Data Solution for Flight Delay

Based on machine learning models

Authors

Songlin YANG Team Leader
Qianting YANG Data Engineer
Shijia RONG Data Scientist
Xinyi JI Business Analyst

Dec 2021

Contents

Contents of Exhibits and Tables

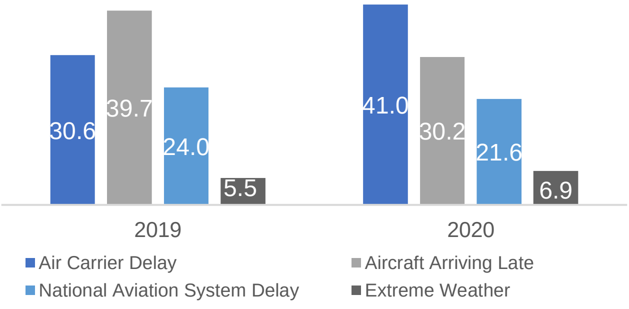
The Headache

Although the aviation infrastructures of the United States are well-equipped and advanced, deficiency problems still exist in its air transport system. The average on-time performance of American airlines fell in 2019 for the third year in a row (Puckett, 2020). Our company considers the U.S. Department of Transportation (DOT) is in a significant and probably the most suitable position to address such issue.

As a department of the government, DOT utilises the data from the Bureau of Transportation Statistics to ensure the efficiency of the air transport system in American and protect the rights of passengers from affecting by lengthy tarmac delays. While airline companies usually offer compensation policies to passengers for the delay of their flights, DOT has the authority to enact and enforce relevant regulations such as the Tarmac Delay Rule (TDR) to reduce flight delays in the aviation system. However, the policy may fail to address the issue effectively as some airline companies will cancel the flights to avoid heavy penalties on delay, leading to extra costs for passengers (Fukui et al., 2014). In addition to the inconvenience brought to passengers, the flight delay issue would also create a substantial financial burden on other parties. For airlines companies, the extra operational costs in retaining crew and aircraft emerge. In addition, the insurance company will suffer from increased expenditure on the delay damages.

Therefore, to proactively improve the current situation, we decide to apply a machine learning model to predict whether the flight will delay or not. Moreover, there are four potential factors related to this delay issue: air carrier, aircraft arriving, National Aviation System (airport operations and air traffic control conducted by NAS that result in flight delay), and extreme weather (BTS, 2021). (Exhibit 1)

Exhibit 1 Delay Cause by Year, Percent of Total Delay Minutes (%)



Source: Bureau of Transportation Statistics

Historical Data Exploration

We are given the historical dataset containing flight information for 50 thousand individual flights scheduled to depart during January 2019 in the U.S. The characteristics of this dataset a shown in Exhibit 2.

50k
Number of observations

0
Missing value

0.2%
Duplicate rows

24
Number of variables

5
categorical variables

19
Numeric variables

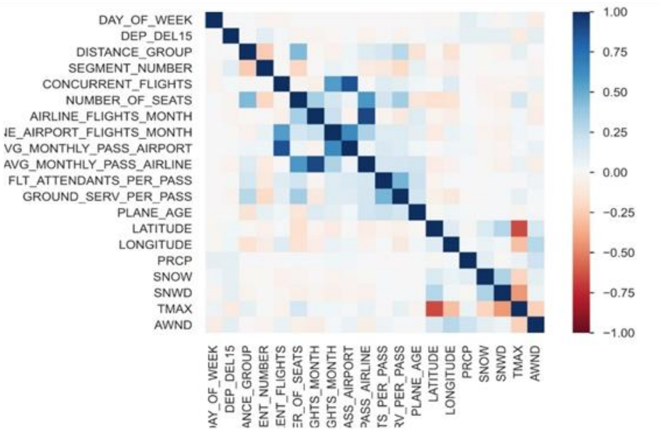


Exhibit 2 Overview of the dataset (left) and correlation (Pearson) heat map (right)

These flights recorded belong to 17 carriers and departed from 84 different airports, and 81.77% of them were short-haul (DISTANCE_GROUP < 6). The most frequent departure time was the morning, with 26% of the whole observations, while only 2% left in the early morning. Overall, 17.34% of the flights were delayed over 15 minutes.

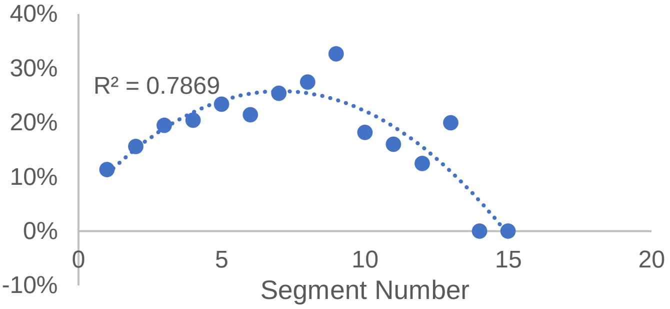


Exhibit 3 Relationship between segment number and delay rate

As mentioned in last section, former flight arriving late is one of the main cause of departure delay for the next flight. Therefore, the segment number of the flight is crucial to the target variable. As Exhibit 3 shows, delay rate increases with segment number increasing, which can be explained as a cumulative effect. After 10, delay rate decreases due to lack of data instances.

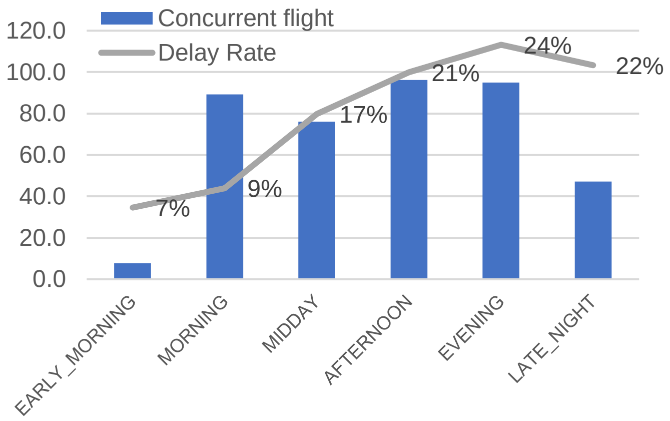


Exhibit 4 Delay rate and average concurrent flight of a departure block

Moreover, the Departure Time also matters, which can be seen from Exhibit 4 that the delay rate in the evening (24%) is three times more than that in the early morning (7%). This factor may be associated with the average number of concurrent flights in each time period. Exhibit 4 illustrates a trend that the time period with a higher number of concurrent flights also displays a higher delay rate.

In addition, features related to extreme weather such as high level of precipitation, heavy snowfall, and low maximum temperature are also significant factors in delay rate. For instance, as we can see from Exhibit 5, high level of precipitation can lead to higher delay rates. In terms of their relationship with other input features, since latitude and longitude are significant factors affecting the local climate, delay rates are high in locations with high level of precipitation, such as the eastern coast area shown below.

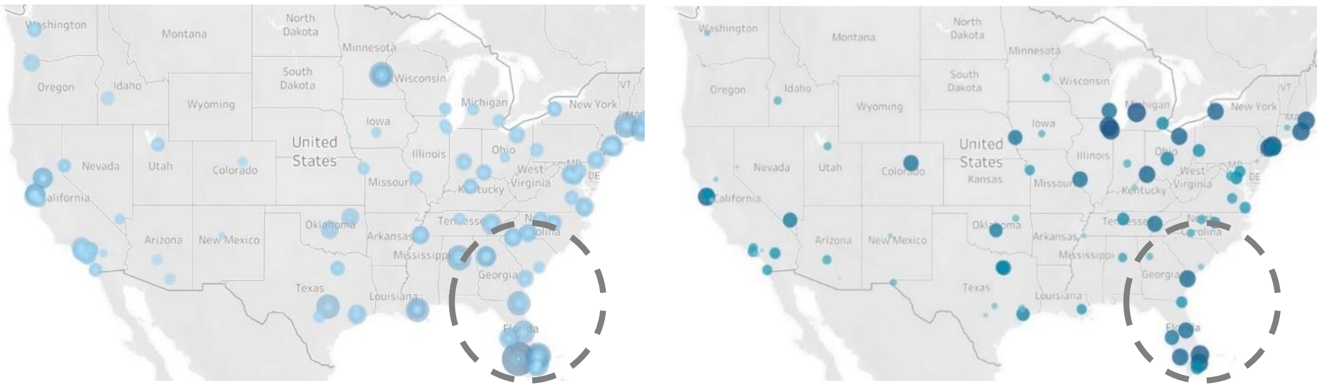


Exhibit 5 Map charts of precipitation (left) and delay rate (right) by different departing airport

Model Building and Evaluation

We present three machine learning classification algorithms to build a predictive model on whether a flight will delay or not: *k*-NN (*k*-Nearest Neighbours), Logistics Regression and Random Forest, which are all suitable to a binary prediction problem.

70% of the dataset was split out as a training set, with the remaining for testing unseen situations. A 10-fold stratified cross-validation is used among the training set to train the model. After the split, we preprocessed the data to get a more algorithm understandable format and conduct feature selection by R (See Appendix I). This step done after the split is to avoid information leakage.

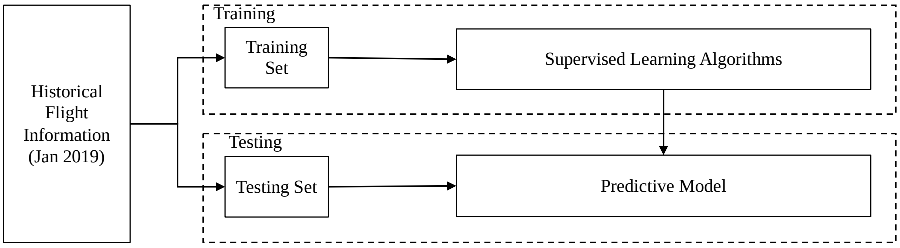

Historical Flight Information (Jan 2019)
Training
Training Set
Supervised Learning Algorithms
Testing
Testing Set
Predictive Model

Exhibit 6 Workflow for building and implementing a classifier

***K*-Nearest Neighbours** is a non-parametric method, which doesn't consist of explicit learning steps and can constantly evolve with new data instances. The object is assigned to the class most common among its nearest neighbours. *K*-NN has three important hyperparameters. The number of nearest neighbours is the most influential one, which is preferred to be set as an odd number. To tune this, we plotted the model AUC and f1-score with respect to the numbers of neighbours. We decided to set it at 23 since diminishing return set in on AUC and f1-score start to be stable after that (See Appendix II for codes).

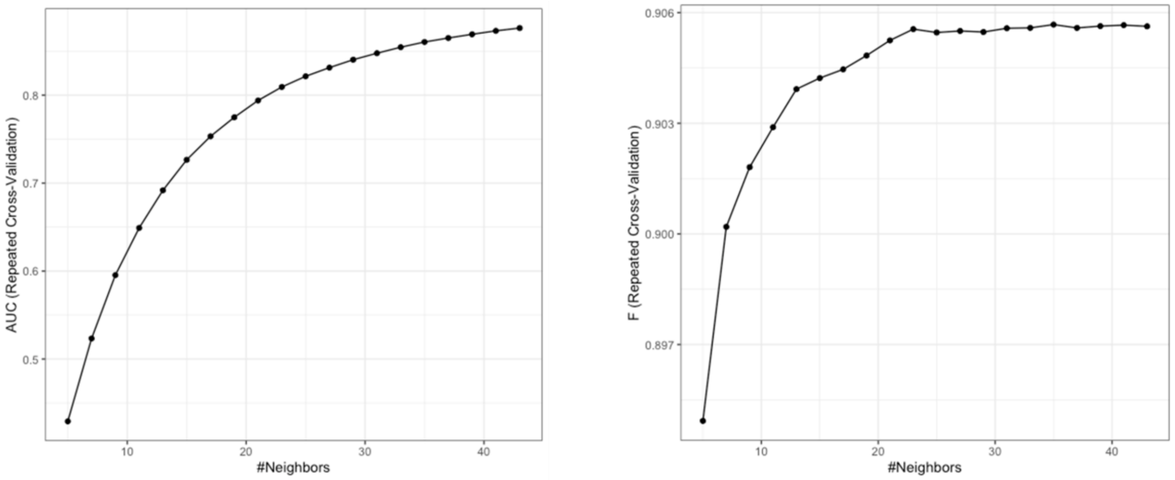


Exhibit 7 Model performance by AUC (left) and f1-score(right) with changing #neighbours

Logistic Regression is used to model the probability of a binary event. The merit of this model is to control overfitting issues by applying regularisation. This algorithm has two main hyperparameters, regularisation type (L1 and L2) and cost strength. The regularisation type L2 works well when the weights of input features are nearly equal in size and all impact the outcomes (Chollet, 2017). This is not the case in our dataset. L1 type is applied to discard the insignificant features and select more relevant ones to compress and regularise the model. In our case, we set the cost strength as default (C = 1).

Random Forest builds multiple decision trees by taking a subset of observations and variables and integrating the results to make a stable and precise prediction. Such an ensemble learning method can overcome the overfitting and bias issue than a single Decision Tree. We have tuned the number of trees included in the forest as 40, given a trade-off between better performance and calculation efficiency (Srivastava, 2015). We have set the number as the default 5 (calculated as the square root of the total number of features) since the performance is not much affected by increasing the number of features in each tree in our case. Similarly, the minimum depth of individual trees and minimum split are also set as the defaulted 3 and 5 accordingly.

Compared with the baseline naive model (constant), which predicts all outcomes as negative (non-delayed), all other three models have a fair good accuracy score with cross-validation (Exhibit 8). Given that this dataset is imbalanced, with a 4.77 ratio of non-delayed/delayed, we move beyond accuracy, focusing on identifying actual delayed (non-delayed) flights as delayed (non-delayed). Therefore, we use the ROC curve to plot TPR ($\frac{TP}{TP+FN}$) against FPR ($\frac{FP}{FP+TN}$) (Exhibit 9). We expect a higher TPR and lower FPR. Thus, a greater value of area under this curve (AUC) indicates a better model performance. Three models have performed better than the benchmark by assessing the value of both accuracy and AUC.

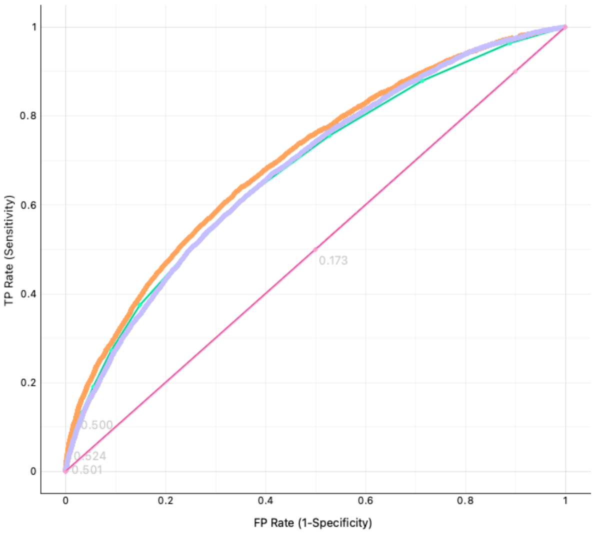


Exhibit 9 ROC curve for 10-Fold Cross-Validation for three models against the benchmark

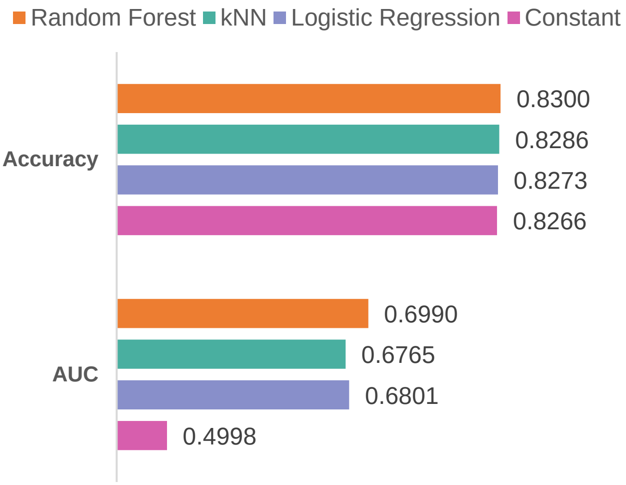


Exhibit 8 10-Fold Cross-Validation result for three models against the benchmark

To further evaluate other issues of applying these classifiers, we offer an MCDA (Multi-criteria Decision Analysis) framework (Table 1). We provide three criteria with different weights. The model performance is the most influential one. LR is quick to train in terms of computational time and complexity, while RF and *k*-NN are relatively time-consuming and computationally expensive. Considering the interpretation ease, *k*-NN and RF is quite intuitive and understandable compared with LR. By putting different weights on different criteria, we found RF with the highest score and recognised it as the winning classifier.

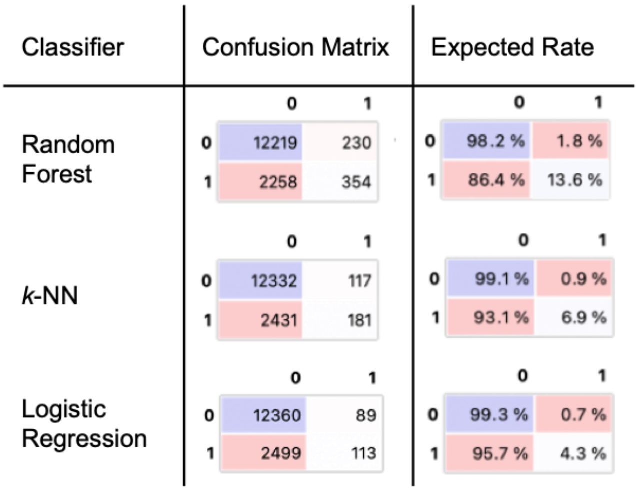
Table 1 MCDA (Multiple-criteria decision analysis) of three classifiers

Criteria	Weight	Random Forest	Logistic Regression	*k*-NN
Performance (AUC on the test set, see Appendix III)	0.7	3	2	1
Computational Ease (calculation time & complexity)	0.2	2	3	1
Interpretation Ease	0.1	2	1	3
Total Score	1.0	2.7	2.1	1.2

Model Implementation and Recommendation

To further evaluate why the Random Forest model is the winning classifier, we use the cost/benefit matrix in the business context. The detailed outcome explanations in the confusion matrix on test set are listed.

	0: Non-delay	1: Delay
0: Non-delay	**TN** (Correctly predict the actual non-delay)	**FP** (Classify the actual non-delay as delay)
1: Delay	**FN** (Classify the actual delay as non-delay)	**TP** (Correctly predict the actual delay)



Classifier	Confusion Matrix			Expected Rate		
		0	1		0	1
Random Forest	0	12219	230	0	98.2 %	1.8 %
	1	2258	354	1	86.4 %	13.6 %
		0	1		0	1
k-NN	0	12332	117	0	99.1 %	0.9 %
	1	2431	181	1	93.1 %	6.9 %
		0	1		0	1
Logistic Regression	0	12360	89	0	99.3 %	0.7 %
	1	2499	113	1	95.7 %	4.3 %

Exhibit 10 Model result on test set in terms of the confusion matrix

When the model obtains the TP outcome, we avoid unforeseen delay costs and thus use the annual delayed cost of 33 billion dollars in 2019 as the revenue (FAA, 2020). Meanwhile, although the incorrect predictions will incur cost, the monetary loss on the FN error is more significant than its FP opposite case (Choi et al., 2017). However, since the exact amounts of misclassification costs are hard to define, the potential costs between FN and FP are compared based on their ratios, which can be reasonably estimated as 2 from empirical research (Choi et al., 2017). Therefore, we set FP's cost as C and FN's cost as $2C$. Then, based on the expected rate matrix and cost/benefit matrix in Table 2, we can calculate the Expected Value of business gain of three models and visualise the linear relationship between EV and C.

Table 2 Cost and Benefit Examples of Different Outcome

Outcome	Revenue Example	Cost Example	Cost/Benefit ($, Billions)
True Positive	Avoidance of unforeseen delay costs	/	33
True Negative	/	/	0
False Positive	/	Unnecessary expense on countermeasures	(C)

False Negative	/	Expense on emergent Air Traffic Control;
		Chaos arrangement;
		Passengers' economic loss;
		Customer loyalty loss
		(Choi et al., 2017).
			$(2C)$

As shown in Exhibit 11, when $C \geq 0$, we can see that the Random Forest model consistently achieves a higher expected value than its counterparts. In addition, when $0 \leq C < 2.46$, the prediction by the Random Forest model can generate profit for clients. In other words, Random Forest tolerate more misclassification costs compared with the other two models. Thus, this model is our winning classifier in the business application as it brings benefits and enhances business value.

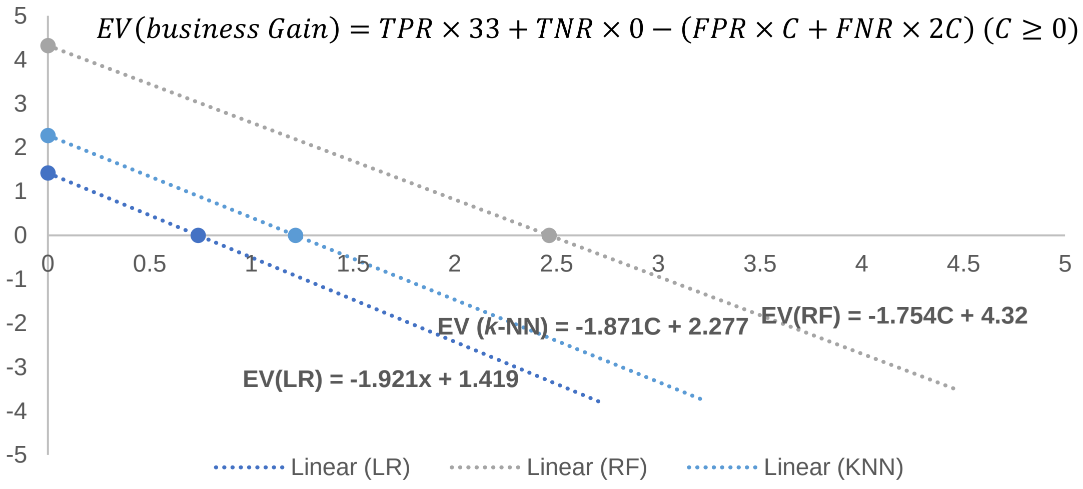
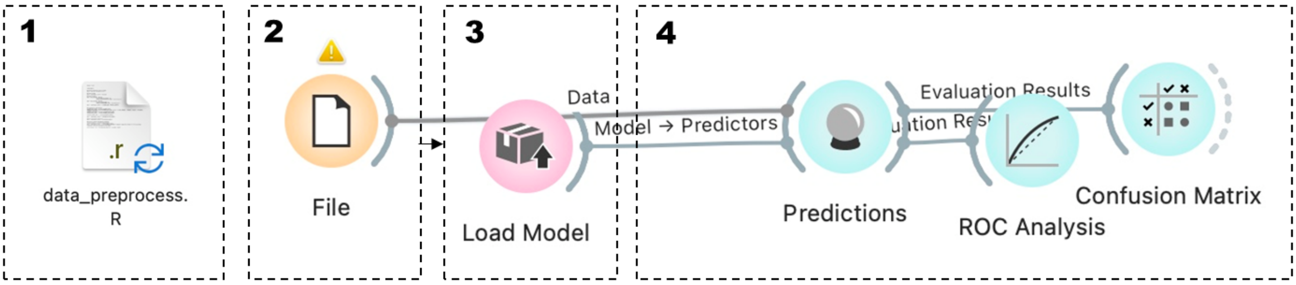


Exhibit 11 Comparison of the expected value of business gain between three models

Details of how to obtain a prediction result by inputting data through our model are described in Exhibit 12. However, there are some limitations of our model. Considering the fickle characteristic of weather, weather forecast data is more suitable than historical data (Priyanka, 2018). Additionally, the data collected from January is limited in scope, and the undetected seasonal difference could also result in an inefficient prediction in delay. Furthermore, some useful features are no contained in our dataset, such as the visibility at departure time and the terminal airport.



Step 1

Preprocess your data with R script

Location
[IDS Coursework] Team 9_Fear of Code/ 3-Relevant Files/data_preprocess.R

Step 2

Load the processed data file

Step 3

Load the saved model

Location
[IDS Coursework] Team 9_Fear of Code/2-Final Model Workflow/ final_model_rf.pkcls

Step 4

Check the model performance score by 'Prediction' widget and visual performance by 'Confusion Matrix' and 'ROC Analysis' widget

Exhibit 12 User instruction for future prediction

In summary, we propose the Random Forest as an effective model to help DOT address the flight delay issue. Through the application of this prediction model, DOT can allocate resources and organize aircraft route more effectively, which in turn benefits all the other stakeholders in the enter transportation system.

References

Bureau of Transportation Statistics (2021) ‘On-Time Performance - Reporting Operating Carrier Flight Delays at a Glance’, *United States Department of Transportation*. Available at: https://www.transtats.bts.gov/HomeDrillChart.asp (Accessed: 26th Nov 2021).

Bureau of Transportation Statistics (2021) ‘Understanding the Reporting of Causes of Flight Delays and Cancellations’, *United States Department of Transportation*. Available at https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations (Accessed: 18th Nov 2021).

Choi.S., Kim, Y.J., Simon.B., Dimitri.M. (2017) ‘Cost-sensitive prediction of airline delays using machine learning’. *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pp.1–8. doi: 10.1109/DASC.2017.8102035. (Accessed: 28th Nov 2021).

Federal Aviation Administration (2020) ‘Cost of Delay Estimates, 2019’, *Federal Aviation Administration*. Available at: https://www.faa.gov/data_research/aviation_data_statistics/media/cost_delay_estimates.pdf (Accessed: 25th Nov 2021).

Fukui, H., & Nagata, K. (2014) ‘Flight cancellation as a reaction to the tarmac delay rule: An unintended consequence of enhanced passenger protection’. *Economics of Transportation*, 3(1), 29-44. Available at: https://doi.org/10.1016/j.ecotra.2014.02.004 (Accessed: 18th Nov 2021).

Priyanka, G. (2018) ‘Prediction of Airline Delays Using K-Nearest Neighbor Algorithm’. *International Journal of Emerging Technology and Innovative Engineering*, Volume 4, Issue 5. Available at: https://ssrn.com/abstract=3340771 (Accessed: 8th Dec 2021)

Puckett, J. (2020) ‘Flight Delays on U.S. Airlines Increased in 2019’. *Condé Nast Traveler*, Available at: https://www.cntraveler.com/story/flight-delays-on-us-airlines-increased-in-2019 (Accessed: 19th Nov 2021).

Srivastava, T. (2019) ‘Tuning the parameters of your Random Forest model’, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/ (Accessed: 29th Nov 2021)

Appendix I: Data Preprocess

The encoding techniques are used as machine learning algorithms show better performance with numerical variables. Training and test sets are processed separately in case of information leakage, but with the same technique.

```
## Target encoding

data = left_join(data, data %>%
                   group_by(PREVIOUS_AIRPORT) %>%
                   summarise(PREVIOUS_AIRPORT_CODED = mean(DEP_DEL15)))

data = left_join(data, data %>%
                   group_by(CARRIER_NAME) %>%
                   summarise(CARRIER_NAME_CODED = mean(DEP_DEL15)))

data <- subset(data, select = -c(CARRIER_NAME, PREVIOUS_AIRPORT))
```

Specifically, we applied target encoding on CARRIER_NAME and PREVIOUS_AIRPORT, in which encoded values are represented as the numeric mean value of the corresponding group of the target variable. In Orange, the default method of encoding a categorical feature is one-hot and ordinal encoding, while the former yields great cardinality and the later yields meaningless alphabetic relationships.

```
## Trigonometric transformation

data$DEP_BLOCK_CODED<- as.factor(data$DEP_BLOCK)
data$DEP_BLOCK_CODED <- factor(data$DEP_BLOCK_CODED,
                               levels   =   c("EARLY_MORNING",   "MORNING",   "MIDDAY",
"AFTERNOON", "EVENING", "LATE_NIGHT"))
data$DEP_BLOCK_CODED <- as.numeric(data$DEP_BLOCK_CODED)
data$DEP_BLOCK_CODED <- sin(2*pi*data$DEP_BLOCK_CODED/6)

data$DAY_OF_WEEK_CODED <- sin(2*pi*data$DAY_OF_WEEK/7)

data <- subset(data, select = -c(DEP_BLOCK, DAY_OF_WEEK))
```

We applied trigonometric functions on periodical features (DEP_BLOCK and DAY_OF_WEEK). For example, set $\mathrm{EARLY}_{\mathrm{MORNING}} = \ sin\left(\frac{1}{6} * 2\pi\right)$ and $\mathrm{LATE}_{\mathrm{NIGHT}} = \ sin\left(\frac{6}{6} * 2\pi\right)$ as sequential blocks.

```
## Remove duplicated information
data <- subset(data, select = -c(DEPARTING_AIRPORT))

## Filter by variance and pearson correlation
correlation_matrix <- as.data.frame(cor(data))
variance_covariance_matrix <- as.data.frame(cov(data))
data <- subset(data, select = -c(FLT_ATTENDANTS_PER_PASS))
```

In terms of feature selection, we firstly filter out FLT_ATTENDANTS_PER_PASS for its variance almost rounding to zero and a low correlation with the target. Next, we removed DEPARTING_AIRPORT as it duplicates the geographical coordinates, LONGITUDE and LATITUDE, which have already been included in our dataset. Noticeably, since *k*-NN and Logistic Regression with regularisation are quite sensitive to the distance between data points, it is necessary to apply to scale, which is done in Orange.

Appendix II: Tune parameters for *k*-NN

```
library(caret)

## Tune k-NN with Caret package in R

data$DEP_DEL15 <- as.factor(data$DEP_DEL15)
levels(data$DEP_DEL15) <- c("No", "Yes")

set.seed(2021)
ctrl <- trainControl(method="repeatedcv",
                         number = 10,
                         repeats = 3,
                         summaryFunction = prSummary,
                         classProbs = T)

knn <- train(DEP_DEL15~., data = data, method = "knn",
                trControl = ctrl,
                preProcess = c("center", "scale"),
                tuneLength = 20,
                metric = "AUC")

knn
ggplot(data = knnfit, metric = "F") + theme_bw()
ggplot(data = knnfit, metric = "AUC") + theme_bw()
```

The above code is used for us to decide how k should be determined and generate Exhibit 7.

Appendix III: List Score of Training and Testing

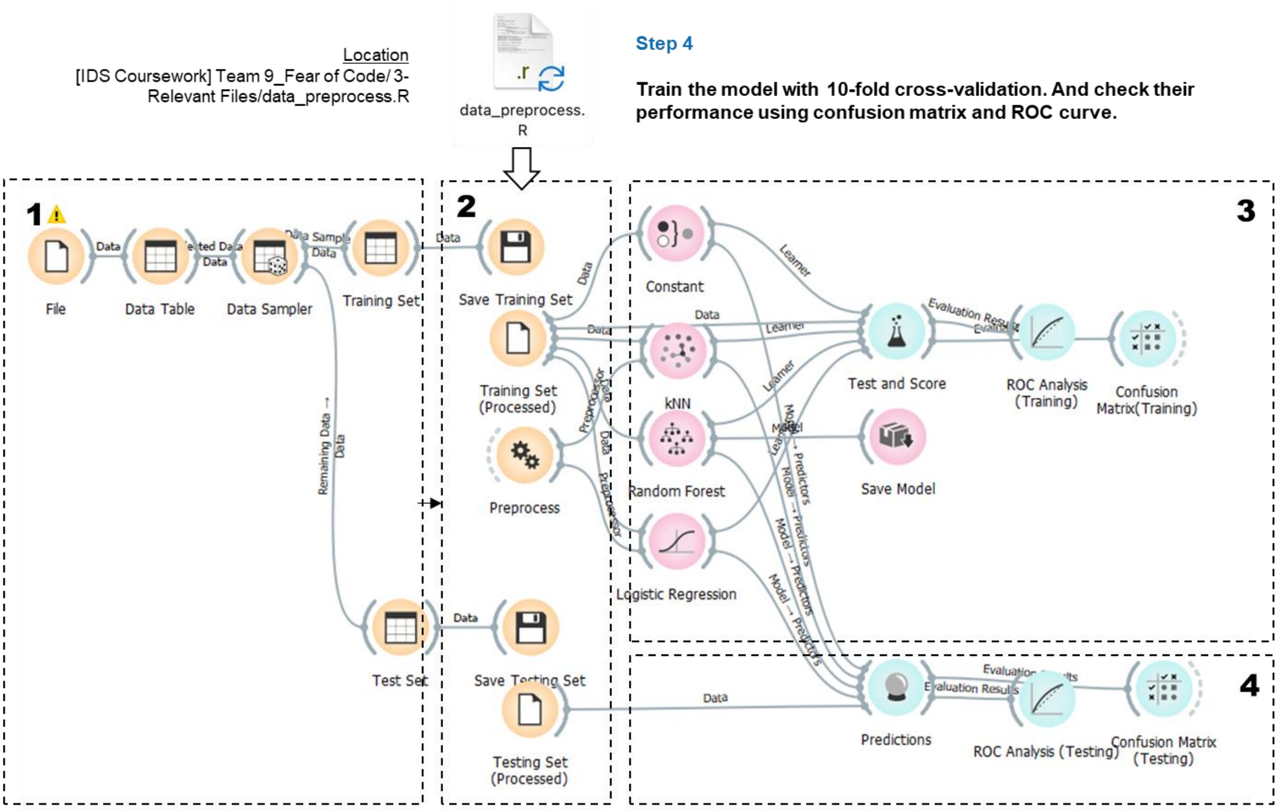
Training set with 10 fold cross-validation

Model	**AUC**	**CA**	**F1**	**Precision**	**Recall**
Random Forest	0.701	0.830	0.785	0.792	0.830
***k*-NN**	0.680	0.827	0.758	0.779	0.827
Logistic Regression	0.677	0.829	0.767	0.786	0.829
Constant	0.500	0.827	0.748	0.683	0.827

Testing set

Model	**AUC**	**CA**	**F1**	**Precision**	**Recall**
Random Forest	0.716	0.835	0.789	0.803	0.835
***k*-NN**	0.682	0.831	0.771	0.796	0.831
Logistic Regression	0.699	0.828	0.762	0.785	0.828
Constant	0.500	0.827	0.748	0.683	0.827

Appendix IV: Model Building and Evaluation Procedure



Location
[IDS Coursework] Team 9_Fear of Code/ 3-Relevant Files/data_preprocess.R

Step 4

Train the model with 10-fold cross-validation. And check their performance using confusion matrix and ROC curve.

Step 1

Load and split the data into training and testing set

Step 1

Preprocess data with R script

Step 3

Test the performance on test set