Christos Kallaras, p2822009
Department of Management Science and Technology
Master in Business Analytics
Athens University of Economics and Business

TASK 3

In this task I will train a linear regression model that could predict the departure delay a flight may have by using, as input, its origin, its airways, and its departure time.

Note 1: The code was written and executed through Jupiter's Notebook. All figures are from there. Configuring the max memory in 14g was necessary for the project to run successfully with Jupiter's Notebook.

Note 2: In order for the code to run successfully with any given path a variable path was created in which you must declare the path of the file in your local PC.

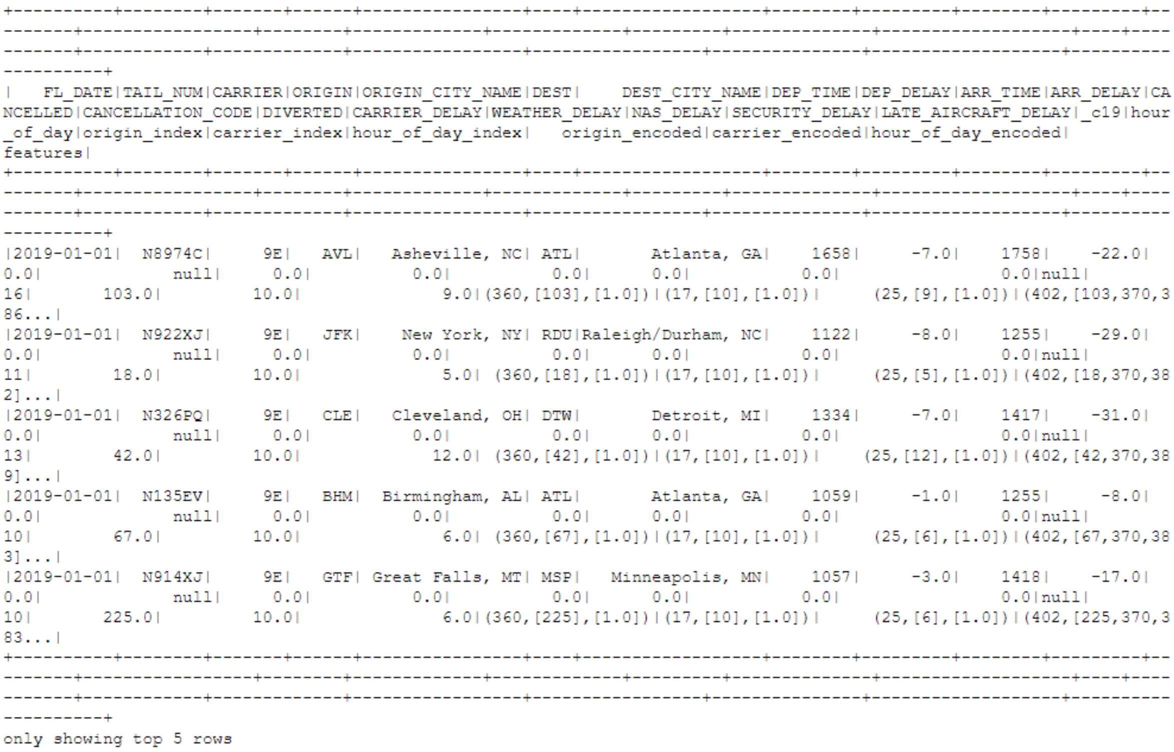
Note 3: In task3.png are the top 10 predictions.

DATA CLEANING

As with the previous task, the dataset needed to be cleaned first. Specifically, I should not consider in my analysis any airport/airway belonging in the lowest 1% percentile, regarding the number of flights. I used SparkSQL again for this transformation. First I created 2 tables one containing all the airports and the number of flights from that airport (lets call this table **airports**), and the other containing the airways and the number of flights they participated in (lets call this table airways). The results of those 2 queries were also transformed into a table and from there I extracted the 1% percentile for airports and airways (**airport_limit** and **airways_limit** respectively). Then I created a query that joined the 2 tables, airports and airways, with the originally dataset with the condition that the number of flights is bigger or equal to the 1% percentile I extracted earlier. Results of this process can be seen in task2.pdf since it is the same.

DATA TRANSFORMATION

From the departure time I only need the hour of the day so I create a new column in the dataset called "hour_of_day" and fill it with the hour. I extract the hour of the day by dividing departure time with 100. The input variables to the regression model are categorical variables. The model takes only numerical so I need to transform them. I will use One Hot Encoding for that. With One Hot Encoding I will convert categorical data to a vector of ones and zeros (binary form). The length of vector is determined by number of expected classes or categories. Each element in the vector represents a class. A one indicates which class it is and everything else will be zero. I apply this transformation to all the variables of input (origin, airways and hour of day). Finally created a vector containing the features to train the model. For that I used Vector Assembler, a transformer that combines a given list of columns(in our case the origin, carrier and hour of the day after their transformation using one hot encoding) into a single vector column. Figure 1 shows the dataset after all the transformations.

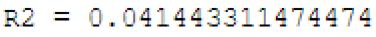


```
+----------+--------+-------+------+----------------+----+------------------+--------+---------+--------+---------+--
-------+-----------------+--------+-------------+-------------+---------+--------------+-------------------+----+----
-------+------------+-------------+-----------------+-----------------+---------------+-------------------+---------
----------+
|   FL_DATE|TAIL_NUM|CARRIER|ORIGIN|ORIGIN_CITY_NAME|DEST|    DEST_CITY_NAME|DEP_TIME|DEP_DELAY|ARR_TIME|ARR_DELAY|CA
NCELLED|CANCELLATION_CODE|DIVERTED|CARRIER_DELAY|WEATHER_DELAY|NAS_DELAY|SECURITY_DELAY|LATE_AIRCRAFT_DELAY|_c19|hour
_of_day|origin_index|carrier_index|hour_of_day_index|   origin_encoded|carrier_encoded|hour_of_day_encoded|
features|
+----------+--------+-------+------+----------------+----+------------------+--------+---------+--------+---------+--
-------+-----------------+--------+-------------+-------------+---------+--------------+-------------------+----+----
-------+------------+-------------+-----------------+-----------------+---------------+-------------------+---------
----------+
|2019-01-01|  N8974C|     9E|   AVL|   Asheville, NC| ATL|       Atlanta, GA|    1658|     -7.0|    1758|    -22.0|
0.0|             null|     0.0|          0.0|          0.0|      0.0|           0.0|                0.0|null|
16|       103.0|         10.0|              9.0|(360,[103],[1.0])|(17,[10],[1.0])|     (25,[9],[1.0])|(402,[103,370,3
86...|
|2019-01-01|  N922XJ|     9E|   JFK|    New York, NY| RDU|Raleigh/Durham, NC|    1122|     -8.0|    1255|    -29.0|
0.0|             null|     0.0|          0.0|          0.0|      0.0|           0.0|                0.0|null|
11|        18.0|         10.0|              5.0| (360,[18],[1.0])|(17,[10],[1.0])|     (25,[5],[1.0])|(402,[18,370,38
2]...|
|2019-01-01|  N326PQ|     9E|   CLE|   Cleveland, OH| DTW|       Detroit, MI|    1334|     -7.0|    1417|    -31.0|
0.0|             null|     0.0|          0.0|          0.0|      0.0|           0.0|                0.0|null|
13|        42.0|         10.0|             12.0| (360,[42],[1.0])|(17,[10],[1.0])|    (25,[12],[1.0])|(402,[42,370,38
9]...|
|2019-01-01|  N135EV|     9E|   BHM|  Birmingham, AL| ATL|       Atlanta, GA|    1059|     -1.0|    1255|     -8.0|
0.0|             null|     0.0|          0.0|          0.0|      0.0|           0.0|                0.0|null|
10|        67.0|         10.0|              6.0| (360,[67],[1.0])|(17,[10],[1.0])|     (25,[6],[1.0])|(402,[67,370,38
3]...|
|2019-01-01|  N914XJ|     9E|   GTF| Great Falls, MT| MSP|   Minneapolis, MN|    1057|     -3.0|    1418|    -17.0|
0.0|             null|     0.0|          0.0|          0.0|      0.0|           0.0|                0.0|null|
10|       225.0|         10.0|              6.0|(360,[225],[1.0])|(17,[10],[1.0])|     (25,[6],[1.0])|(402,[225,370,3
83...|
+----------+--------+-------+------+----------------+----+------------------+--------+---------+--------+---------+--
-------+-----------------+--------+-------------+-------------+---------+--------------+-------------------+----+----
-------+------------+-------------+-----------------+-----------------+---------------+-------------------+---------
----------+
only showing top 5 rows
```

Figure 1: Data after transformation

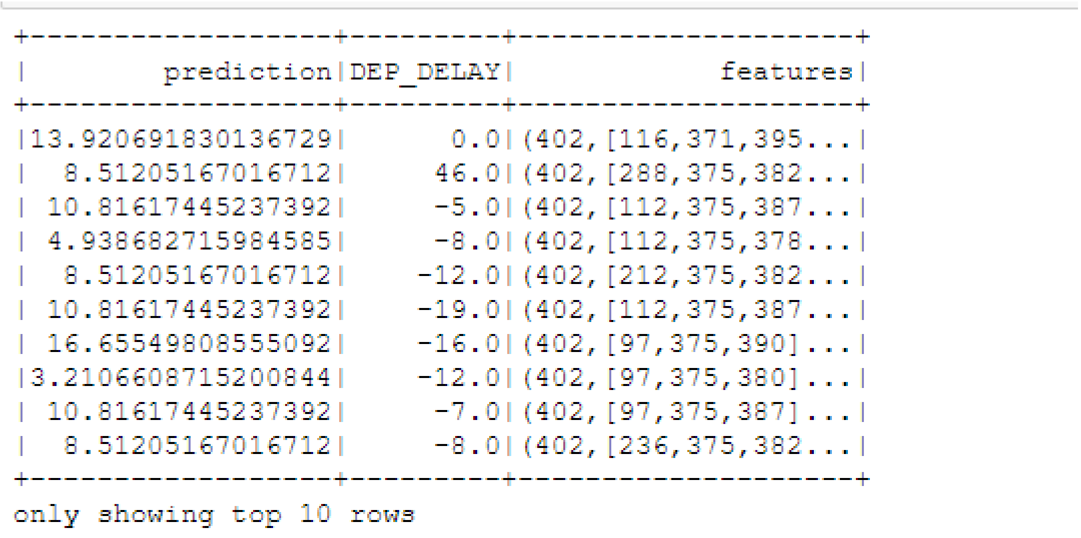
REGRESSION MODEL

The dataset was splitted; 70% for training the model and 30% for testing. I used the train dataset to fit and train the model. The adjusted R-squared is 0.041 (Figure 2). This is not good, it means that the model does not explains all the variability of the departure delay around its mean. Still is acceptable although an R2 value greater than 0.06 would be ideal. Figure 3 shows the top 10 predictions along with the actual value and the vector of features.



```
R2 = 0.041443311474474
```

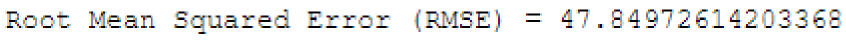
Figure 2: Adjusted R-squared



```
+------------------+---------+--------------------+
|        prediction|DEP_DELAY|            features|
+------------------+---------+--------------------+
|13.920691830136729|      0.0|(402,[116,371,395...|
|  8.51205167016712|     46.0|(402,[288,375,382...|
| 10.81617445237392|     -5.0|(402,[112,375,387...|
| 4.938682715984585|     -8.0|(402,[112,375,378...|
|  8.51205167016712|    -12.0|(402,[212,375,382...|
| 10.81617445237392|    -19.0|(402,[112,375,387...|
| 16.65549808555092|    -16.0|(402,[97,375,390]...|
|3.2106608715200844|    -12.0|(402,[97,375,380]...|
| 10.81617445237392|     -7.0|(402,[97,375,387]...|
|  8.51205167016712|     -8.0|(402,[236,375,382...|
+------------------+---------+--------------------+
only showing top 10 rows
```

Figure 3: Top 10 Predictions

As we can see from Figure 4 , the RMSE is high at 47,8.



```
Root Mean Squared Error (RMSE) = 47.84972614203368
```

Figure 4: Root Mean Squared Error (RMSE)

CONCLUSION

Adjusted R-squared is small while RMSE is almost 50. The predictions were also not accurate. So in conclusion we can say that it is not suggested to use a linear regression model that could predict the departure delay a flight may have by using as input its origin, its airways, and its departure time. Maybe it could be possible if we change the input or use a different Machine Learning model.