CS 613 Final Project

Doug Woodward

**Project Brief and Scope of Work**

For this project flight performance data was pulled from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS)(<https://www.transtats.bts.gov/ot_delay/ot_delaycause1.asp>).

Five models were trained on the data to predict if a flight was going to be delayed or not. Theses models are K-Nearest Neighbors, Logistic Regression, Linear Discriminant Analysis, Support Vector Machine, and a Neural Net. An additional model - Decision Tree was also fit. Each model was given the same subset of data, split with a 70/30 train/test split. Additional modifications were made to the data as required by each model.

Each model and the results obtained for the test data is detailed below, with model specific specifications, the accuracy, and the confusion matrix each outlined along with some brief notes. More analysis was conducted during the report, including accuracy on training data as well as hyperparameter tuning. The code is attached to this report as well for reproducibility.

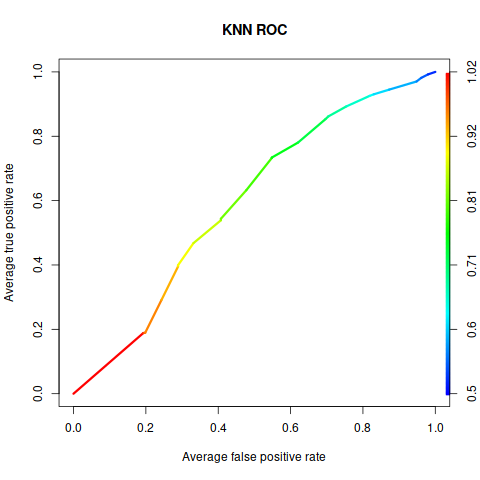
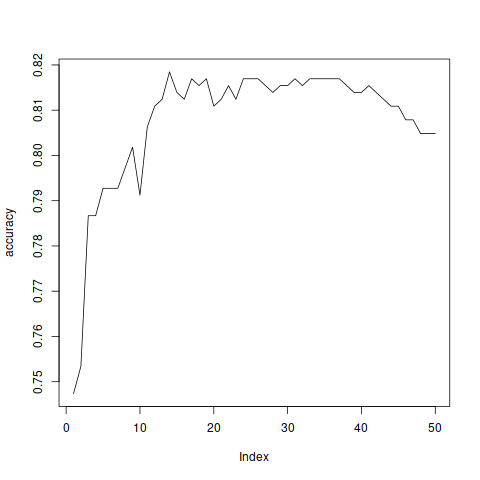
**Analysis and Conclusion**

Following the analysis of each model, is can be see that the K-Nearest Neighbors model performs the best out of all models. This may be surprising given the relatively simplistic nature of KNN, but the nature of the data makes this unsurprising. Because this is a binary classification problem, and two of the the most influential variables, Carrier and destination are also categorical, this is expected. Though KNN performs best for the given data, it does have low bias and high variance, and thus may be overfit to unseen data outside of this test set, because of that in a real world situation, logistic regression would be a better choice for unseen data as it is less likely to overfit.

It must also be noted that by simply guessing that a flight will always be on time, we can correctly classify the flight 80% of the time. Given this, none of our models perform particularly well on this data set. This means that none of the models here are producing much of an improvement at all, which is shown in the ROC curves as well, indicating that most are simply assuming a flight is on time unless given some few combinations of the predictor variables. Because of this, and the indication from the coefficients of Logistic Regression showing that the most influential variables are destination(categorical), scheduled time, and carrier(categorical), a decision tree may actually be the best choice as it gives 83% accuracy.

**KNN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **KNN** | K = 14  (see figure below left for accuracy/K value) | 0.8184569 | delayed ontime  delayed 20 6  ontime 115 520 |



**Logistic Regression**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **Logistic Regression** | Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) 1.5403 0.2965 5.194 2.06e-07 \*\*\*  carrier 0.6824 0.2415 2.826 0.00472 \*\*  dest 0.5786 0.1539 3.760 0.00017 \*\*\*  origin 0.2237 0.2552 0.877 0.38067  schedtime -1.2001 0.2503 -4.795 1.63e-06 \*\*\*  weather -14.1642 324.7438 -0.044 0.96521  date -0.3076 0.1975 -1.557 0.11937 | 0.8154312 | delayed ontime  delayed 13 0  ontime 122 526 |

|  |  |
| --- | --- |
| We see that given the p-values, in decreasing order carrier, scheduled time, distance, origin, and day of month have significance to if a flight is delayed.We note that carrier is positively correlated with delay, and scheduled time is negatively correlated via the coefficients. Straight guessing of ontime for each flight gets 0.8055429 accuracy. So linear regression yields only a 1%-2% increase. |  |

**LDA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **Linear Discriminant Analysis** |  | 0.8154312 | delayed ontime  delayed 13 0  ontime 122 526 |

|  |  |
| --- | --- |
| QDA does not work given our data, due to having categorical variables such as carrier. Because QDA assumes multivariate normality of the data, computing given categorical variables that violate this assumption does not work. In fact, LDA is a poor choice as well given that even though it can disregard the need to multivariate normality and carry out a computation with categorical variables, it is a lesser version of logistic regression.  Identical results to using Logistic Regression, which is not that surprising. |  |

**SVM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **Support Vector Machine** |  | **Linear Kernel**  0.8154312  **Polynomial Kernel**  0.8093797  **Radial Kernel**  8154312 | **Linear** delayed ontime  delayed 13 0  ontime 122 526  **Polynomial** delayed ontime  delayed 9 0  ontime 126 526  **Radial** delayed ontime  delayed 13 0  ontime 122 526 |

|  |  |
| --- | --- |
| The linear kernel is probably our best option. But SVM is still not any better than logistic regression. |  |

**Neural Net**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **Neural Net** | No hidden layers as we have linearly separable data so a perceptron will converge. | 0.8154311649 | delayed ontime  delayed 13 0  ontime 122 526 |

|  |  |
| --- | --- |
| Neural nets are not the best option for this data set, they do work, and do achieve equivalent accuracy to other methods, but given that the data appears to be linearly separable, there is no need for such a complex method. |  |

**Decision Tree**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Specifications** | **Accuracy** | **Confusion Matrix** |
| **Decision Tree** |  | 0.8305598 | delayed ontime   delayed 44 21   ontime 91 505 |

|  |  |
| --- | --- |
| Decision Tree works best out of all models as it can handle the large influence of categorical variables, however, it may be overfit to unseen data. |  |