Predicting Airline Delays Using K-Nearest Neighbor

Michael Srouji
Max Sestero
Kishan Baliga

Motivation

- Very prevalent in everyday business
- Frustrating issue
- There is previous work to lead us in the right direction

Resources Used

- Sklearn
  - Sklearn is a python library that has an implementation of KNN
  - Sklearn has functions to measure the accuracy of our model
- Pandas
  - Used to create our data frame
- Google Colab
  - Allows us to code on google servers with more resources than our own computers
  - Easy to collaborate

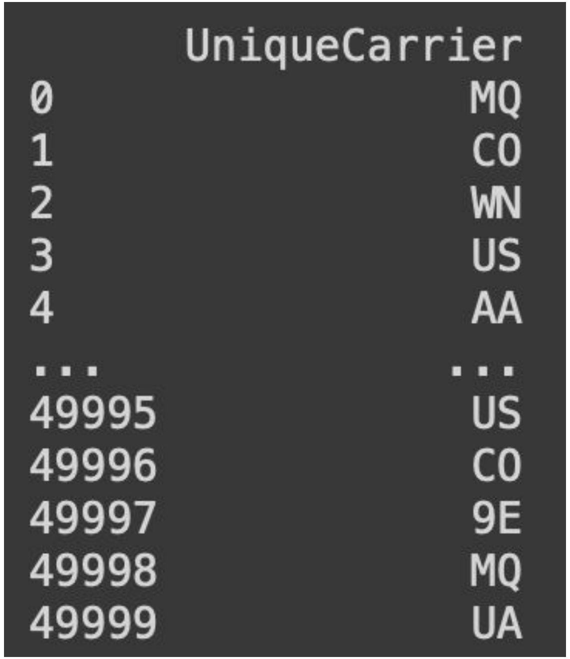
The Data

- Data Expo 2009: Airline on time data
- All domestic flights from October 1987 to April 2008
- Over 100 Million total entries

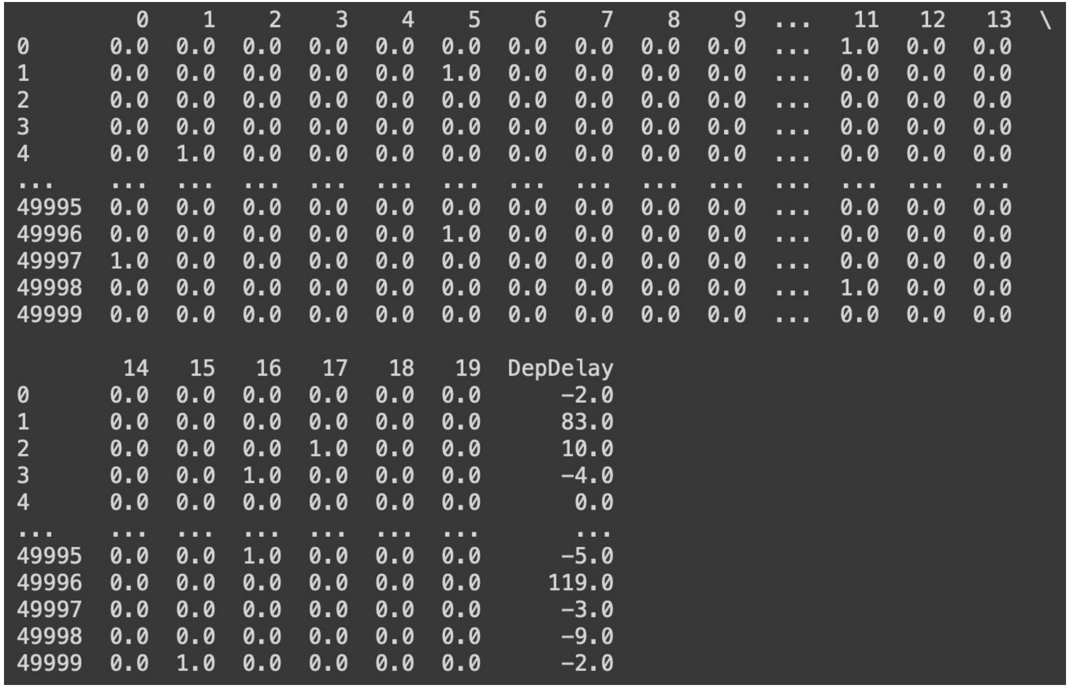
Features Considered

- Year
- Month
- Day
- Day of Month
- Day of Week
- Scheduled Departure Time
- Scheduled Arrival Time
- Unique Carrier Code
- Flight Number
- Tail Number
- Scheduled length
- Origin Airport code
- Distance
- Taxi in Time
- Carrier Delay
- Weather Delay
- NAS Delay
- Security Delay
- Late Aircraft Delay

One-hot encoding



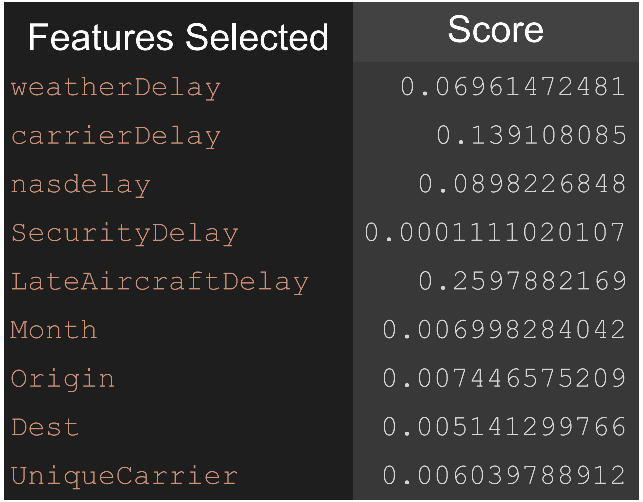
```
      UniqueCarrier
0                MQ
1                CO
2                WN
3                US
4                AA
...             ...
49995            US
49996            CO
49997            9E
49998            MQ
49999            UA
```



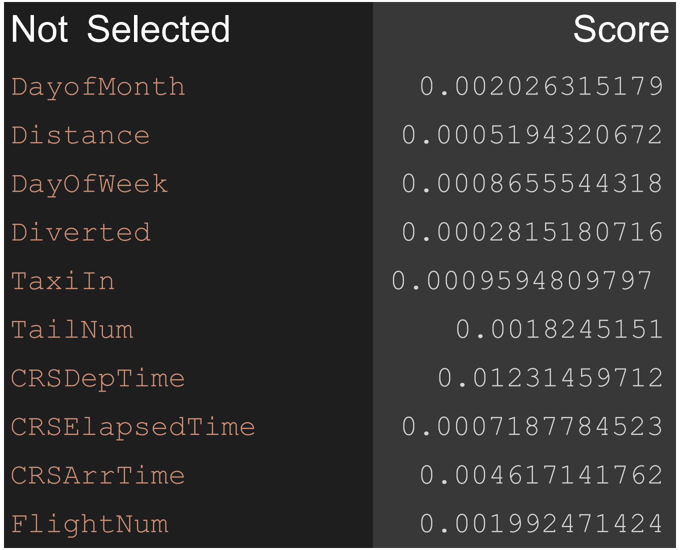
```
         0    1    2    3    4    5    6    7    8    9  ...   11   12   13  \
0      0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
1      0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
2      0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
3      0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
4      0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
...    ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
49995  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
49996  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
49997  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
49998  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  1.0  0.0  0.0
49999  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

        14   15   16   17   18   19  DepDelay
0      0.0  0.0  0.0  0.0  0.0  0.0      -2.0
1      0.0  0.0  0.0  0.0  0.0  0.0      83.0
2      0.0  0.0  0.0  1.0  0.0  0.0      10.0
3      0.0  0.0  1.0  0.0  0.0  0.0      -4.0
4      0.0  0.0  0.0  0.0  0.0  0.0       0.0
...    ...  ...  ...  ...  ...  ...       ...
49995  0.0  0.0  1.0  0.0  0.0  0.0      -5.0
49996  0.0  0.0  0.0  0.0  0.0  0.0     119.0
49997  0.0  0.0  0.0  0.0  0.0  0.0      -3.0
49998  0.0  0.0  0.0  0.0  0.0  0.0      -9.0
49999  0.0  1.0  0.0  0.0  0.0  0.0      -2.0
```

Feature Selection



Features Selected	Score
weatherDelay	0.06961472481
carrierDelay	0.139108085
nasdelay	0.0898226848
SecurityDelay	0.0001111020107
LateAircraftDelay	0.2597882169
Month	0.006998284042
Origin	0.007446575209
Dest	0.005141299766
UniqueCarrier	0.006039788912



Not Selected	Score
DayofMonth	0.002026315179
Distance	0.0005194320672
DayOfWeek	0.0008655544318
Diverted	0.0002815180716
TaxiIn	0.0009594809797
TailNum	0.0018245151
CRSDepTime	0.01231459712
CRSElapsedTime	0.0007187784523
CRSArrTime	0.004617141762
FlightNum	0.001992471424

Choosing a K value

K	Score
50	0.4411189053
100	0.4438037517
150	0.4420239524
200	0.4396128147
250	0.4391826344
300	0.4362385688
350	0.4339085182
400	0.4314667864
450	0.4287227357
500	0.4249183134

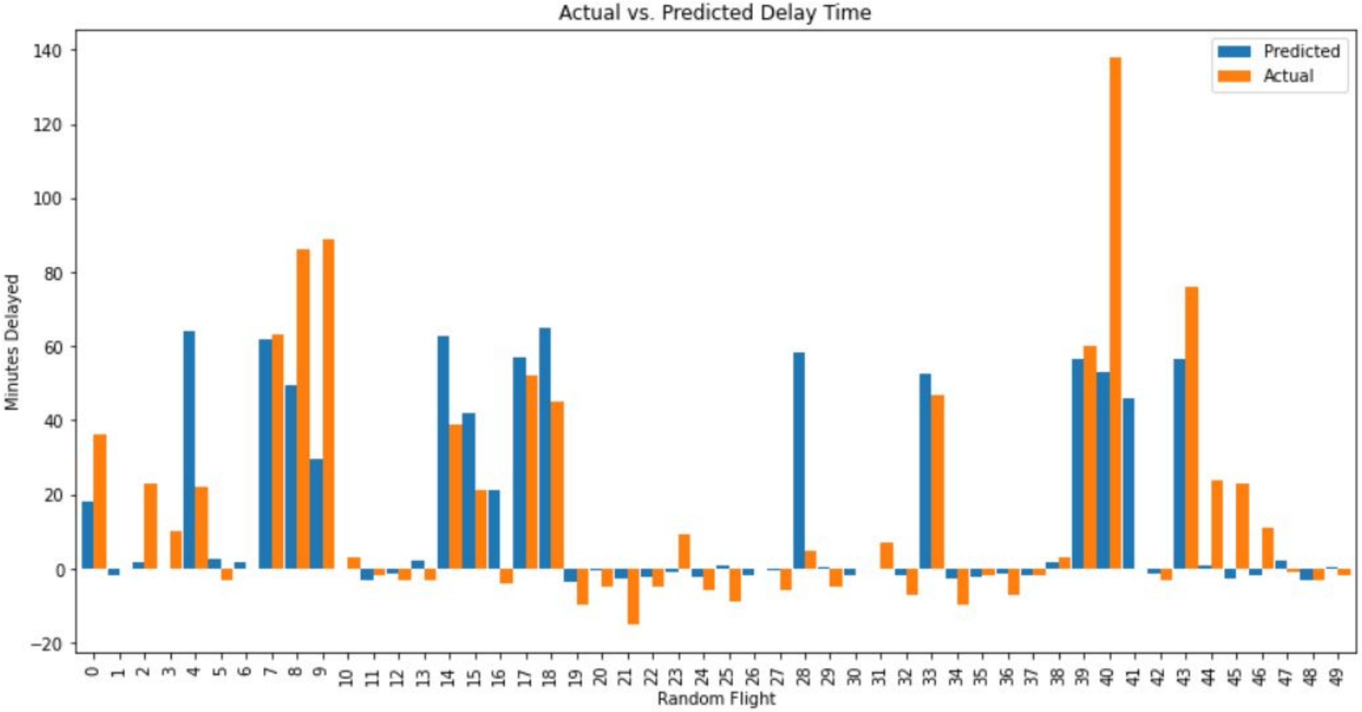
Our Code

https://colab.research.google.com/drive/1_PWjD0uwC1LcRnNCNbHjQYEiL2ySPTTd?usp=sharing

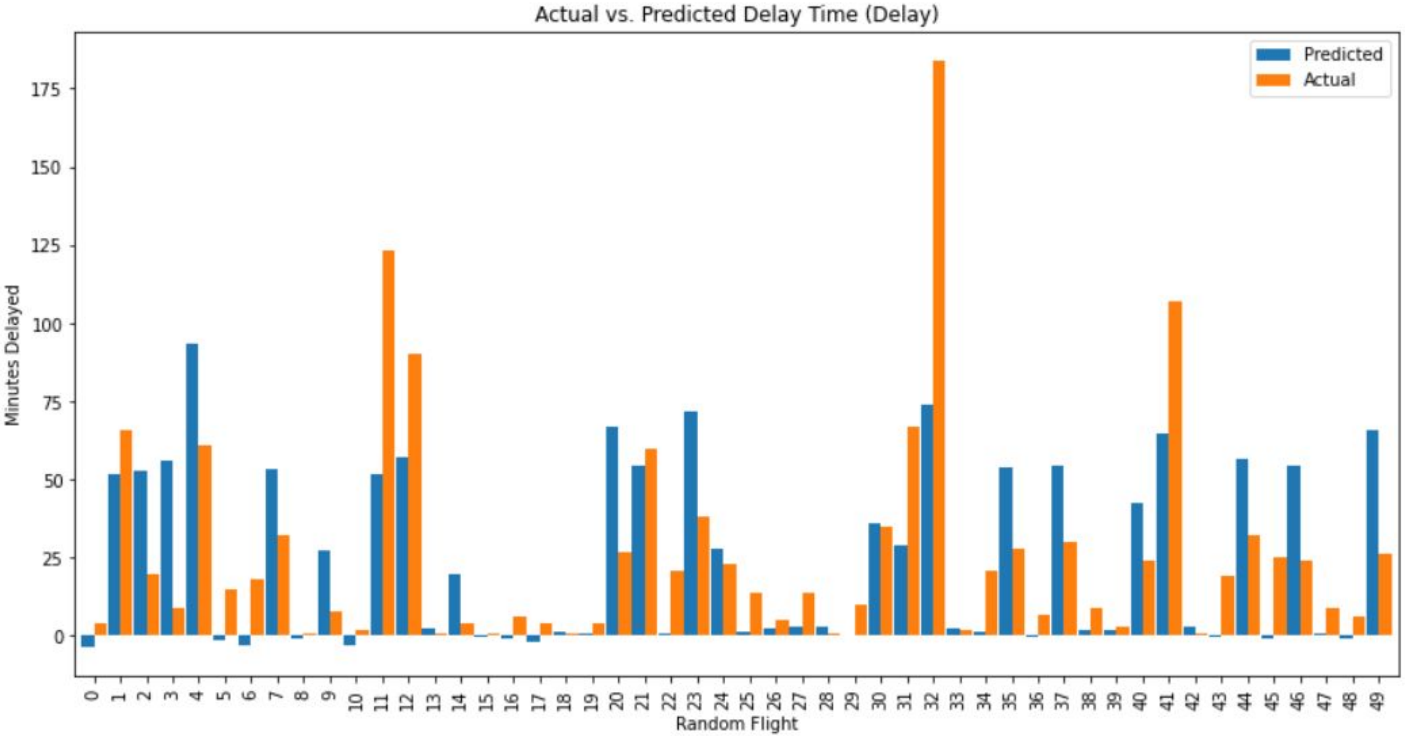
Validation

- We used the Sklearn score function to measure the accuracy of our model
- The score function returns a value from 0 to 1 representing the coefficient of determination of the prediction
- We used a test set that used 30% of our total data
- Our model achieved a score of 0.44

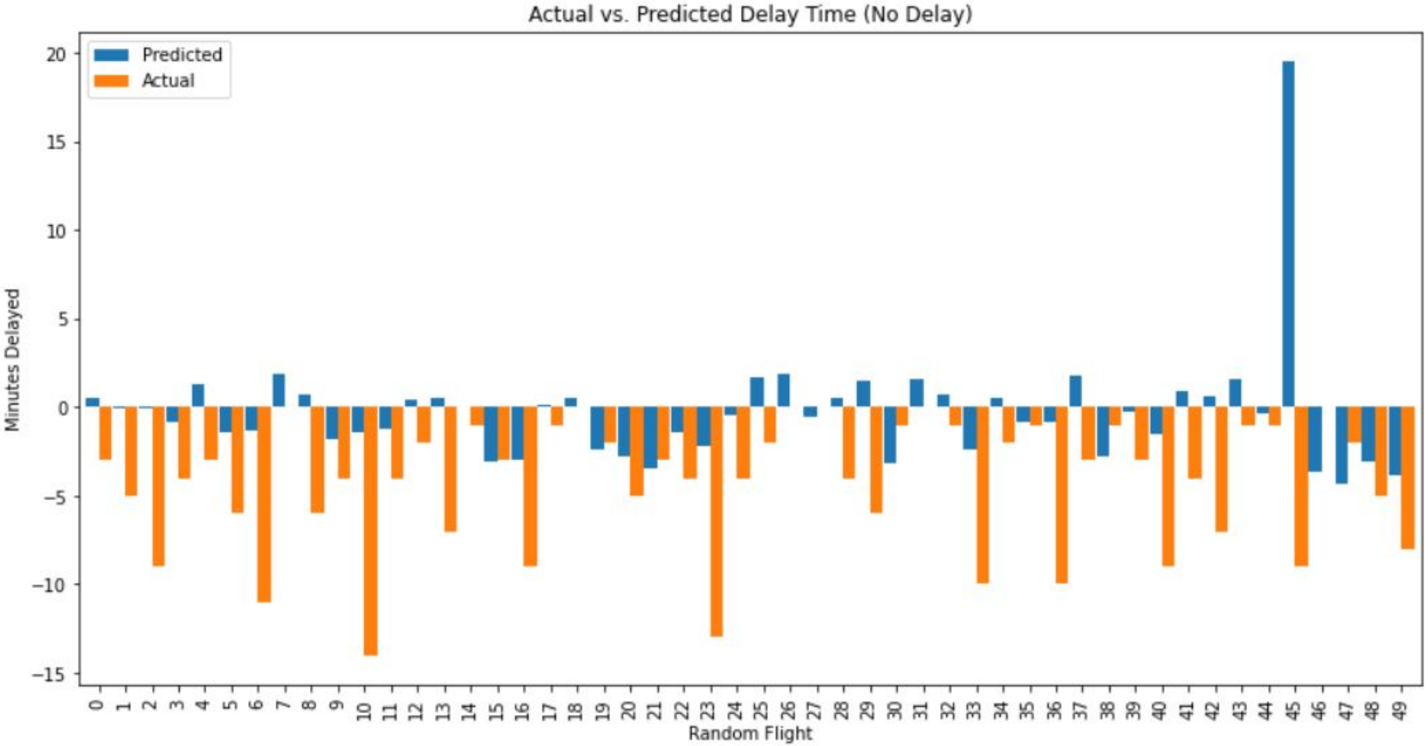
Prediction Visualization



Only Delayed Flights



Only Non Delayed Flights



Limitations

- KNN is a slow algorithm
    - Does not work well with large datasets and high dimensional datasets
    - Only 50000 total entries considered
- Sensitive to noise and outliers
- Prone to Overfitting