Finding Most Informative Variables in Predicting Flight Delay

Bilge Elitok
M.Sc. Student in Computational Biomedicine
belitok@student.uef.fi

December 25, 2024

Abstract

Temporal and financial costs being brought to passengers and airlines by the flight delays, positions on-time efficiency as a rewarding goal. A departure delay happens when a plane's departure time exceeds the scheduled time by the Central Reservarion System (CRS). With this project, one important step towards more efficient delay prediction was performed: selection of most important predictors of a delay. For training the model to predict whether a flight will be delayed, the Airline On-Time Performance Data was used. Target variable, which is initially a continuous value of flight delays in minutes, was binarized so that delays exceeding 15 minutes would be represented by ones. For predictive model class, Random Forest Classifier was used due to its inherent feature importances attribute and robustness while handling of both categorical and numerical variables. Best performing RFC appeared to have no errors, therefore validating the model on an external set was necessary. Prior to feature selection, data preprocessing, numerical and categorical encoding, small scale feature engineering was performed. One of the engineered features, 'InHolidaySeason', appeared to be in the 6.25th percentile among the features that are returned by the tuned model. For the experiment, an external dataset and the highest ranking feature, arrival delay or 'ArrDelay', was used as the sole input for an untuned Random Forest Classifier. The predictions based solely on this feature measured to have a precision of 0.94, recall of 0.97, and F1-score of 0.95 for predicting an on-time flight. For delayed flights, the performance appeared to be weaker with a precision of 0.82, recall of 0.69, and F1-score of 0.75. This results revealed that, Arrival Delay is indeed a significant feature to predict flight delay, and there is room for improvement in reducing false negative detection.

Nature of Variables

First, features required to be classified as random or deterministic. Based on the understanding of Maxim Raginsky and B. H. Juang 's lecture notes, the following guideline was created.

- IF feature is predictable based on a schedule (features set by the central airline reservation (CRS) system) or a rule (e.g. binning of Distances into the DistanceGroup feature) these values are considered as **deterministic**.
- IF a feature depend on unpredictible, external conditions (e.g. delay Related features: DepDelay, WeatherDelay) or they reflect variations of the outcomes (such as AirTime, ActualElapsedTime), they are considered random.

Likewise, a guideline for estimating the distributions of the features were constructed:

- If counting events on fixed intervals $\longrightarrow$ Poisson
- Event has binary outcome $\longrightarrow$Bernoulli/Binomial
- If variable has nominal values with mutliple categories $\longrightarrow$Categorical Distribution
- If variable follows time to event format or if it is skewed $\longrightarrow$ exponential distribution
- If variable represents continuous data over time, balanced and unskewed $\longrightarrow$ Normal distribution.
- If data symmetric and is bounded between $\longrightarrow$ beta
- If data is essentially positive continuous values and it is right skewed $\longrightarrow$ gamma.

Features and their nature were represented in the table 4, at the end of this document.
The observed variables and target variables as well as their nature and observed distributions are shown in Table 7.
For the latent variables, three candidates were considered and attempted to be integrated into the analysis:

1. Weather Forecast between 01.11.2006 and 31.12.2006 was considered as the latent variable contributing to the underlying structure give rise to delays.*(fig.6 for details)*
2. Holiday Seasons for United States (Easter, Christmas, New Year's Eve) were incorporated by 'InHoliday-Season' column. This column is constructed by including the days before every weekend and two days prior to every major U.S. holiday in that period.*(fig.5 for details)*
3. Repetitive use of an aircraft, was hypothesized to have a latent effect however, it was observed to be misleading.*(fig.8 for details)*

While the weather patterns do not appeared to be in line with the observed data, the effect of holiday seasons were observed on the 13th highest ranked among 193 features in final feature importances. This observation was interpreted as, further modelling strategies, for example hierarchical Bayesian models might be required for the proper modelling of the latent variable space (Alleby, 2005).

Methods Used

Features are closely examined by the descriptions provided by Bureau of Transportation Statistics (BTS). The task of finding the most informative features for predicting flight delay, was implemented in multiple steps:

1. The quality of 57 features was assessed based on their contribution to the prediction task.
2. Specific columns were discretized or encoded to be included in the final dataframe used for prediction.
3. External data on the holiday season in 2006 was analyzed to determine its potential impact on flight delays.
4. General trends in the data, as well as deviations from these trends, were examined for their potential contribution to departure delays.
5. Based on the insights from the previous analyses, and personal understanding, reduntant features were dropped, combined or, transformed.
6. The truly informative features were extracted and copied into a new dataframe.
7. The resulting dataframe, obtained from the feature selection process, was used to train a predictive model for target prediction.

Alternative Methods:

The Bayesian Linear Regression Workflow of this section is based on the PyMC3 Core Notebook and blog entry by Wiecki, 2013

Bayesian Network Classifiers (Friedman et al., 1997) or their simplest mamber NaiveBayes Classifier could be implemented for the flight classification task (Minsky, 1961). Even though it inherently do not return feature importances, it can be set for this purpose using the method described in stack.overflow: https://stackoverflow.com/a/50530697.
Main challenge for NaiveBayes is its zero colinearity among features assumption.
The problem can be constructed as a Bayesian Network, and the R package **bnlearn** could be used for performing inference(Scutari, 2010).
Lastly, probability of a delay could be modeled based on the features X, through Bayesian Logistic Regression. The likelihood function in this case would be derived from the Bernoulli Distribution:

$$p(y_i|X_i,\beta,\beta_0) = \text{Bernoulli}(p_i) \tag{1}$$

Posterior distribution of the coefficients β and Intercept β_0 given the data could be estimated through sampling. Markov Chain Monte Carlo methods (e.g. NUTS (No-U-Turn-Sampler) would be one option to directly sample from their posterior distributions (Blei, 2006).

For Bayesian Logistic Regression workflow, `pgm3.Model()` can be initialized with the two priors:

$$\beta \sim \mathrm{N}(\mu, \sigma), \quad \beta_0 \sim \mathrm{N}(\mu, \sigma)$$

After priors are specified, probability of the delay p_i could be modeles using logistic function, which transforms the linear combination of the features into values between the range of 0 and 1, allowing binary classification:

$$p_i = \sigma(X_i \cdot \beta + \beta_0)$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Posterior distribution of parameters will be derived as:

$$p(\beta, \beta_0 | \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} | \mathbf{X}, \beta, \beta_0) \cdot p(\beta) \cdot p(\beta_0)$$

Running NUTS sampler to approximate the posterior distribution can be stored in a `trace` object and posterior distributions can be visualized using PyMC3's own plotting function `plot_posterior()`
Lastly, posterior predictive modeling can be performed on the external kaggle dataset used on this project with using PyMC3's `sample_posterior_predictive` function.
To summarize, Bayesian Logistic Regression would provide a probabilistic framework by allowing the priors to be incorporated and leveraging the posterior inference.

Requirements and their Function

Packages and libraries central to data preprocessing, model training, hyperparameter tuning and evaluation as well as importing the RandomForestClassifier model class was used.

- numpy for numerical operations, array handling
- pandas: dataframe handling
- seaborn, matplotlib are used for plot based visualization
- sklearn.preprocessing was central for scaling, variable encoding and feature transformations
- sklearn.metrics for performance evaluation of the models based on accuracy, precision, recall and F1 scores.
- GridSearchCV from sklearn.model_selection for hyperparameter tuning
- RandomForestClassifier from sklearn.ensemble is the main classifier used for feature importances.

Summary and Key Properties of All_PIT_2006 dataset.

Dataset contains 94944 samples, the flight information to and from Pittsburg International Airport (PIT) between the dates 01/11/2006 and 12/29/2006. Each row represents one flight. When a flight is cancelled, this is rendered as an empty cell in the DepDel Column. Cancellations have 4 distinct codes: A-Carrier Caused, B-Weather, C-National Aviation System, D-Security. Some features are interacting such as arrival delay (ArrDelay = ArrTime - CRSArrTime) and for cancelled, only system related delay columns ArrDelSys15 and ArrdelSys30 are filled.

Exploratory Analysis of Factors Contributing to the Delay

Data was analyzed for exploring the connection between the metadata and the flight delay.
A custom function `load_data()` was used for loading the dataset from a local .csv file, globalizing dataframe object and returning the summary of the dataframe.

File Name	Rows	Columns	Has Null	Total Null	30% missing
all_PIT_2006	94,944	57	True	107,138	[CancellationCode]

Table 1: Dataset Summary

Table 2: Delay Causes and their Contributions

Delay Code	Delay Cause	Total Delay (minutes)	Percentage of Total Delay (%)
I	Late Aircraft	459238	38.17
G	National Aviation System	403444	33.53
E	Carrier Caused	278855	23.18
F	Weather	60219	5.00
H	Security	1500	0.12

For any delay, five different delay sources was given: due to Carrier, to Weather, to Security Reasons, NAS (National Aviation System) Delays, or due to a late arriving aircraft delay. Some delays appeared to be a combination of multiple reasons and minute-wise percentage delays caused by each of the reasons are given by the following table:
Then, `DepDel` variable was binarized in a new `DepDelayBinary` column. If the value exceeds 15 minutes, it was respresented with 1 = Delay. For the values lower than 15 minutes, it was considered as No Delay and encoded with 0. This discretization was essential for following exploratory analysis.

The Effect of the Airline to the Departure Delay:

A common experience of travellers is that the specific airlines to be known for the delays in the arrival or departure. Therefore, the number of delays exceeding 15 minutes per airline has been a prioritized feature to be explored. In case of using a tree-based model for regression, numerical encodings would be needed. Therefore, a custom function `analyze_and_encode_airline_delays()` to plot departure delays per airline and return the numerical encodings, keys and average delay besides the delay count was implemented. To identifty unique carriers, [AirlineID] column including the International Air Transport Association (IATA) codes for spesific airlines were used. This codes commonly used in flight tickets (e.g. DL for Delta Airlines, YV for Mesa Air Group), schedules, and databases.

Table 3: Airline Delays Summary (Sorted): Average Delay and Number of Delayed Flights per Airline

IATA Code	**AverageDelay**	**DelayedFlightsCount**	**UniqueCarrier_Encoded**
US	5.5765	4525	10
NW	6.8059	550	6
FL	7.1582	660	4
CO	8.0172	211	1
OH	8.6839	1088	7
DL	9.7369	832	2
WN	10.5100	2771	11
OO	11.5434	333	8
XE	17.3450	1972	12
B6	14.2571	496	0
MQ	14.1150	2412	5
UA	15.9204	1141	9
EV	18.6985	416	3
YV	18.4221	917	13

For certain airlines, top 5 being US, NW, , the threshold value of 15 minutes were already exceeded by the average. These results led this feature to be considered as a prioritized feature for final step of the project: delay prediction.

The Effect of the State and Cities to Departure Delay

Dataset included 12 relevant features that describe origin and destination points for the flight:

1. `Origin, Destination` for Origin and Destination airports
2. `OriginCityName` and `DestCityName`
3. `OriginState` and `DestState`

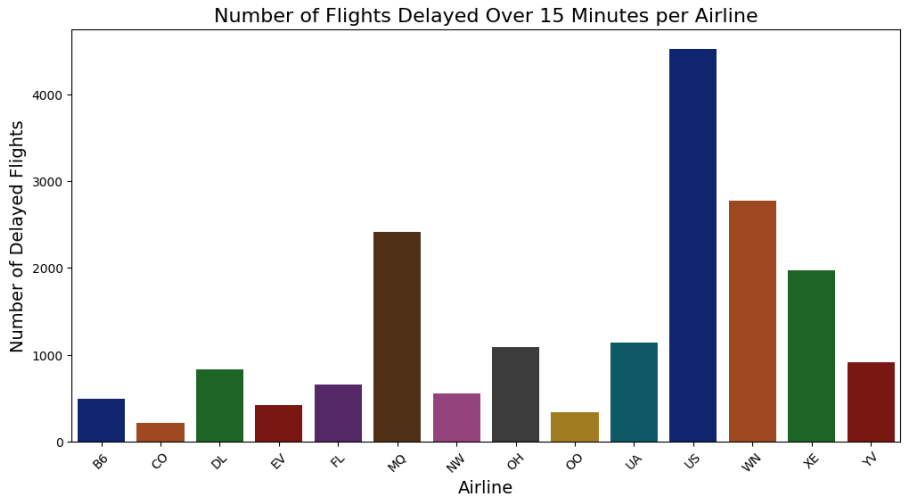

Figure 1: Airline Delays Plot: Average Delay and Number of Delayed Flights per Airline. Labels on the X axis represents identifiers assigned to airlines by the International Air Transport Association (IATA), i.e. appeared as the unique string values in AirlineID column. Y axis represents the count of instances that every time a delay occured. Data appeared to have categorical distirbution.

4. `OriginWac` and `DestWac`, which is numerical encoding of the variables
5. `OriginStateFips` and `DestStateFips` are the FIPS (U.S. Federal Information Processing Standard) codes for `OriginState` and `DestState`.

Due to duplicate representation of those features, the above mentioned 12 column was examined by plotting 6 of them. The distribution of `DepDelayBinary` was plotted for `Origin, Destination`, `OriginState` and `DestState`, OriginState and `DestState`.
For those 12 relevant features, half of them was required to capture the information they represent. This was also taken into account. Results are given by the figure 2.

Inspecting Temporal Variables

Next, temporal variables 'Year', 'Flights', 'DayOfWeek', 'DayofMonth', 'Month', 'Quarter', 'FlightDate' was inspected. Year and Flights included only one value for each, 2006 and 1, therefore they are not considered relevant. Furthermore the setting up of the Month column was considered misleading as it ranged from 1 to 4 even though the dataset only included 2 months: November and December.

To gain insight to the effect of the retained columns, their relationship with `DepDelayBinary` target was visualized with countplots.
Upon inspection, it was considered necessary to ensure including those four variables out of six to the cleaned dataframe. Additionally, FlightDate was transformed into Unix Timestamps for better compatibility with the upcoming model.

Numerical Variables Continued:

Next, the remaining numerical variables were examined for an informative visualization method selection. For some variables there are a predefined CRS estimate of its value (CRSArrTime, CRSDepTime,CRSElapsedTime) and an actual value. This variables plotted side by side. Relevant variable: AirTime and Distance, TaxiIn-TaxiOut (minutes), WheelsOn-WheelsOff (Hours and Minutes) were also plotted in pairs, results are given by the figure 4.

CRSElapsed time and ActualElapsedTime did not show significant difference for Delayed and On-time Flights, and same observation was made for pairs TaxiIn-TaxiOut, Airtime-Distance. Even though it was expected to find operational patterns related to delay through inspection of Taxi-In, Taxi-Out durations, a slight difference for delayed flights was observed.

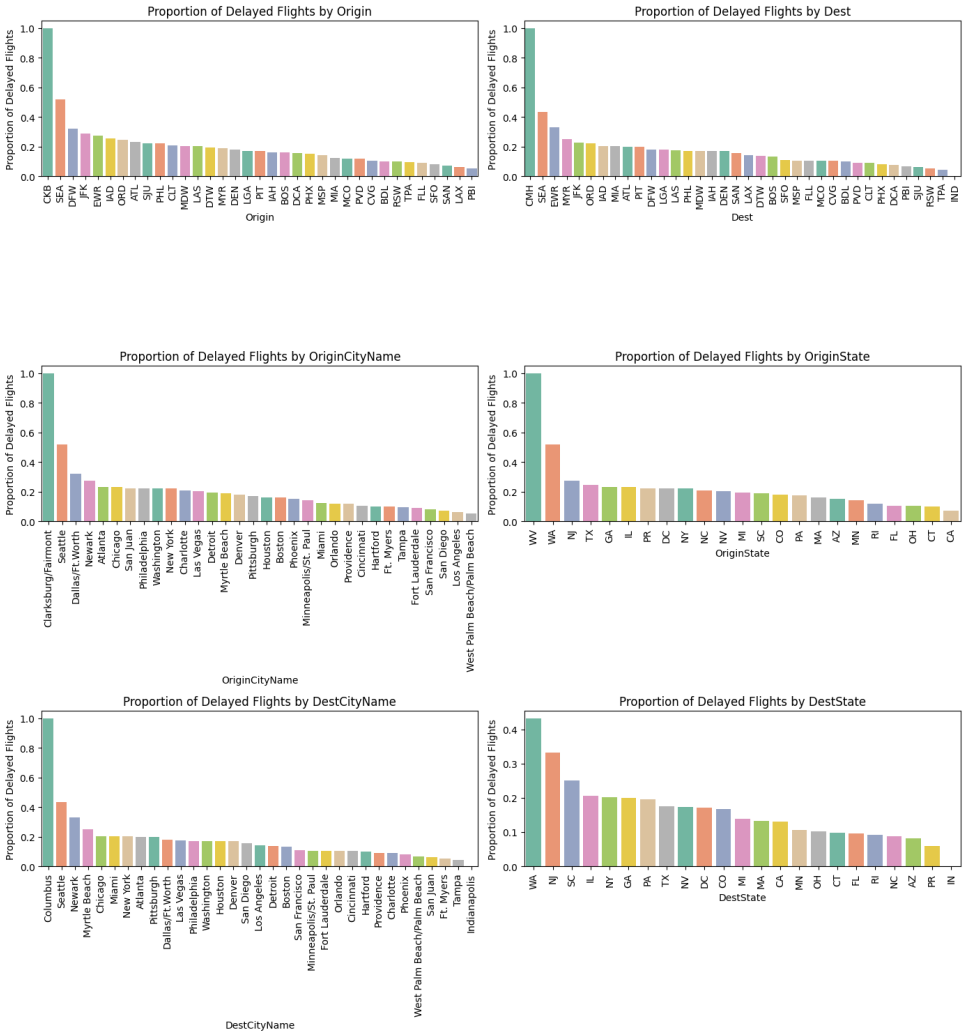

Figure 2: Number of flight delays by `Origin`, `Destination`, `OriginState` and `DestState`, OriginState and `DestState` normalized for the first ranking feature on each.

However for pairs concerning Departure and Arrival times, a pattern was found: the closer the arrival or departure times get to the hour 23:00, more flight get delayed. This effect represented by inverse violin plots, and it was also observed for WheelsOff-WheelsOn pair. Since WheelsOff-WheelsOn values was under the same representation of "hh:mm" (Hours-Minutes), it was considered as an extension of the information already given by ArrTimeBlk and DepTimeBlk, two important features that will be discretisized during downstream steps.

Departure Delay Distribution across Numerical Features:

For the next step of exploratory analysis, features were separated into classes based on the data types involved. The nature of the variable and the distribution summary were stored in the table 4 for further analysis.
As it was represented by the table 4, features come with two inherent challenges: They are high in numbers and they are at the risk of multicolinearity. Numerical features such as `ActualElapsedTime, AirTime, CRSElapsedTime` are closely related since they are relevant measures of the travel of the plane. Similarly, `Distance` and `DistanceGroup` might be significantly correlated as latter is likely to be the categorization of the former.

Feature Engineering to Improve Inputs for the Predictive Models

So far, the dataset appeared to be more and more open to improvement by feature engineering, i.e. transformation of raw values into more meaningful ones by various methods such as numerical representation, creation of interaction features, aggregations and categorical transformation. For this purpose, categorical features (UniqueCarrier, Carrier, TailNum, Origin, Dest, OriginCityName, DestCityName, CancellationCode, DayofMonth, DayOfWeek) were numerically encoded. Three of the temporal Features; Year, Quarter and Month were represented by the Unix timestamp. Cancellation related features were dropped as they temporally come after a delay has occured.

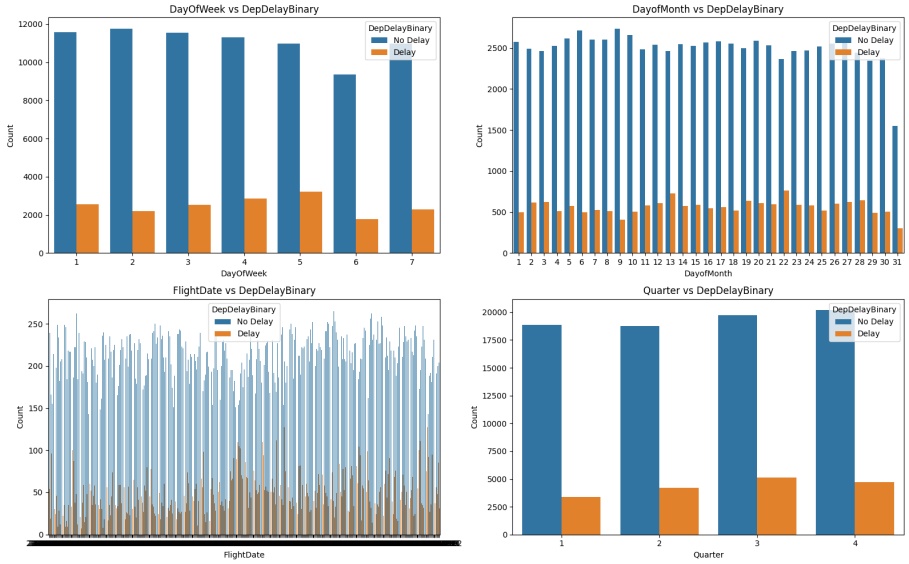

Figure 3: Countplots of temporal variables DayOfWeek, DayofMonth, FlightDate and Quarter. Distributions are observed to be uniform.

Delay Related features were combined into the 'TotalDelay' feature, as it was not expected from the model to distinguishing between delay sources.

Weather Forecast, Repetitive use of an Aircraft and Holiday Season

Next, the effects of the weather temperature trends, whether an aircraft is in frequent use, and the nation-wide holidays were analysed.
In order to extract more information from the date variable, the holidays in U.S. between 01.11.2006 and 31.12.2006 were taken into account. By plotting the delays on weekly bases the patterns and peaks in flight delays, and their potential association with holiday travel or other factors was aimed to be highlighted. Top ten weeks with total delays were returned as an array.
The graph shows two major peaks, in the weeks 26 of the total weeks and 26 It was initially expected that the flight delays would match the week prior to major holidays in 2006. However, the analysis of top 10 weeks with highest delay revealed that, most significant peaks fell outside the typical holiday periods as they occured in week 29 with 44.906 total delays, and week 26 with 42.780 of total delays. Therefore, external resources investigated for an unusual weather event that might explain the peak delays on those dates.
Likewise the holiday seasons, unusual weather events did not appear to have significant contribution to the peaks.

Encoding the Timeblock Columns into Categories

As features are in abundance, and for some cases, information were in repetition, next steps aimed to reduce the number of features. The abundance of features introduced a challenge for delay prediction task as the redundant features distilled the features that are truly informative for prediction.
For this purpose, first step was encoding the timeblock columns (`ArrTimeBlk, DepTimeBlk`) into four quarters of the day. Then, the total delays for corresponding quarter of the day was plotted.
Figure 3 visualized that the most delays for departing planes happens in the third quarter of the day, for the planes arriving between the 12:00 - 17:59, followed by the fourth quarter of the day. Initially, delay in departures expected to have a similar trend as the delays in arrivals, that is, the number of delays increasing for the later times of the day. Furthermore, for planes only transit passing Pittsburg Airport, the effect of a late arriving plane is expected to compound the plane's late departure. As different quarters of the day showed significantly different numbers of delays, the numerical encodings corresponding features `ArrTimeBlk, DepTimeBlk` were retained in the final dataframe that will be used for prediction.

Tail Numbers and Previous Usage Count

If an aircraft is in repetitive use, the operational processes, wear and tear or scheduling issues between its use can contribute to the delay. Repetitive usage of the same aircraft might indicate higher likelihood of that aircraft to experience delays. Spesifically, by examining the usage count with the help of tail numbers (`TailNum`), patterns that help predicting the delays might be inferred.

In order to give numerical expression to this idea, a new column derived by how many times a specific tail number was appeared: `PreviousUsageCount`.Upon inspecting the column and printing the top ten most used aircrafts, a peculiar tailnumber was recognized: 'TailNum = 0'. This was unusual in comparison to the rest of the dataset since tailnumber of any other aircraft follows the pattern of Letter-Number-Number-Number-Letter-Letter, such as N110UW or N851MJ. Another peculiarity is, the Tailnumber 0 has almost 6x more counts than the second most used tail number. Therefore tailnumber , might represent missing entries, and as it is shown in the figure below, it might have an artifically populated PreviousUsageCount value. This extreme value might make is prone to introduce biases for the prediction.
Upon this observation, TailNumber itself or any variaton of it was avoided to be included among predictors. All other aircrafts were shown continuous usage over time as expected, with their values increasing with the timestamps.

Model Class Selection and Final Dataframe to be the Input

This section is based on related Scikit-Learn documentation and Chapter 3 of Géron's 2017 book.

The main task of this project is to determine the features that contributes most to the flight delay prediction. Therefore model class was selected as Random Forest Classifier (RFC), as the feature importance scores are available within the model itself. Furthermore, each decision tree of RFC works on different subset of the large dataset, the risk of overfitting was aimed to be avoided.

RFC requires minimal preprocessing, in our case, simple LabelEncoder() implementations were sufficient for the preliminary model which has above 97% accuracy. Additionally they do not rely on scaling or normalization of of feature values.

Even though RFC can handle both continuous and categorical values of features, highly cardinal categorical data such as: AirlineIDs, Origin or Destination can lead to inefficiency. Therefore, one-hot encoding and label encoding were retained processing steps towards final dataframe.
First, final features are separated into `Categorical_Features` and `Numerical_Features`. The target variable, `DepDelayBinary`, is also separated and stored in a target object to build the model. For numerical features, `SimpleImputer` with the median strategy replaced missing columns with median values, while for categorical features, missing values were replaced with the most frequent ones. Categorical columns were one-hot encoded, and the `drop_first=True` argument was passed to ensure that the model remains interpretable. The final feature set was stored in `X_encoded` and passed into the next step.
For the next step, a `pipeline` object was created to perform feature scaling (using sklearn's `StandardScaler()`) and model building, using `RandomForestClassifier()` as the main predictive model.
To optimize the model and extract feature importances from the tuned estimator, a grid search was conducted to find the parameters for the RFC. Due to the computational cost of RFC, a smaller grid was passed, as shown below:

- `n_estimators`: The number of trees in the forest: 100 or 200.
- `max_depth`: The maximum depth of each tree: 10, 20, or None.
- `min_samples_split`: The minimum number of samples required to split a node, evaluated with values of 2 and 5.
- `bootstrap`: A boolean indicating whether bootstrap sampling should be used for the trees.

Next, five fold cross validation was used for model evaluation for above parameter combinations. The gridsearch appeared to be costly and finished within 15 to 20 minutes, even if the use of all available CPU was allowed (`n_jobs = -1`).
The best-performing model that used the highest-performing parameters was stored in the `best_model_final` object. It was then utilized to provide a measure of feature importances through the `feature_importances_`

Table 4: Features in the Final Input Dataframe: The following features were retained as predictors for Random Forest Classifier(s) to be trained, tuned and tested.

Feature Name	**Data Type**
AirlineID	float64
DayofMonth	int64
DayOfWeek	int64
ActualElapsedTime	float64
CRSElapsedTime	int64
AirTime	float64
ArrDelay	float64
ArrTimeBlk	object
CRSArrTime	int64
DepTime	float64
DepTimeBlk	object
CRSDepTime	int64
Origin	object
OriginCityName	object
OriginState	object
Dest	object
DestCityName	object
DestState	object
DistanceGroup	int64
TaxiIn	int64
TaxiOut	int64
InHolidayPeriod	bool
Timestamp	float64
ArrTimeBlk_Encoded	object
DepTimeBlk_Encoded	object
DepDelayBinary	int32

attribute. The returned `feature_importances_` were stored in a `pandas.DataFrame`, as this object is compatible with downstream visualization steps.
To conclude, the generalization capability of `best_model_final` was assessed using hold-out validation. A confusion matrix, classification report as well as key evaluation metrics such as accuracy, precision, recall, and F1-score, were calculated to provide a comprehensive understanding of the model's performance.

Results

Most Important Features to Predict Flight Delays:

The tuned RFC made predictions based on the following feature importances:
Distributions of the most important features were needed in order to build a mixture model that perform the generative process for the experiment. Therefore, the frequency most important 9 features were plotted and KDE curves were overlaid *(figure 10)*.
Confusion matrix for the best performing model essentially resulted in a suspiciously perfect model. However, as model will be tested on experimental data, the best model was retained.

Experiment: Using 'ArrDelay' to Predict Departure Delay on an External Dataset

As the tuned classifier's predictions evaluated to be 100% accurate, an external test was considered necessary. **Dataset Carrier On-Time Performance Dataset** on Kaggle, provided by user *mexwell* was used. The aim for this experiment was to measure how well the delays will be predicted by relying solely on the highest ranking feature. The most informative feature was given as the only input for a untuned random forest classifier. The training process took less then 3 minutes and the classifier achieved an overall accuracy of 92% predictiong the departure delays. The classification report were given in the table 6.

Index	Feature	Importance
3	ArrDelay	0.597468
1	ActualElapsedTime	0.071761
7	TaxiOut	0.059047
2	AirTime	0.048523
8	Timestamp	0.046846
4	CRSArrTime	0.036189
6	TaxiIn	0.032435
5	CRSDepTime	0.029013
0	AirlineID	0.009485
185	ArrTimeBlk_Encoded_1	0.008291
187	ArrTimeBlk_Encoded_3	0.007699
186	ArrTimeBlk_Encoded_2	0.002980
188	InHolidayPeriod_True	0.002565
175	DestState_NJ	0.001531
52	Dest_EWR	0.001211

Table 5: Feature Importance Table

Class	Precision	Recall	F1-score	Support
0	0.94	0.97	0.95	499,917
1	0.82	0.69	0.75	100,083
Accuracy			0.92	600,000
Macro avg	0.88	0.83	0.85	600,000
Weighted avg	0.92	0.92	0.92	600,000

Table 6: Classification Report: the classification report reveals that the model performs well in predicting non-delayed flight while having weaker performance on delayed flights.

Overall results of the project reveal that, arrival delay or 'ArrDelay' can be considered as the most important feature predicting the flight delays, and was assigned 59% importance by the best performing RandomForestClassifier model.
For future directions the compounding effect of arrival delay, 'the flight chain' can be further inspected.The **flight chain** is a term used by Liu and Wu for their remixed Bayesian network-based algorithm for estimating flight delays *figure 12*. Following figure from their study illustrated the compounding effect of flight delays, further increasing the feature ArrDel's importance.

Following pages contains the figures and tables mentioned in the text.

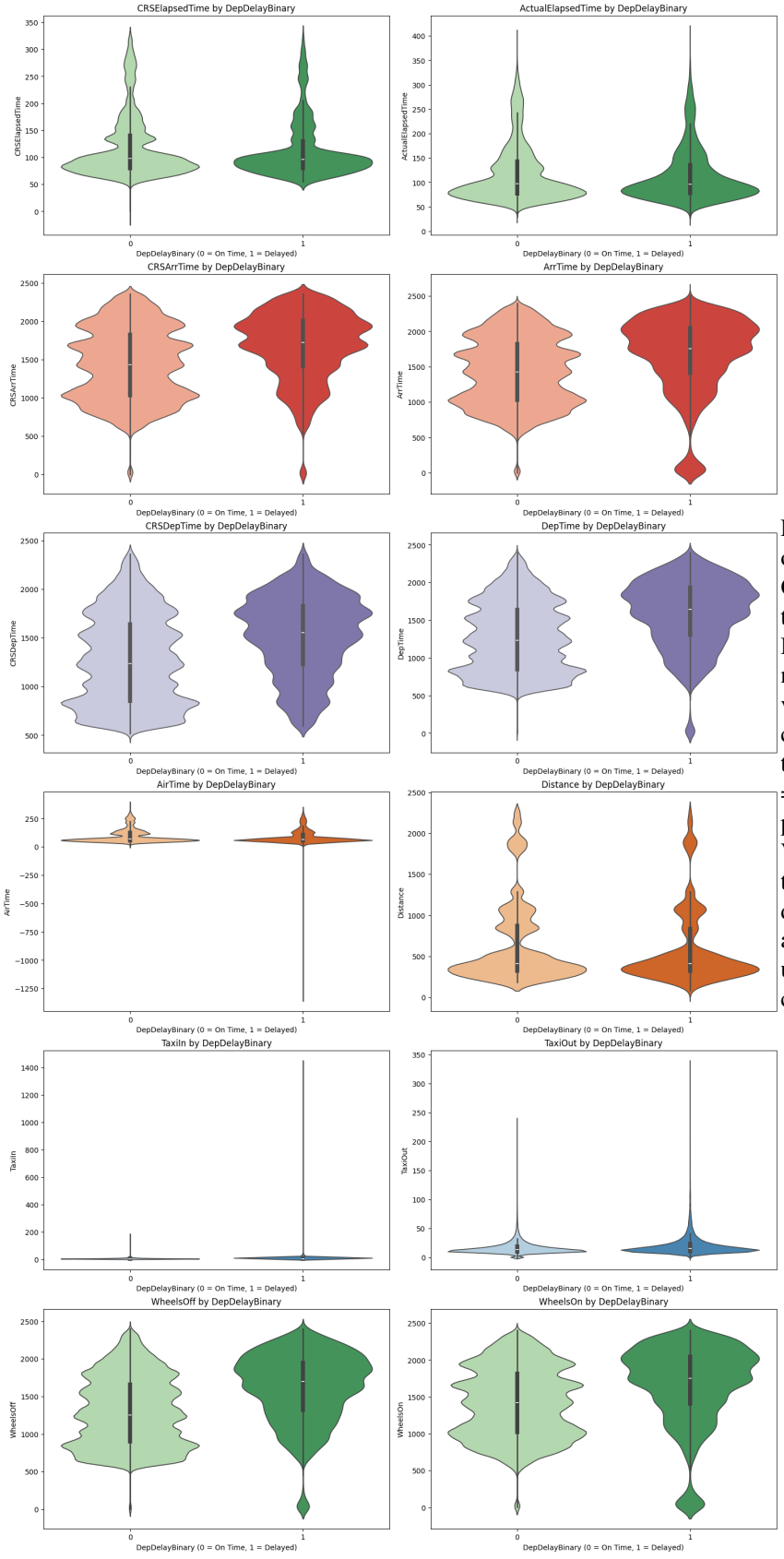

Figure 4: Distribution of System, Flight and Operational Characteristics by Departure Delay Status. Each numerical feature is visualized across two categories—flights on time (DepDelayBinary = 0) and delayed (DepDelayBinary = 1). Violin plots illustrate the data density and distribution while alternating colors are used for improved comparison.

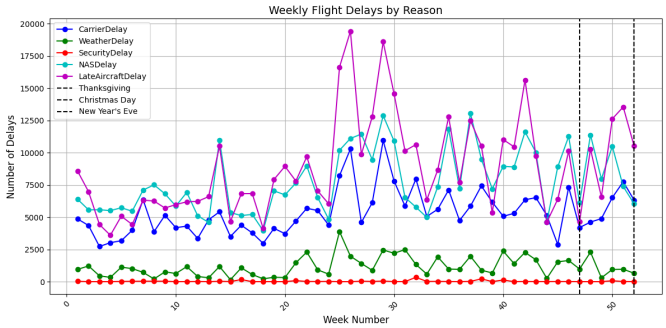

Figure 5: Flight Delays in given fixed timeline in 2006: The number of flight delays attributed to different reasons for each week of the year. Dashed vertical lines indicate major holidays: Thanksgiving (Week 47), Christmas Day (Week 52), and New Year's Eve (Week 52). figure and data from Weather Spark

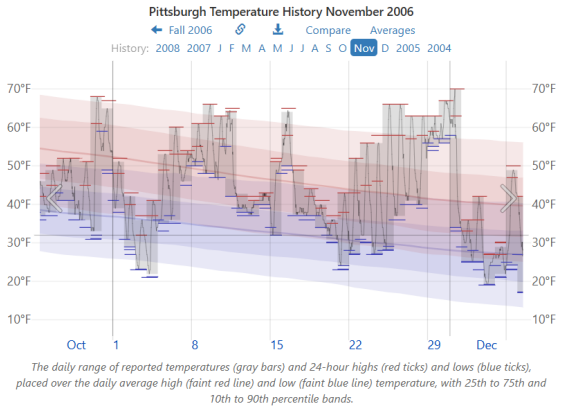

Figure 6: Temperature Graph Showing 2006 Weather History in Pittsburgh,the final weeks of November is brought into focus for unusual temperature falls that might affect routine aviation schedules.

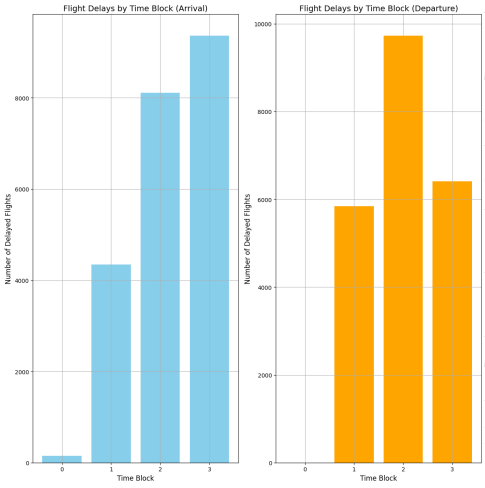

Figure 7: Number of flight delays by time block for arrival (left) and departure (right). The time blocks are encoded as '0' for the time between 01:00-05:59, '1' for time between 06:00 - 11:59, '2' for the time between 12:00 - 17:59, and '3' for the time between 18:00 - 23:59. The arrival and departure plots show the delay entry counts for flights arriving within each quarter.

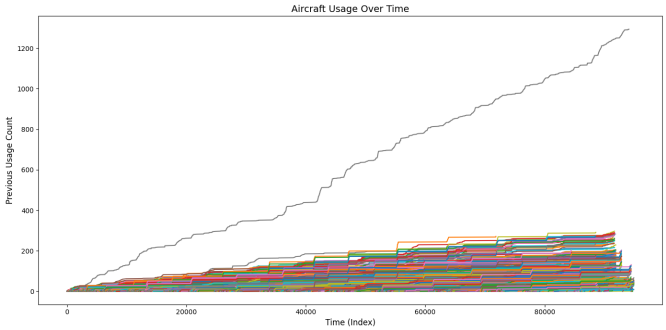

Figure 8: Plot for visualizing the aircraft usage over time. Colors represent different tail numbers therefore aircrafts, and how their usage increased over time. The outlier line represents 'tailnum = 0', which might be a general term used for the missing tailnumber entries. Data appeared to have log-normal distribution with a low shape parameter.

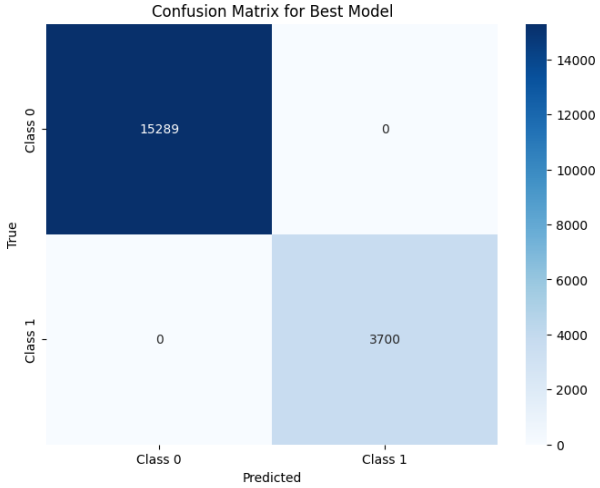

Figure 9: Confusion Matrix of Best Performing RandomForestClassifier

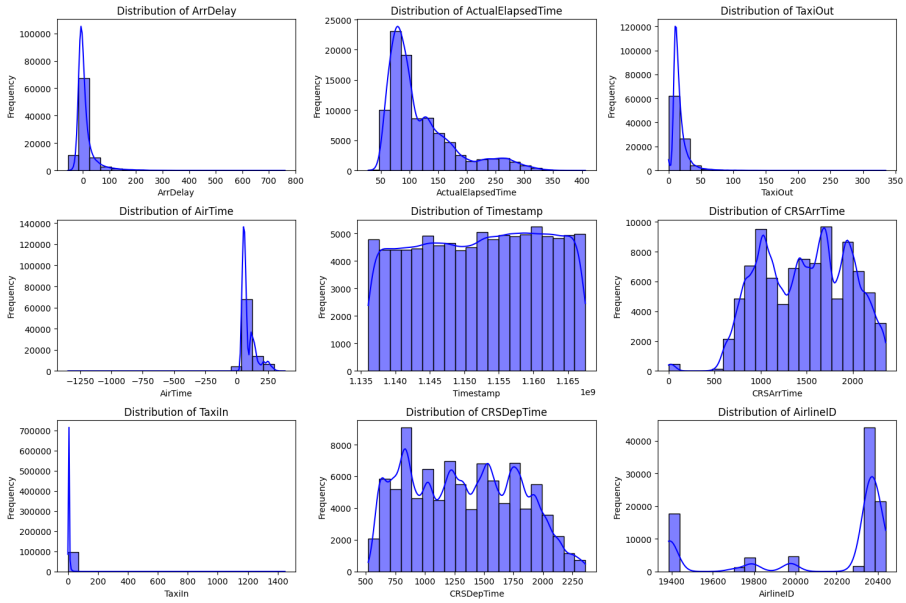

Figure 10: Nine of the highest ranking features that guides the predictions of best performing Random Forest Classifier model. Distributions: ArrDelay:Normal-Gamma, ActualElapsedTime: Normal-Gamma, TaxiOut: Normal-Gamma, Airtime:Normal, Timestamp: Uniform, CRSArrTime: Normal, TaxiIn: Gamma, CRSDepTime: Uniform, AirlineID: Categorical

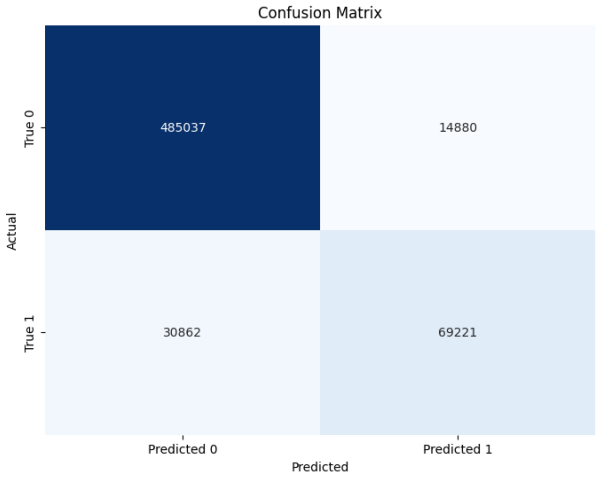

Figure 11: Confusion Matrix of the predictions of the model that uses arrival delay information as the only feature.

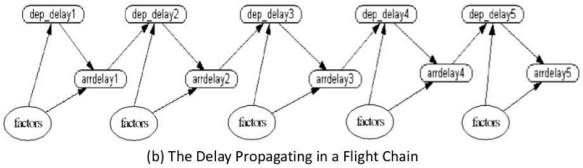

Figure 12: Flight Chain and the Compounding Effect of a Delay (Liu & Wu, 2024)

Feature	Nature
Year	Deterministic
Quarter	Deterministic
Month	Deterministic
AirlineID	Deterministic
UniqueCarrier	Deterministic
Carrier	Deterministic
FlightDate	Deterministic
DayofMonth	Deterministic
DayOfWeek	Deterministic
Flights	Deterministic
FlightNum	Deterministic
TailNum	Deterministic
ActualElapsedTime	Random
CRSElapsedTime	Deterministic
AirTime	Random
ArrDel15	Random
ArrDel30	Random
ArrDelSys15	Random
ArrDelSys30	Random
ArrDelay	Random
ArrTime	Random
ArrTimeBlk	Deterministic
CRSArrTime	Deterministic
DepDel15	Random
DepDel30	Random
DepDelSys15	Random
DepDelSys30	Random
DepDelay	Random
DepTime	Random
DepTimeBlk	Deterministic
CRSDepTime	Deterministic
Origin	Deterministic
OriginCityName	Deterministic
OriginState	Deterministic
OriginStateFips	Deterministic
OriginStateName	Deterministic
OriginWac	Deterministic
Dest	Deterministic
DestCityName	Deterministic
DestState	Deterministic
DestStateFips	Deterministic
DestStateName	Deterministic
DestWac	Deterministic
Distance	Deterministic
DistanceGroup	Deterministic
TaxiIn	Random
TaxiOut	Random
WheelsOff	Random
WheelsOn	Random
Cancelled	Random
CancellationCode	Deterministic
Diverted	Random
CarrierDelay	Random
WeatherDelay	Random
NASDelay	Random
SecurityDelay	Random
LateAircraftDelay	Random
InHolidayPeriod	Deterministic

Table 7: Feature Nature

References

1. T. Wiecki. The Inference Button: Bayesian GLMs Made Easy by PyMC, [Blog Entry] 2013. Retrived from: https://twiecki.io/blog/2013/08/12/bayesian-glms-1/

2. M. Scutari. Learning Bayesian networks with the bnlearn R package. Journal of Statistical Software, 2010.

3. Minsky, M. L. (1961). Steps toward artificial intelligence. Proceedings of the IRE, 49(1), 8–30. ://doi.org/10.1109/JRPROC.1961.287775

4. Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. Machine Learning, 29(2–3), 131–163. https://doi.org/10.1023/A:1007465528199

5. B. H. Juang , ECE 3075A: Random Signals, Lecture 22 - Random Processes. School of Electrical and Computer Engineering, Georgia Institute of Technology, Fall 2003. Accessed [24.12.24].

6. Raginsky M. (2016). Intro to stochastic systems (Spring 16): Lecture 7 - Randomness and determinism [Lecture notes]. University of Illinois.

7. Bureau of Transportation Statistics. U.S. Department of Transportation. Retrieved December 24, 2024, from https://www.bts.gov/

8. Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

9. Weather in Pittsburg on November 2006. Data was retrived from Weather Spark https://weatherspark.com/h/y/19773/2006/

10. Allenby, G. M., Rossi, P. E., & McCulloch, R. E. (2005). Hierarchical Bayes models: A practitioner's guide. Fisher College of Business, Ohio State University & Graduate School of Business, University of Chicago.

11. mexwell. (2024). Carrier On-Time Performance Dataset. Kaggle. Retrieved from https://www.kaggle.com/mexwell/carrier-on-time-performance-dataset

12. Liu, Y., & Wu, H. (2024). A remixed Bayesian network-based algorithm for flight delay estimating. College of Computer Science and Technology, Civil Aviation University of China; Department of Mathematics and Statistics, Minnesota State University.