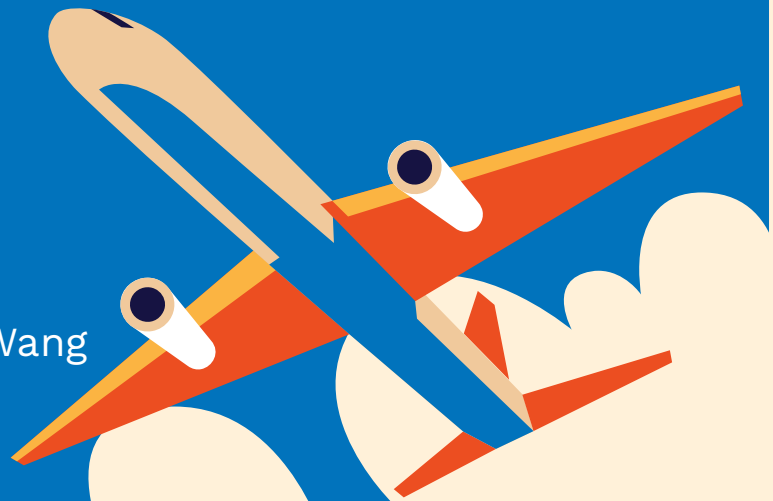
Predicting Flight Delays

Andrew Reilly, Bryce Sheedy, Damien Klein, Jiesen Wang



Summary of topic

- There’s nothing quite like arriving at your assigned gate only to hear your flight has been delayed
- What if we could predict whether a given flight will be delayed?
- The aim of our project is to do just that. Our cleaned dataset has around 58,000 observations and ten different features
- These include airline, departure airport, the day of the week the flight occurred, the length of the flight, and whether the flight was delayed or not.

Methods

- We trimmed the large dataset to only include flights from the top 3 airports and top 4 airlines that represented the majority of the data
- Airports included ATL (Atlanta), ORD (Chicago), and DFW (Dallas/Fort Worth)
- Airlines included American, Delta, Envoy, and ExpressJet
- Created dummy variables for the airline and departure airport columns
- We used grid search to test a variety of models to predict a delayed flight
- Logistic regression, decision tree classification, kNN, and stochastic gradient descent (SGD)

Why These Methods?

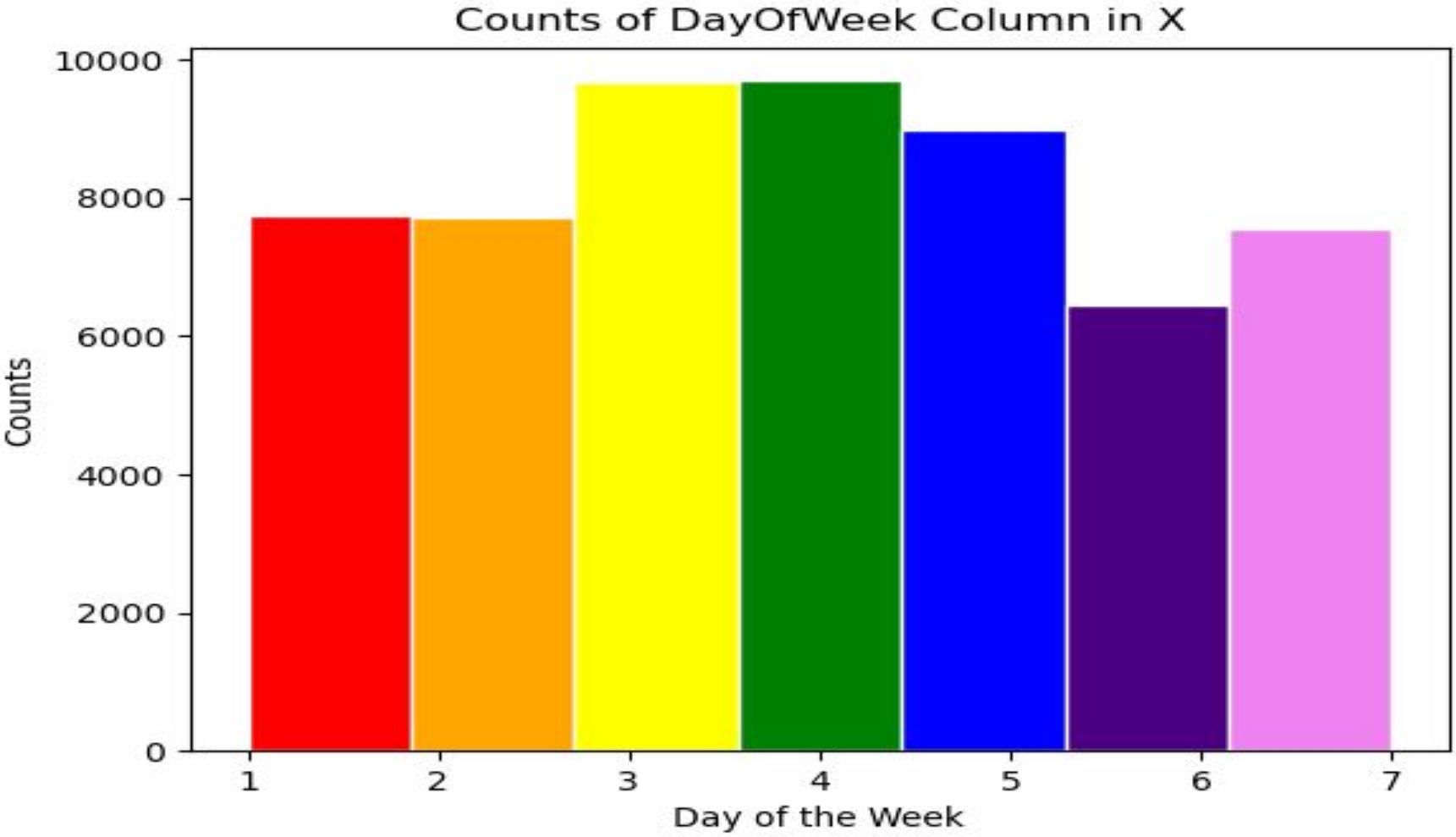
- Since the response variable was binary, we thought that trying SVC and logistic regression seemed reasonable
- We also chose to try using a decision tree because there were multiple features on which to build the tree
- kNN classification made sense to try because of how large the dataset actually was
- Stochastic gradient descent wasn't in our original list of choices, but we followed the Scikit-learn flowchart with our data and it suggested using SGD as an algorithm

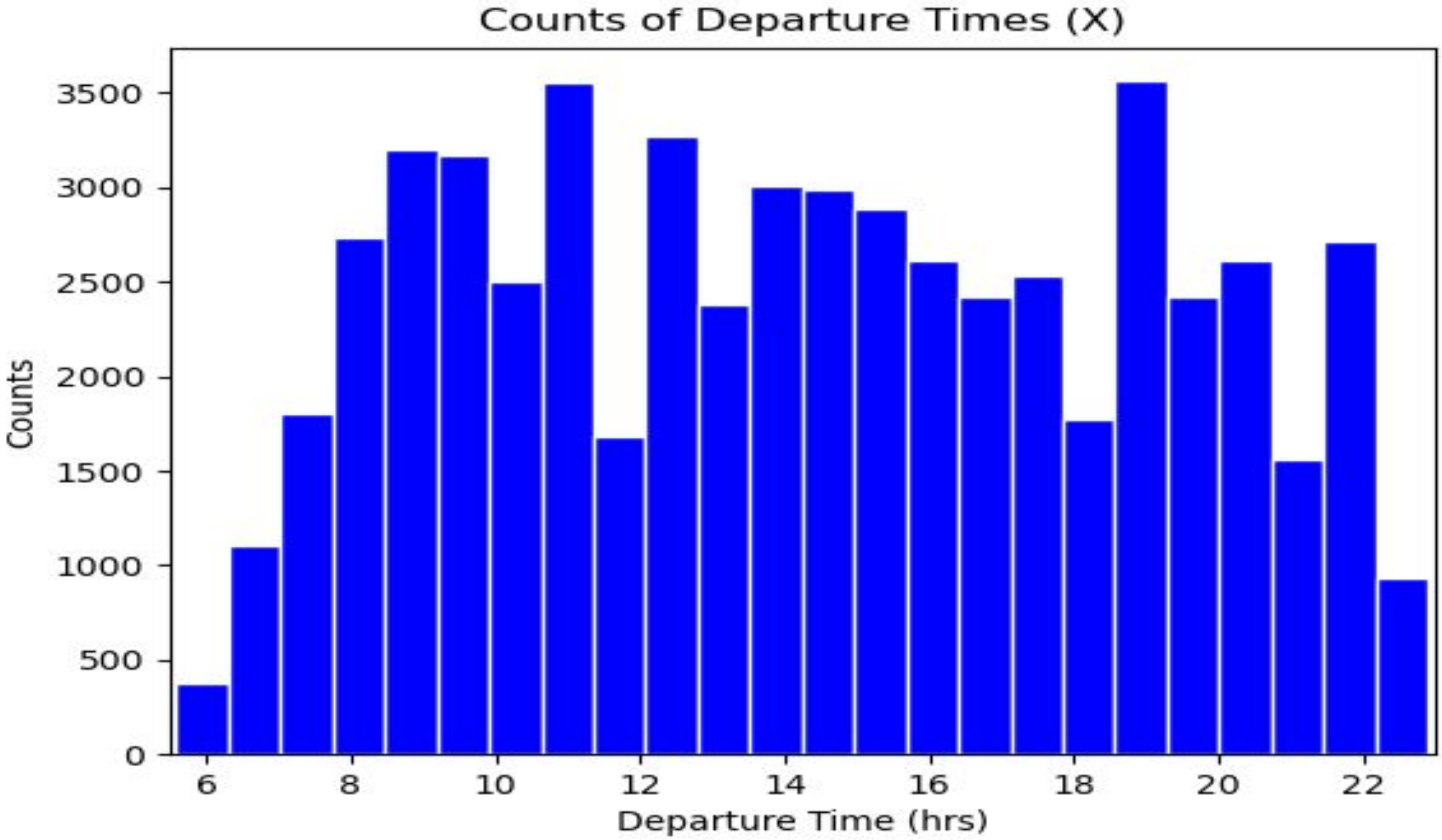
Support Vector Machine

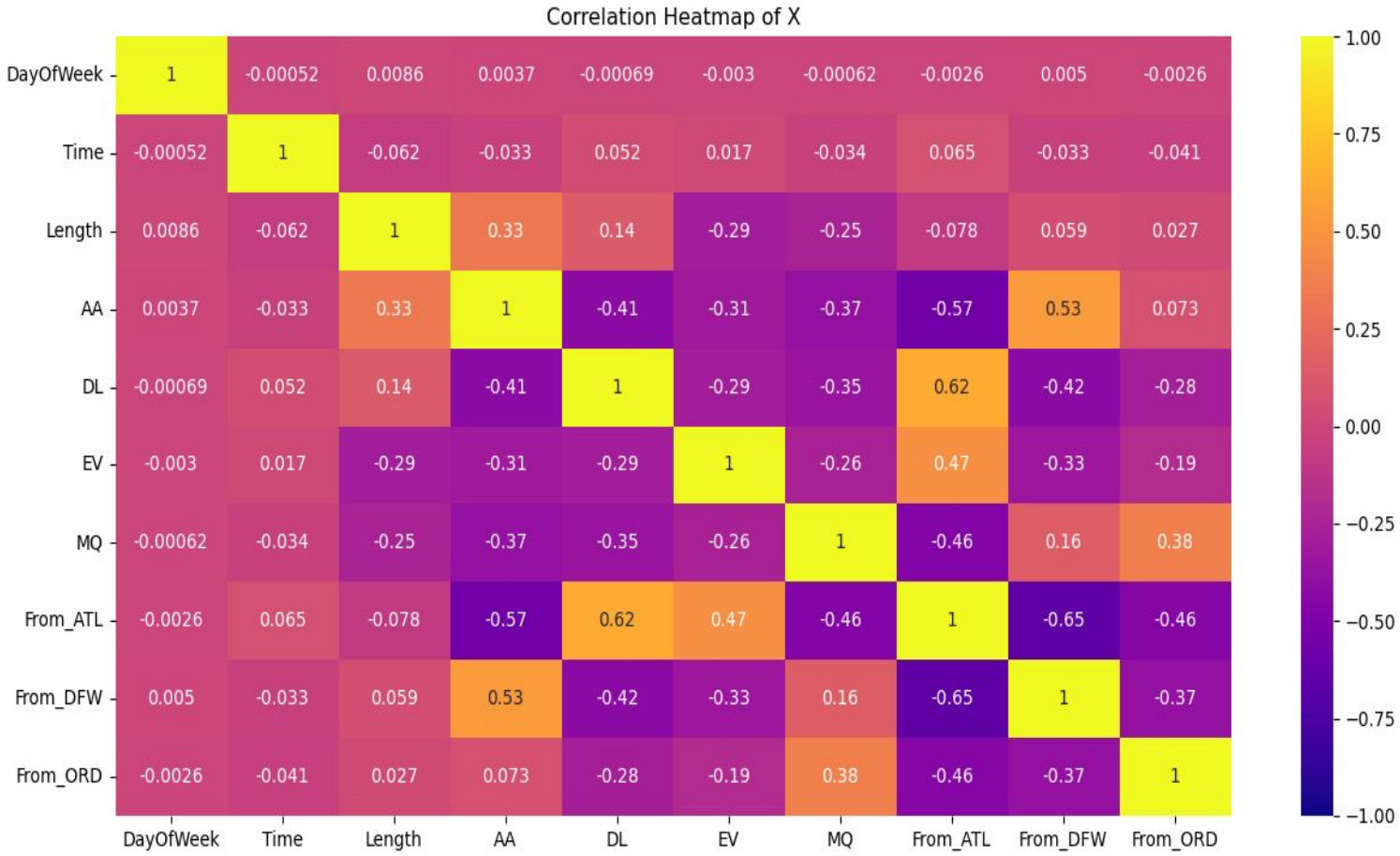
- Originally we had also included a Support Vector Classifier as a possible algorithm to use
- Our attempts to use it led to extremely slow (and unfinished) calculations
- Even using multiple cores did not speed up the process
- We reasoned that this was likely due to the fact that since X is 10-dimensional, the SVM was trying to search for the best 9-dimensional hyperplane for ~58,000 observations (which would understandably take a very long time)
- After numerous attempts using smaller and smaller subsets of the original data, we decided to not use SVM as an algorithm

Additional Points of Interest

- In the uncleaned dataset, 1 is the value of the response variable around 44.54% of the time
- In our cleaned dataset, 1 is the value of the response variable around 44.4% of the time
- We ideally want an algorithm that achieves a score of at least 0.444 on the validation dataset
- The parameter values we chose for GridSearch are as follows:
  - Logistic Regression: penalty = ‘l2’, max_iter = 5000, C = {0.01, 1, 100}
  - kNN: n_neighbors = {5, 10, 15}, metic = {‘euclidean’, ‘manhattan’}
  - Decision Tree: max_depth = {10, 20, 30}, criterion = ‘entropy’
  - SGD: loss = ‘hinge’, penalty = ‘l2’, max_iter = 5000, alpha = {0.01, 0.001, 0.0001}

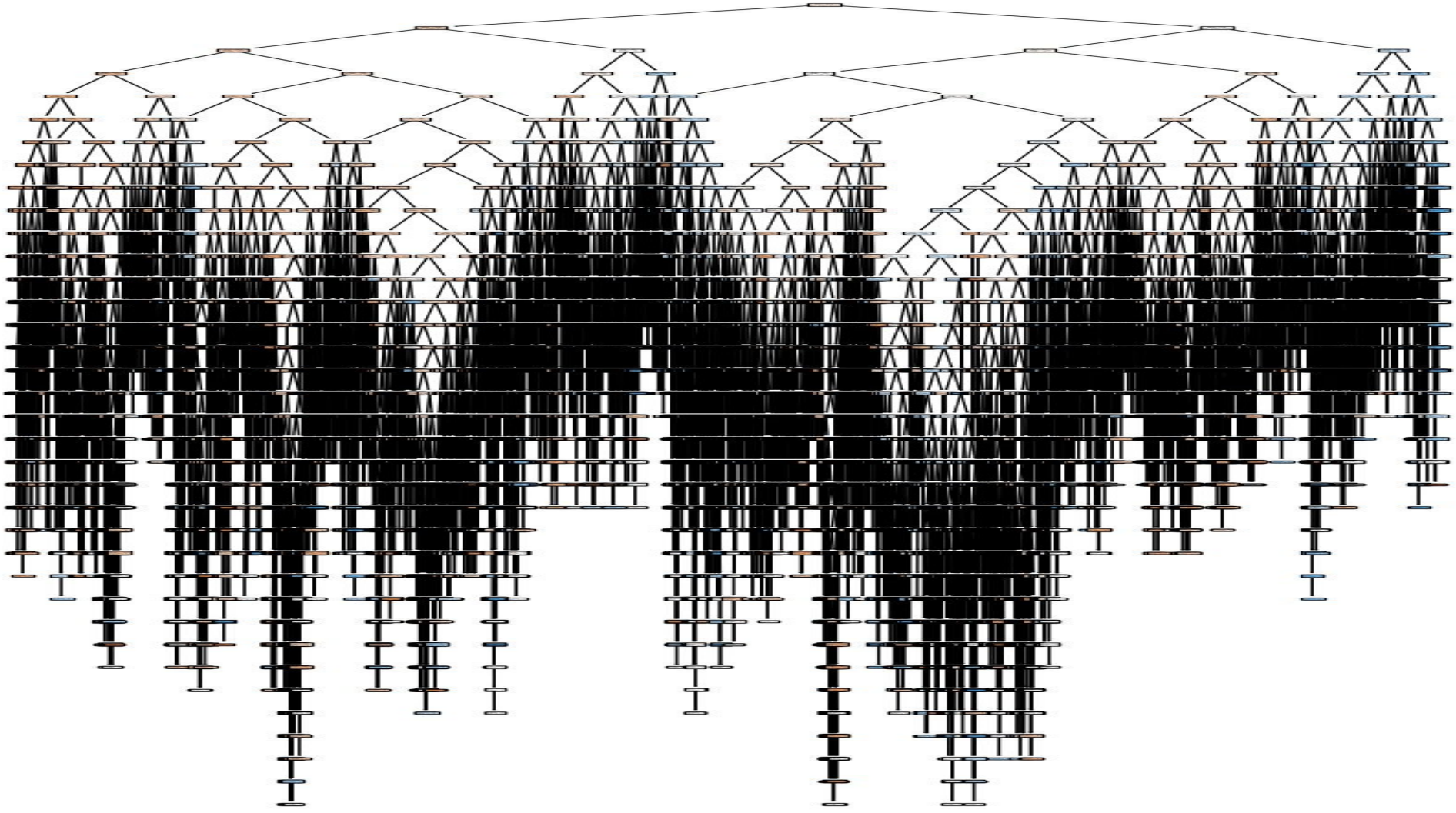

Counts of DayOfWeek Column in X
10000
8000
6000
4000
2000
0
Counts
1
2
3
4
5
6
7
Day of the Week

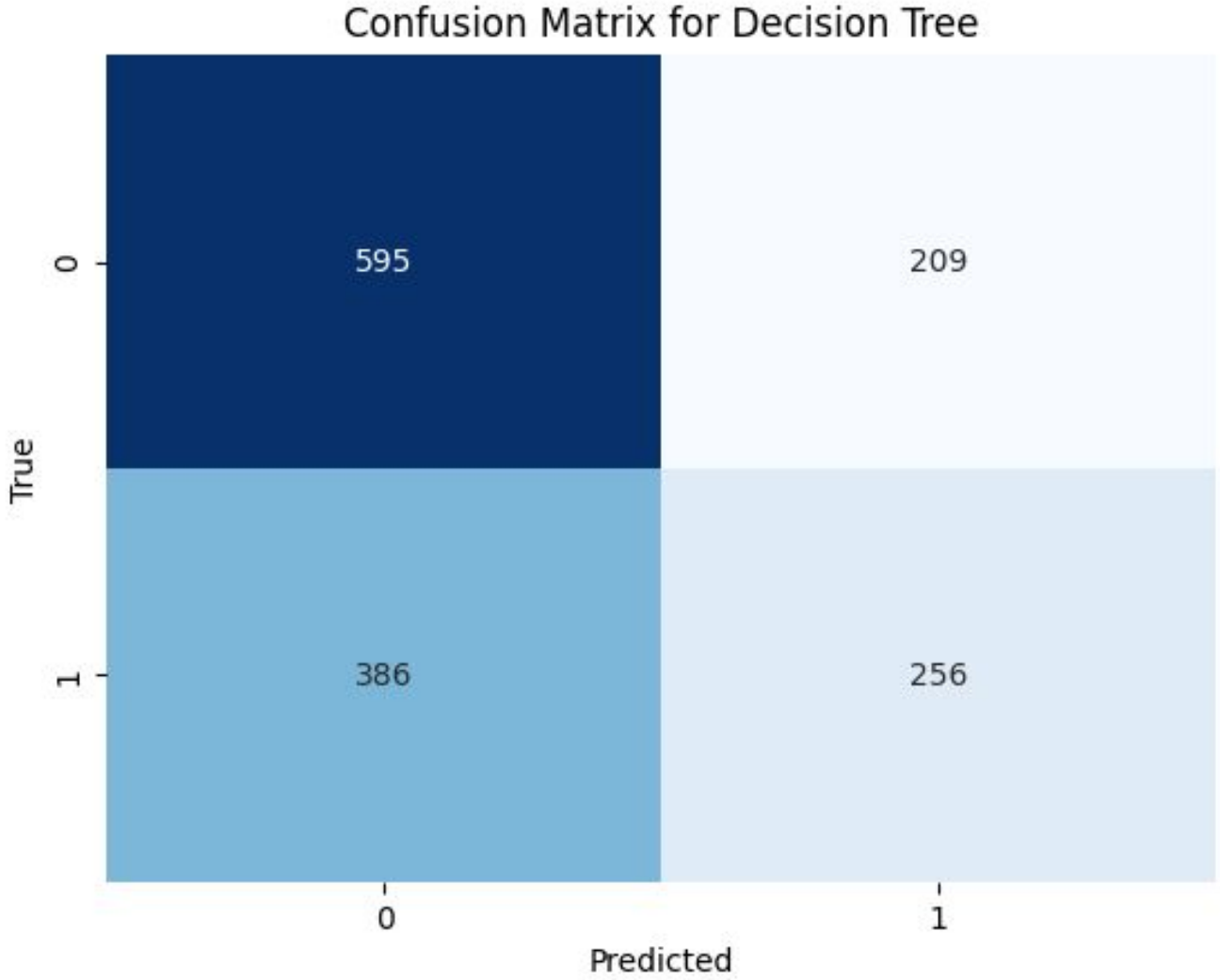

Counts of Departure Times (X)
Counts
0
500
1000
1500
2000
2500
3000
3500
6
8
10
12
14
16
18
20
22
Departure Time (hrs)


Correlation Heatmap of X
DayOfWeek 1 -0.00052 0.0086 0.0037 -0.00069 -0.003 -0.00062 -0.0026 0.005 -0.0026
Time -0.00052 1 -0.062 -0.033 0.052 0.017 -0.034 0.065 -0.033 -0.041
Length 0.0086 -0.062 1 0.33 0.14 -0.29 -0.25 -0.078 0.059 0.027
AA 0.0037 -0.033 0.33 1 -0.41 -0.31 -0.37 -0.57 0.53 0.073
DL -0.00069 0.052 0.14 -0.41 1 -0.29 -0.35 0.62 -0.42 -0.28
EV -0.003 0.017 -0.29 -0.31 -0.29 1 -0.26 0.47 -0.33 -0.19
MQ -0.00062 -0.034 -0.25 -0.37 -0.35 -0.26 1 -0.46 0.16 0.38
From_ATL -0.0026 0.065 -0.078 -0.57 0.62 0.47 -0.46 1 -0.65 -0.46
From_DFW 0.005 -0.033 0.059 0.53 -0.42 -0.33 0.16 -0.65 1 -0.37
From_ORD -0.0026 -0.041 0.027 0.073 -0.28 -0.19 0.38 -0.46 -0.37 1
DayOfWeek Time Length AA DL EV MQ From_ATL From_DFW From_ORD
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00

Results

- Grid search determined the decision tree method was best for our data
- Accuracy on the test data was 58.9%. Precision was 55.1% and recall was 39.9%
- Logistic regression, kNN, and SDG models had similar accuracies, around 56-58%
- These scores are still better than just guessing 1 each time (which would correspond to a score of around 44%).




Confusion Matrix for Decision Tree
595
209
386
256
0
1
True
0
1
Predicted

Future Directions/Revisions

- If we had more time, we would first try using SVM unimpeded
- We would also want to try and fully visualize the data and how each classifier separated the data
- Experimenting with less features would facilitate this, but at an unknown cost to accuracy and overall predictability
- Since we know that SVC was causing issues with its needed time, utilizing more rows might provide a boost to our overall score
- Trying more hyperparameter options in Grid Search to further cover levels which might be more applicable to a dataset of this size