Flight Delay Predictor

Kush Aggarwal 2020516
Anoushka Kumar 2021133
Mehak Gopal 2021475
Nalin Arora 2021478

Abstract

Aeroplanes are indispensable for modern society, enabling swift global travel and connectivity. People rely on aeroplanes for business, leisure, and emergency purposes. Air travel shrinks distances, facilitates trade, fosters cultural exchange, and aids disaster response. The efficient functioning of air travel is vital for an interconnected and mobile world. Through precise categorization and anticipation of delay reasons using diverse attributes, this model aims to take proactive steps to predict the delay in a flight.

1. Introduction

Flight delays have a profound effect on both passengers and the aviation sector. They bring about inconveniences for travellers, including missed connections and disrupted itineraries. Additionally, delays lead to operational inefficiencies, elevated expenses, and reduced airline customer satisfaction. Hence we intend to predict the delay in a flight to enhance user experience and cut down on inefficiency costs.

2. Related Work/Literature

1. `https://dl.acm.org/doi/fullHtml/10.1145/3497701.3497725`: This paper performs binary classification using seven classification models. Five models were base classifiers and two were ensemble classifiers. The dataset used for this spans over one year for flights from the JFK airport. It uses four evaluation scores; accuracy, precision, recall, and the f1-score. Due to a class imbalance, weighted evaluation scores are used. Decision Tree had the best performance on their dataset out of all the seven models they used.

2. `https://www.hindawi.com/journals/cin/2020/8878681/`: This paper analyses the flight time deviation of Lithuania airports. They utilize SMOTE (Synthetic Minority Oversampling Technique) analysis for the class imbalance. They also utilised seven supervised learning models and also implemented grid search to find the best parameters which give the highest accuracy for each model. They used five evaluation scores; sensitivity, precision, specificity, F-measure and accuracy. Gradient Boosted tree had the best performance on their dataset.

3. `https://aircconline.com/ijdkp/V8N3/8318ijdkp01.pdf`: This paper uses eight classifiers; four rule-based algorithms, and four tree-based algorithms for analyzing the flight delay pattern in Egypt Airline's Flight dataset. This was implemented in Java using the WEKA data mining tool. They used the confusion matrix for each model for seven evaluation scores; sensitivity, specificity, precision, recall, F-measure, ROC, and accuracy. The rule-based algorithm classifiers.rules.PART performed best with the highest accuracy however REPtree was the most efficient with respect to accuracy and running time.

4. `https://ieeexplore.ieee.org/abstract/document/8102138`: This papers conducts flight delay prediction in two steps: classification and regression. Both arrival and departure delays were taken into consideration. Classification was used to check whether a flight was delayed or not. For this, the authors use four models, of which the Gradient Boosting Classifier gave the highest accuracy. Regression was used to calculate the actual amount of delay. Four regression models were utilised and Extra-Trees Regressor gave the best results. Accuracy, precision and recall were used to evaluate classification results while MSE and R2 score were used to evaluate regression results.

3. Dataset

Using the Kaggle Dataset, we had data about flights leaving from JKF airport between Nov 2019 and Dec 2020. With no preprocessing, we had 23 columns/features present in the dataset.

Following are the preprocessing steps and dataset analysis we did:

1. We checked the data for NaN entries and removed all of the required samples. This accounted for only two samples.

2. We dropped six columns that we considered unnecessary for our delay analysis. We dropped columns like 'OP_UNIQUE_CARRIER', 'TAIL_NUM', 'CRS_DEP_M', 'DEP_TIME_M', 'CRS_ARR_M' and 'TAXI_OUT which are irrelevant to our predictor overall. These features are completely independent and don't affect the delay in any way. Moreover, 'CRS_DEP_M' and 'DEP_TIME_M' can mislead the models since from these two columns, delay can be inferred directly.

Column Name	Data Type
MONTH	int64
DAY_OF_MONTH	int64
DAY_OF_WEEK	int64
OP_UNIQUE_CARRIER	object
TAIL_NUM	object
DEST	object
DEP_DELAY	int64
CRS_ELAPSED_TIME	int64
DISTANCE	int64
CRS_DEP_M	int64
DEP_TIME_M	int64
CRS_ARR_M	int64
Temperature	int64
Dew Point	object
Humidity	int64
Wind	object
Wind Speed	int64
Wind Gust	int64
Pressure	float64
Condition	object
sch_dep	int64
sch_arr	int64
TAXI_OUT	int64

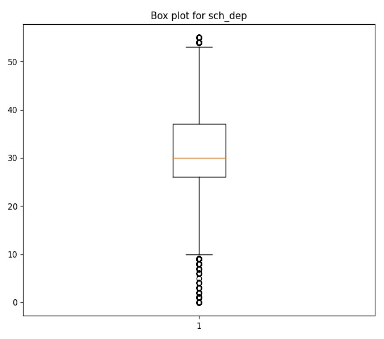
Table 1. Description of Dataset

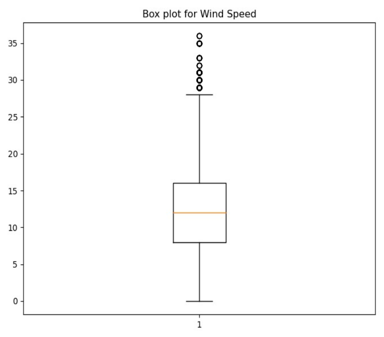
3. The “DEW_POINT” column was changed from object datatype to integer datatype, and its null values were dropped.

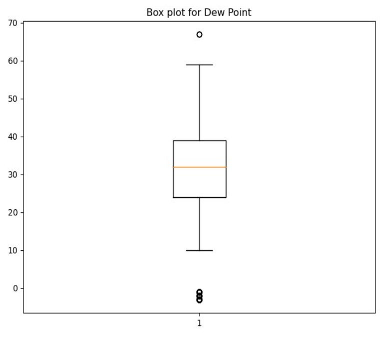
4. For our initial classification models, we had to change our ’DEP_DELAY’ column. Our initial column was the number of minutes by which the flight was delayed. For now, we changed it to a 0/1 format where we encoded any flight that is delayed by fifteen minutes or more to 1 and 0 otherwise. This column was moved to the end. Following this, for regression, we kept the ’DEP_DELAY’ column as is and added the ground truth of classification (0/1 format) to a separate column ’delayed’.

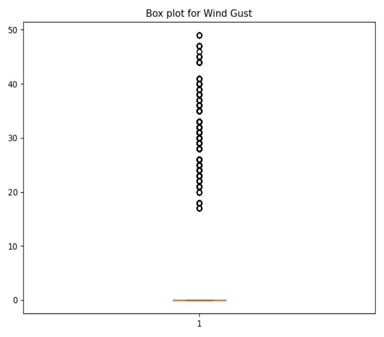
5. We performed Exploratory Data Analysis by plotting the Boxplots and the Correlation Matrix. From the boxplots, we can infer the distribution of the data varies quite a lot for different columns (due to the number of columns, we cannot show all the boxplots, we have shown 4 to highlight the variation). Through this, we realized the need to normalize the data. We scaled the data using Z-normalisation. From the correlation plot, there is no clear indication of which feature may affect the delay the most. There are clear correlations between different weather aspects which highlights the fact that the dataset might be accurate with its readings.

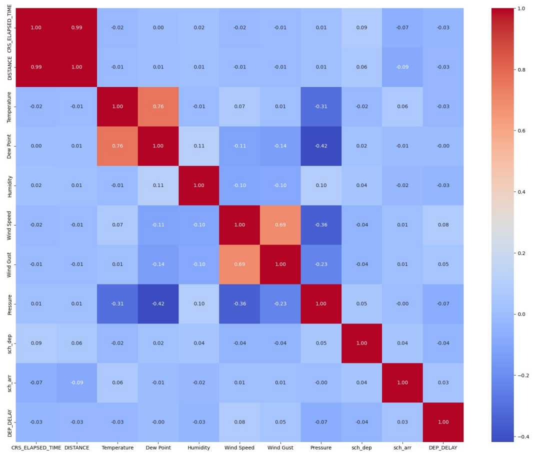
6. We have three categorical columns: ’DEST’, ’Wind’, ’Condition’. We label encoded the columns as part of our data preprocessing.











4. Methodology and Models

Following data preprocessing, we split the dataset into train and test and ran several classification models. Classification is done to check whether a delay is present or not. If the value in the delay column is 1, i.e. delay is present, we will apply regression to calculate the delay. We also ran K fold cross validation to train the model and calculate the average accuracy rates of each model.

K fold cross validation: K fold cross validation is used to test and compare a model and its performance. The dataset is divided into K subparts and run with one part being the test set and the rest being training data. This process is repeated K times i.e. by leaving out one set or fold as test data each time. The final average accuracy is given by the average of all these intermediate averages.

We set K=10 and used the following classification models with the following hyperparameters after using GridSearchCV:
(If not mentioned, default parameters to be assumed)

1. Decision Tree: The parameters set were; criterion='entropy' and class_weight='balanced'.
2. Random forest classifier: The parameters set were; criterion='entropy'and n_estimators=1700.
3. Naive Bayes classifier: The parameter set was; var_smoothing=001.
4. KNN classifier: Default parameters were used.
5. Ada Boost classifier: Default parameters were used.
6. Gradient Boosting classifier: The parameters set were; n_estimators=950.
7. Logistic Regression: The parameters set were; c=0.001,solver='liblinear',multi_class='ovr' and penalty='l1'.
8. SVM: The parameter set was; probability='True'.
9. XGB Classifier: This classifier was from the xgboost module. The parameters set were; subsample=0.8, reg_lambda=0.5, reg_alpha=1, n_estimators=800, max_depth=900, learning_rate=0.01, gamma=2 and colsample_bytree=0.6.

After these models were trained, SHAP analysis was done on Gradient Boosting Classifier and XGBClassifier to analyse which feature affects the delay the most.

Evaluation Metrics: We obtained the confusion matrix

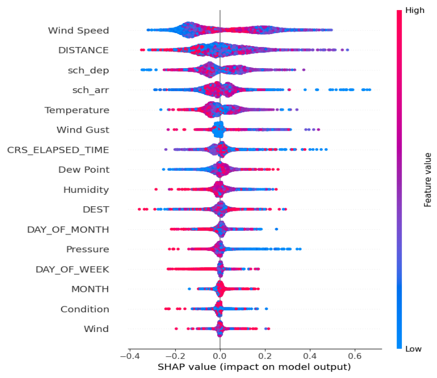


Figure 1. XGBoost SHAP Analysis

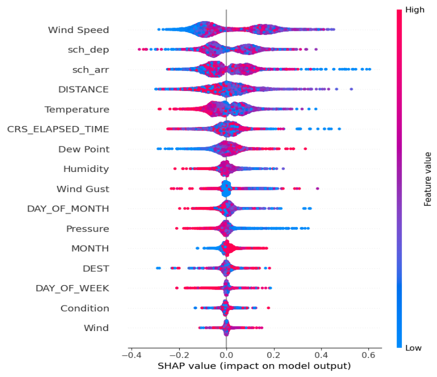


Figure 2. Gradient Boost SHAP Analysis

for all the models and using the inbuilt classification report feature of sklearn, got all the required scores for our models. We used the following metrics to analyse our models.

1. Precision
2. Recall
3. F1 score
4. Accuracy Score

The higher these metrics for a model, the better the performance of the model.

Regression: Following classification, class imbalance was managed to avoid bias. This was done using SMOTE analysis. Regression was the next step.Their parameters were tuned using RandomisedSearchCV. K fold Cross Validation was also done and K was set to 10. The models used were as follows:
(If not mentioned, default parameters to be assumed)

1. K-nearest-neighbours: The parameters set were; weights='distance', n_neighbors=3, metric='manhattan'
2. Gradient Boosting: The parameters set were;subsample=0.9,min_samples_leaf=2 and max_depth=4.
3. XGB Regressor: This regressor was from the xgboost module. The parameter set was; n_estimators=130.
4. Decision Trees Regressor: The parameters set were; min_samples_leaf=4 and max_features='sqrt'.
5. Random Forest Regressor: The parameters set were;n_estimators=230 and max_depth=10.
6. Linear Regression: Default parameters were used.
7. Ridge regularisation: Default parameters were used.
8. Lasso regularization: The parameter set was; alpha=0.1.
9. Support Vector Regression: Default parameters were used.
10. AdaBoost Regressor: Default parameters were used.

Evaluation Metrics: For our regression models, we used metrics different from the classification models. We used the following metrics:

1. Mean Square Error
2. R2 score

The lower the MSE for a model, the better it performs. The closer R2 is to 1, the better the model.

5. Results and Analysis

Classification:

Table 2 shows the performance of various machine learning models. Notably, Decision Trees, Random Forest, Gradient Boosting and XGBoost show the highest accuracies on both the training and testing sets, indicating their robust generalization capabilities. XGBoost achieves an accuracy of 92.66 on the testing data, followed closely by Gradient Boosting with 91.96 and Random Forest with 91.29. Logistic Regression has the worst accuracy in comparison i.e. 55.34%.

Model Name	Training Acc.	Testing Acc.
D. Tree	86.80	86.67
RF	91.11	91.29
KNN	80.93	81.21
NB	58.00	58.14
Log. Reg	55.44	55.34
SVM	62.01	62.40
GB	91.89	91.96
AdaBoost	74.41	74.31
XGBoost	92.79	92.66

Table 2. Model Performance Comparison

Model	Class	Precision	Recall	F1-Score
D. Tree	Class 0	0.86	0.87	0.87
	Class 1	0.87	0.87	0.87
RF	Class 0	0.90	0.93	0.91
	Class 1	0.93	0.89	0.91
KNN	Class 0	0.90	0.69	0.78
	Class 1	0.76	0.93	0.83
NB	Class 0	0.59	0.51	0.55
	Class 1	0.58	0.65	0.61
Log. Reg	Class 0	0.55	0.54	0.55
	Class 1	0.56	0.56	0.56
SVM	Class 0	0.61	0.64	0.63
	Class 1	0.63	0.61	0.62
GB	Class 0	0.87	0.98	0.92
	Class 1	0.98	0.86	0.92
AdaBoost	Class 0	0.74	0.74	0.74
	Class 1	0.75	0.75	0.75
XGBoost	Class 0	0.88	0.98	0.93
	Class 1	0.98	0.87	0.92

Table 3. Performance Metrics for Binary Classification Models

Table-3 results show the performance metrics (precision, recall, and F1-score) of various machine learning models across two classes (Class 0 and Class 1). Among the models, XGBoost consistently exhibits the highest metrics, with impressive precision and recall for both classes. Decision Trees and Random Forest follow closely, showcasing balanced metrics. Notably, Naive Bayes lags behind, displaying lower scores across precision, recall, and F1-Score, making it the least performing model in this binary classification evaluation.

Regression:

Table 4 shows that in the domain of regression analysis, K-Nearest Neighbours emerges as the preeminent model, demonstrating superior predictive capabilities with the lowest Mean Squared Error (MSE) on both the training and test datasets, complemented by elevated R^2 scores. Followed by XGBoost, exhibiting robust performance with competitive metrics. Linear Regression, Ridge, and Lasso yield comparable results, signifying poor predictive efficacy. Conversely, AdaBoost and SVR faces challenges, evidenced by negative R^2 values, indicating suboptimal model fit. K-Nearest Neighbors achieves commendable performance with a noteworthy R^2 score. In summation, K-Nearest Neighbour and XGBoost distinguish themselves for their heightened predictive precision in this regression context.

Model	**Train**		**Test**	
	MSE	R^2	**MSE**	R^2
KNN	1677.12	0.64	1569.56	0.68
GB	2976.23	0.37	3159.80	0.35
AdaBoost	7371.10	-0.57	6371.42	-0.29
Ridge	4464.48	0.05	4644.43	0.05
Lasso	4469.52	0.05	4652.91	0.05
Lin. Reg	4464.48	0.05	4644.38	0.05
RF	3102.40	0.34	3289.54	0.33
XGBoost	2199.39	0.53	2315.34	0.53
D. Tree	3743.06	0.20	3819.77	0.22
SVR	4957.62	-0.04	5165.62	-0.04

Table 4. Model Performance Comparison

Best Models:
Using the best models obtained for both classification and regression, a final evaluation was done on unseen data. Their respective evaluation metrics were calculated and they were as follows:
Best Classifier: XGB Classifier

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.91	0.99	0.95	4705
1	0.99	0.90	0.94	4672
Accuracy			0.95	9377
Macro Avg	0.95	0.95	0.95	9377
Weighted Avg	0.95	0.95	0.95	9377

Table 5. Classification Metrics for XGB

Best Regressor: KNN

Metric	**Value**
R-squared	0.68
Root Mean Squared Error (RMSE)	48.92
Mean Squared Error (MSE)	2392.70

Table 6. Regression Evaluation Metrics for KNN

6. Conclusion

The aim of our project was to be able to predict the delay of a given flight by training a model on a dataset with previous delays and features that affect them. We procured a dataset that could be used for this, pre-processed the data, fit classification models to check for delays and finally used regression to predict the amount of delay that could be expected. For classification, XG Boost gave the highest accuracy. Following classification, a class imbalance was noticed. This was addressed using SMOTE analysis and finally, regression was carried out. Several models were fitted, out of which K-nearest neighbours had the best performance.

We were able to predict the amount of delay that could be expected in accordance with several influencing factors. This project could further help analyze the major causes of delays and be used to address these. Delays affect both the passenger as well as the airline. With due effort and analysis, these can be reduced and inconveniences managed, resulting in higher satisfaction for all involved parties.

References

[1] I. Gheorghiu, "Historical flight delay and Weather Data USA," Kaggle, `https://www.kaggle.com/datasets/ioanagheorghiu/historical-flight-and-weather-data`.

[2] D. Kansal, "Flight take off data - JFK airport," Kaggle, `https://www.kaggle.com/datasets/deepankurk/flight-take-off-data-jfk-airport`.

[3] B. Ye, B. Liu, Y. Tian, and L. Wan, "A methodology for predicting aggregate flight departure delays in airports based on supervised learning," Sustainability, vol. 12, no. 7, p. 2749, 2020. doi:10.3390/su12072749. `https://www.mdpi.com/2071-1050/12/7/2749`

[4] Y. Tang, "Airline flight delay prediction using machine learning models," 2021 5th International Conference on E-Business and Internet, 2021. doi:10.1145/3497701.3497725

[5] A. Chiu, "A step by step example in binary classification," Medium, `https://shorturl.at/huyS1`