)
```

```
MSE = 15.318449236930254
RMSE = 3.913879052414657
R-squared = 0.882414380338879
```

It is important to keep in mind that this data was from before covid, so I thought it would be interesting to see how my model works on data that is taken from a month where covid was prominent. As a result, I decided to keep the training on the same original month and then just test the model on the covid data. I took a million rows from February of 2020 to test. Below we can see the results:

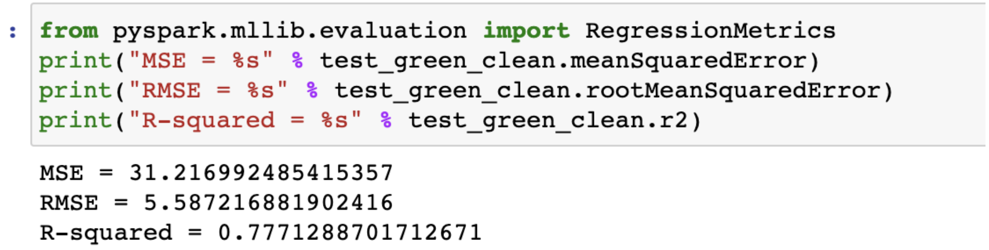


```
from pyspark.mllib.evaluation import RegressionMetrics
print("MSE = %s" % test_results_covid.meanSquaredError)
print("RMSE = %s" % test_results_covid.rootMeanSquaredError)
print("R-squared = %s" % test_results_covid.r2)
```

```
MSE = 18.825386131772845
RMSE = 4.338823127505067
R-squared = 0.8397283534974264
```

This is interesting to see because the results were slightly worse. This implies that the data from pre covid and during covid did change behavior slightly.

This got me thinking more about what other datasets I could apply my model to. As mentioned earlier, the site where I obtained the yellow taxi data also had data for green taxis. So I decided to test my model on the green taxi data next, and obtained the following results:

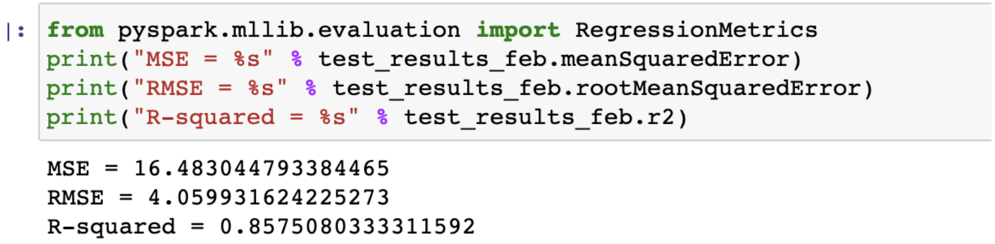


```
from pyspark.mllib.evaluation import RegressionMetrics
print("MSE = %s" % test_green_clean.meanSquaredError)
print("RMSE = %s" % test_green_clean.rootMeanSquaredError)
print("R-squared = %s" % test_green_clean.r2)
```

```
MSE = 31.216992485415357
RMSE = 5.587216881902416
R-squared = 0.7771288701712671
```

Given that the green taxis have a very different service and client base, it makes sense that my model does not predict this test data as well as the original test data.

The last dataset I wanted to test was testing my model on the month following my original dataset. My original model with an r2 of .88 was trained and tested on data from January of 2019, so I wanted to test on February of 2019 to see if the results were similar. Before I ran this test data, I expected this to produce very similar results to the month of January. We can see the results below:



```
from pyspark.mllib.evaluation import RegressionMetrics
print("MSE = %s" % test_results_feb.meanSquaredError)
print("RMSE = %s" % test_results_feb.rootMeanSquaredError)
print("R-squared = %s" % test_results_feb.r2)
```

```
MSE = 16.483044793384465
RMSE = 4.059931624225273
R-squared = 0.8575080333311592
```

The results are very similar which means the trends of the data are not much different.

Conclusion

Throughout this project I was exposed to spark and its ability to analyze large data sets. I successfully built a linear regression model and tested on a variety of datasets. If I were to continue working on this project in the future, I would like to continue testing on different datasets as well as testing on a larger amount of data. In general, I learned to be very patient with spark especially when getting it up and running. I struggled for days when first installing it and getting it running, but was able to persevere and make a project I can be proud about.