Course Name	**Master of Science in Business Analytics**
Module Name	**APPLIED BIG DATA AND VISUALIZATION (CS6502)**
Module Leader	**Dr. Patrick Healy**
Batch	**2019-2020**
Semester	**Spring**

ML Project Assignment

Submitted by:
Aishwarya C Inamdar
19084404

Table of Contents

Query to select 1000 rows 3

Key steps for PCA 3

Data Analysis 3

SQL query for data analysis 3

Checking the outlier data 4

SQL query for checking the outliers data 4

SQL query for setting a benchmark value 4

Creating a Linear Regression model over the selected features 6

Model description 6

SQL query creating the linear regression model creation 6

Screenshot of the model evaluation report 7

SQL query for Model Evaluation 7

SQL query for Model Prediction 8

Query to select 1000 rows

SELECT * FROM `bigquery-public-data.chicago_taxi_trips.taxi_trips`
where pickup_latitude IS NOT NULL and dropoff_latitude IS NOT NULL
LIMIT 1000
(Refer taxi_records.csv for the extracted output)

Key steps for PCA

Following are the steps carried out for PCA:
1) Exploratory Data Analysis: In this step, I have understood and summarized the dataset. It is a crucial step before building the machine learning model so as to create models that correctly interpret the results.
2) Feature Extraction: In this step, we have used a feature selection technique, heat map to find correlation between features which helped us in deciding on the features that created an impact for creating the model.
3) Standardize the data: Before applying PCA, the features in the dataset needs to be scaled.
4) PCA projection to 2D: The dataset of 23 columns is projected in 5 components.
5) Plotting the principle components: To visualize the variance, I have plotted the principle components.
(Refer file PCA.ipynb for the python code)

Data Analysis

Before creating the model, analyse the data and observe that tips and extras are included in the trip_total.

SQL query for data analysis

```
SELECT
 fare, tips, tolls, extras, trip_total
 FROM
  `bigquery-public-data.chicago_taxi_trips.taxi_trips`
 WHERE
 extract(year from trip_start_timestamp) IN (2013,2014,2015,2016,2017);
```

Row	fare	tips	tolls	extras	trip_total
1	12.5	0.0	*null*	0.0	12.5
2	13.75	0.0	*null*	1.0	14.75
3	11.0	0.0	*null*	0.0	11.0
4	25.25	5.25	*null*	1.0	31.5
5	45.5	9.9	*null*	4.0	59.4
6	14.0	3.5	*null*	0.0	17.5

Checking the outlier data

I don't want to include tips in our model, because it's too random, so we use tolls + fare for total price.

SQL query for checking the outliers data

```
SELECT
 CAST(IF(Min(pickup_latitude) > Min(dropoff_latitude),Min(dropoff_latitude),Min(pickup_latitude)) as INT64)  as min_latitude,
 CAST(IF(MAX(pickup_latitude) < MAX(dropoff_latitude),MAX(dropoff_latitude),MAX(pickup_latitude)) as INT64) as max_latitude,
 CAST(IF(Min(pickup_longitude) > Min(dropoff_longitude),Min(dropoff_longitude),Min(pickup_longitude)) as INT64)  as min_longitude,
 CAST(IF(MAX(pickup_longitude) < MAX(dropoff_longitude),MAX(dropoff_longitude),MAX(pickup_longitude)) as INT64) as max_longitude,
 MIN(tolls + fare) as min_price,
 Max (tolls + fare) as max_price
 FROM
  `bigquery-public-data.chicago_taxi_trips.taxi_trips`
```

Row	min_latitude	max_latitude	min_longitude	max_longitude	min_price	max_price
1	42	42	-88	-88	0.0	9999.99

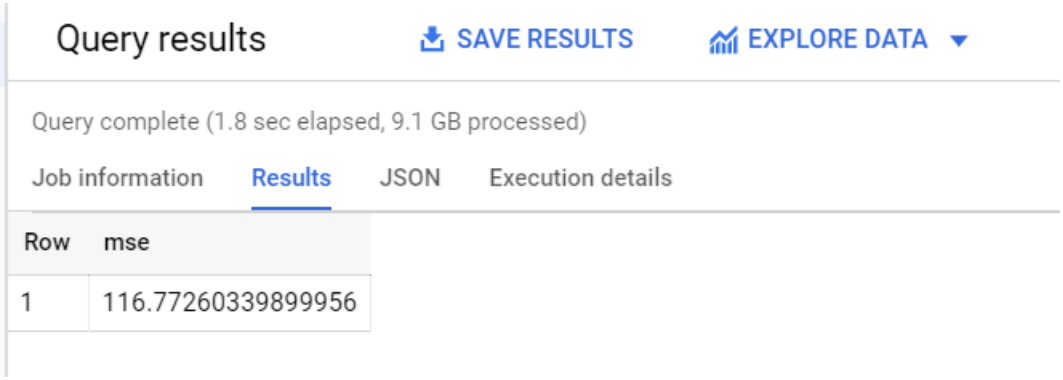
I am taking prices greater than $0 and lower than $2000.
For the benchmark, we choose to calculate the average cost per distance and predict for each trip the cost with this formula:
taxiPrice = euclidianDist *avgCostPerDist

SQL query for setting a benchmark value

```
With avg as (
 SELECT
   AVG(tolls+ fare) /  AVG(dist) as price_per_dist
 FROM
  (SELECT
  tolls,
  fare,
  SQRT(POW((pickup_longitude - dropoff_longitude),2) + POW(( pickup_latitude - dropoff_latitude), 2)) as dist
  FROM
   `bigquery-public-data.chicago_taxi_trips.taxi_trips`
  WHERE
```

```
  extract(year from trip_start_timestamp) IN (2013,2014,2015,2016,2017)
  AND (tolls+ fare) BETWEEN 0 and 2000
    )
 WHERE dist > 0
)

SELECT
 AVG( POW(( predict_price - price),2)) as mse
FROM
 (SELECT
  (tolls+ fare) as price,
  SQRT(POW((pickup_longitude - dropoff_longitude),2) + POW(( pickup_latitude - dropoff_latitude),
2)) * price_per_dist as predict_price
 FROM
   `bigquery-public-data.chicago_taxi_trips.taxi_trips`
   CROSS JOIN avg
 WHERE
 (tolls+ fare) BETWEEN 0 and 2000)
```



I got a 116.77260339899956 in MSE (Mean Square Error). This means that when we predict with our benchmark model, we got an average of $ 10.8061373024 difference with truth.
We got our initial benchmark.
Benchmark = $ 10.8061373024

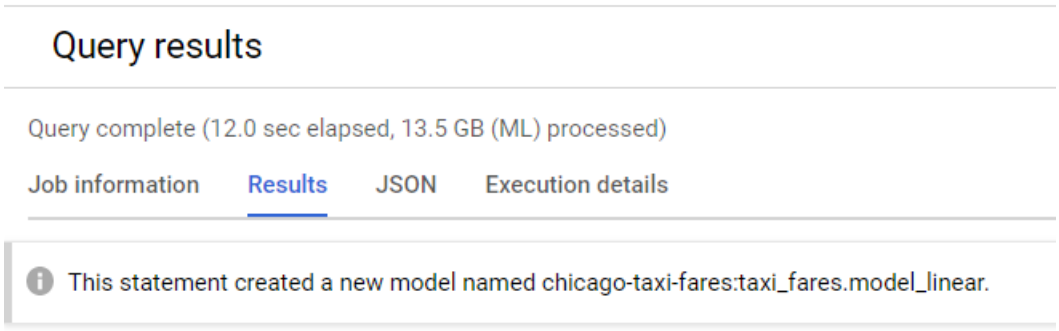
Creating a Linear Regression model over the selected features

Model description

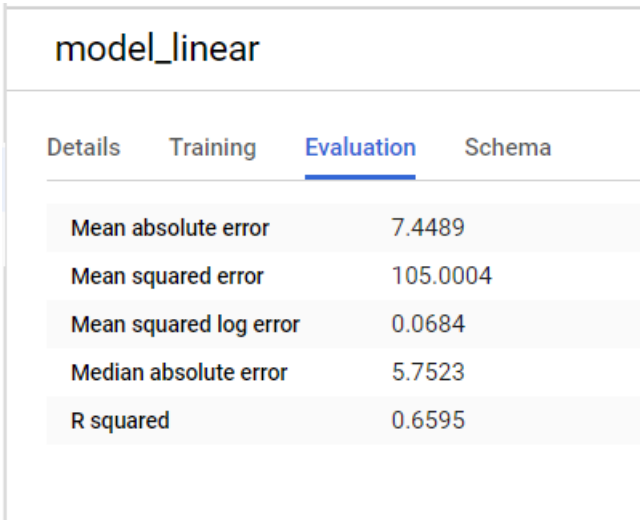
- I have created a simple linear regression model for predicting the taxi fare using the Chicago Taxi Trips dataset. I have calculated the average cost per distance using the 'taxiPrice = euclidianDist *avgCostPerDist' logic, considering the fare from 0 to 100 and the total fare (fare + tolls) from 0 to 2000.
- The data is trained for years from 2013 to 2017 and is evaluated and predicted for the years 2018,2019 and 2020.

SQL query creating the linear regression model creation

```
CREATE MODEL
 `taxi_fares.model_linear`
OPTIONS
 (model_type='linear_reg',
  labels = ['total_amount'])
WITH taxitrips AS
SELECT
 SQRT(POW((pickup_longitude - dropoff_longitude),2) + POW(( pickup_latitude - dropoff_latitude),
2)) as dist,
 SQRT(POW((pickup_longitude - dropoff_longitude),2)) as longitude,
 SQRT(POW(( pickup_latitude - dropoff_latitude), 2)) as latitude,
 (tolls + fare) as total_amount,
 EXTRACT(YEAR FROM trip_start_timestamp) as year
 FROM
   `bigquery-public-data.chicago_taxi_trips.taxi_trips`
 WHERE
     extras >0 AND fare >0 AND  tips>0 AND tolls>0 AND trip_miles>0 AND
     extract(year from trip_start_timestamp) IN (2013,2014,2015,2016,2017)
     AND pickup_longitude > -88
     AND pickup_longitude < -86
     AND dropoff_longitude > -88
     AND dropoff_longitude < -86
     AND pickup_latitude > 41
     AND pickup_latitude < 42
     AND dropoff_latitude > 41
     AND dropoff_latitude < 42
     AND fare BETWEEN 0 and 100
     AND (tolls + fare) BETWEEN 0 AND 2000;
```



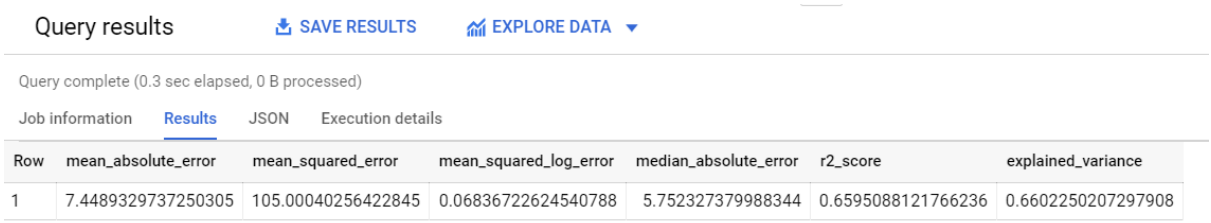
Screenshot of the model evaluation report



SQL query for Model Evaluation

```
WITH eval_table as
 (SELECT
 SQRT(POW((pickup_longitude - dropoff_longitude),2) + POW(( pickup_latitude - dropoff_latitude),
2)) as dist,
 SQRT(POW((pickup_longitude - dropoff_longitude),2)) as longitude,
 SQRT(POW(( pickup_latitude - dropoff_latitude), 2)) as latitude,
 (tolls + fare) as total_amount
 FROM
   `bigquery-public-data.chicago_taxi_trips.taxi_trips`
 WHERE
extras >0 AND fare >0 AND  tips>0 AND tolls>0 AND trip_miles>0 AND
     extract(year from trip_start_timestamp) IN (2013,2014,2015,2016,2017)
     AND pickup_longitude > -88
     AND pickup_longitude < -86
     AND dropoff_longitude > -88
     AND dropoff_longitude < -86
     AND pickup_latitude > 41
     AND pickup_latitude < 42
     AND dropoff_latitude > 41
```

```
    AND dropoff_latitude < 42
    AND fare BETWEEN 0 and 100
    AND (tolls + fare) BETWEEN 0 AND 2000)
select * from ml.evaluate(model `taxi_fares.model_linear`);
```



Query results SAVE RESULTS EXPLORE DATA

Query complete (0.3 sec elapsed, 0 B processed)

Job information | Results | JSON | Execution details

Row	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	r2_score	explained_variance
1	7.4489329737250305	105.00040256422845	0.06836722624540788	5.752327379988344	0.6595088121766236	0.6602250207297908

We have a **MSE** of 105.00040256422845 and **RMSE** of **$ 10.2469704091**

SQL query for Model Prediction

```
SELECT * FROM   ml.PREDICT(MODEL `taxi_fares.model_linear`, (
 WITH taxi_fares AS
(
SELECT
 SQRT(POW((pickup_longitude - dropoff_longitude),2) + POW(( pickup_latitude - dropoff_latitude),
2)) as dist,
 SQRT(POW((pickup_longitude - dropoff_longitude),2)) as longitude,
 SQRT(POW(( pickup_latitude - dropoff_latitude), 2)) as latitude,
 (tolls + fare) as total_amount,
 EXTRACT(YEAR FROM trip_start_timestamp) as year
 FROM
   `bigquery-public-data.chicago_taxi_trips.taxi_trips`
 WHERE
    extras >0 AND fare >0 AND  tips>0 AND tolls>0 AND trip_miles>0 AND
    extract(year from trip_start_timestamp) IN (2018,2019,2020)
    AND pickup_longitude > -88
    AND pickup_longitude < -86
    AND dropoff_longitude > -88
    AND dropoff_longitude < -86
    AND pickup_latitude > 41
    AND pickup_latitude < 42
    AND dropoff_latitude > 41
    AND dropoff_latitude < 42
    AND fare BETWEEN 0 and 100
    AND (tolls + fare) BETWEEN 0 AND 2000
)
select * from taxi_fares
limit 1000
));
```

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.3 sec elapsed, 13.5 GB processed)

Job information | Results | JSON | Execution details

Row	predicted_total_amount	dist	longitude	latitude	total_amount	year
1	77.66878177978845	0.24579047579358068	0.15210537199999408	0.19307230199999736	77.0	2018
2	81.29093324638849	0.24579047579358068	0.15210537199999408	0.19307230199999736	77.0	2019
3	81.29093324638849	0.24579047579358068	0.15210537199999408	0.19307230199999736	79.0	2019
4	36.88965896013477	0.0	0.0	0.0	20.0	2018
5	64.361822788328	0.2975290591478324	0.2841713059999904	0.08814879399999853	67.15	2018
6	65.23991500639274	0.2973249016059996	0.28204674799999907	0.09408362799999992	65.4	2018
7	68.86206647299278	0.2973249016059996	0.28204674799999907	0.09408362799999992	66.65	2019
8	70.07175463289272	0.2988895051675825	0.28106800899999484	0.1016646969999968	67.65	2019
9	74.70956215179116	0.30763928847467586	0.2789043629999952	0.12982406599999763	67.65	2019

Rows per page: 100 1 - 100 of 1000

Refer predicted_results.csv for the output of the first 1000 records.