KINGDOM OF SAUDI ARABIA
Ministry of Education
University of Jeddah
College of Computer Science and Engineering

المملكة العربية السعودية
وزارة التربية
جامعة جدة
كلية علوم وهندسة الحاسب - جدة

Flight Delay Prediction using XGBoots Model

CCAI 312 - Pattern Recognition

Shouq Ahyaf - 2211559
Renad Alharbi - 2210625
Jori Baaljahr - 2210778

Dr. Saffa alsfry
28 November - 2024 A.D

1 INTRODUCTION

The latest hype in the airline industry is to predict flight delay as this aspect plays a huge role in managing the efficiency of airlines, improving customer experience, and reducing operational cost [1]. Delays are typically caused by airlines and give inconveniences on the schedules, missed connections, and customer dissatisfaction. Airport management should also be able to predict delays to ensure effective use of airport infrastructure, gate assignments, and provision of information to passengers in a timely manner. Flight delays bring about many negative effects [3], not only causing losses to airlines and passengers [3], but also causing conflicts between passengers and airlines [4] and reducing the operational efficiency of the entire civil aviation industry [5].

Being able to accurately predict flight delays helps airlines take proactive steps such as rescheduling crew, redirecting flights, or advising passengers well in advance. Such predictions can also assist the airports in better management of their resources, avoid overbookings, and smooth the entire operational flow. In light of this, predicting flight delays with reasonable accuracy is a key area of interest for the aviation industry [2].

The advent of ML algorithms has provided very powerful tools for solving complex prediction problems. The algorithms are more suitable especially when handling large datasets which are common in the aviation sector. Such algorithms have the ability to capture various complex patterns and relationships associated with different variables like flight time, weather conditions, aircraft type, and the historical delays of an airway. Predictive models can automatically identify such a pattern and, therefore make reliable predictions about the occurrence of a delay, which is based on the occurrence of historical data.

This report is on the usage of XGBoost, as it is considered the state-of-the-art machine learning algorithm for the purpose of the prediction [6]. XGBoost is quite famous for achieving high efficiency and performance regarding structured datasets, and as such, its usage is presented in this report for prediction of flight delays. The same report reveals how the tuning of the hyperparameters associated with the XGBoost model improves predictive accuracy [7]. Hyperparameter tuning is an important step in machine learning because it allows the model to optimize its parameters for better generalization and performance.

The core objective of this study is to evaluate the performance of the XGBoost model after fine-tuning its hyperparameters and compare its results with other machine learning classifiers such as K-Nearest Neighbors (KNN) [8] and Naive Bayes [9]. Though this report briefly mentions these models in comparison, it is mainly focused on the XGBoost algorithm. From this comparison, it will show the strength and weaknesses of XGBoost about the flight delay prediction model to check if it's the best one for this particular task.

The XGBoost algorithm is known for handling large datasets and for performing well on classification problems. Therefore, it is to be tested with the set of performance metrics involving accuracy, precision, recall, and F1-score. Results obtained from the hyperparameter-tuned XGBoost model will give valuable insights as to how well it could predict flight delays in comparison to the much simpler models such as KNN and Naive Bayes.

Through results obtained from the fine-tuning process and comparisons among various machine learning models, this report will provide an all-rounded understanding of how flight delays can best be predicted with high accuracy and reliability.

2 PROBLEM DESCRIPTION

Flight delays are a commonplace problem in the aviation business, resulting in lost hours, increased costs, and worse customer experiences. Accurate prediction of flight delays can help airlines deal with schedules better and spare passengers from the hassle associated with flight delays. Classifying whether a flight would be delayed or not using factors such as departure times, distance, cancellations, and other operational data is the problem this project addresses.

The target variable in this case is ARR_DEL15, which indicates whether the flight arrived 15 minutes later than scheduled (delayed = 1, on-time = 0). The objective is to predict this binary outcome using machine learning models.

Several flight-related attributes influence delays, including:

1. **Departure Time and Distance:** Delays can accumulate over time, and longer flights may face more operational challenges.
2. **Day of the Week, Carrier, and Cancellations:** Delays may be caused due to airline operations, busy travel time, and cancellations.
3. **External Conditions:** Weather and airport may also play a significant impact on delays.

The work is to create a Machine Learning model using XGBoost to predict flight delay accurately. The fine-tune the model and evaluate whether the model performs well such that airlines can improve operating efficiency and satisfy customers if delays are predicted ahead.

3 DATA DESCRIPTION

The dataset for this project is flight information in January 2019 and contains multiple features concerning that flight's departure and arrival. The following are some of the main attributes for this dataset:

- **DAY_OF_MONTH:** Day of the month on which the flight event occurs.
- **DAY_OF_WEEK:** The day of the week on which the flight happened.
- **OP_UNIQUE_CARRIER:** A unique identifier for the operating airline.
- **ORIGIN DEST:** Composite feature representing origin and destination airports.
- **DEP_DEL15:** Whether the flight was delayed at departure (1 for delayed, 0 for not delayed).
- **CANCELLED:** Whether the flight was canceled (1 if cancelled, and 0 otherwise).
- **DIVERTED:** Whether the flight was diverted (1 for diverted, 0 for not).
- **DIST_GROUP:** Distance of flight (Short, Medium, Long).
- **ARR_DEL15:** The flight was delayed at arrival (target variable: 1 for delayed, 0 for not delayed).

The dataset is preprocessed by managing missing values, encoding categorical variables and designing a combined feature of origin and destination of flights. Distance feature also got categorized into groups: Short, Medium, Long; so distances can be well captured by distance and delays.

4 METHODOLOGY

XGBoost Algorithm

XGBoost, Extreme Gradient Boosting, is one of the powerful machine learning algorithms and decision tree ensembles. However, it mainly attracts by its efficiency as well as accuracy. Because of the reasons mentioned, it was one of the most prominent algorithms for any structured data prediction tasks. It develops multiple decision trees sequentially but corrects mistakes in such a way that every mistake done by any of these trees is rectified and improved in a new sequence. The strong sides are:

- **Gradient boosting**: XGBoost does gradient boosting to minimize error iteratively, which explains why it is such a strong candidate for classification problems.
- **Regularization:** XGBoost includes L1 and L2 regularization to prevent overfitting, enhancing model generalization.
- **Parallelization:** It supports parallelized training, which makes it faster than many other boosting algorithms.
- **Tree Pruning:** XGBoost uses the technique known as "max depth" to limit the depth of trees so that it does not overfit.

Hyperparameter Tuning

Hyperparameter tuning was done using a GridSearchCV approach, where different values for hyperparameters such as learning_rate, max_depth, n_estimators, subsample, and colsample_bytree were tested. This was to find the best-performing combination of those parameters. The final selected hyperparameters were:

- **learning_rate:** 0.05
- **max_depth:** 3
- **n_estimators:** 150
- **subsample:** 1.0
- **colsample_bytree:** 0.6

and some other factors also

All of these fine-tuned parameters trained the XGBoost model to have a performance evaluation upon the test set.

5 EXPERIMENTS AND RESULTS

The model was evaluated using various metrics, including accuracy, precision, recall, and F1-score. The evaluation focused on determining how well the XGBoost model predicted flight delays, with results indicating its high performance.

Model Performance:

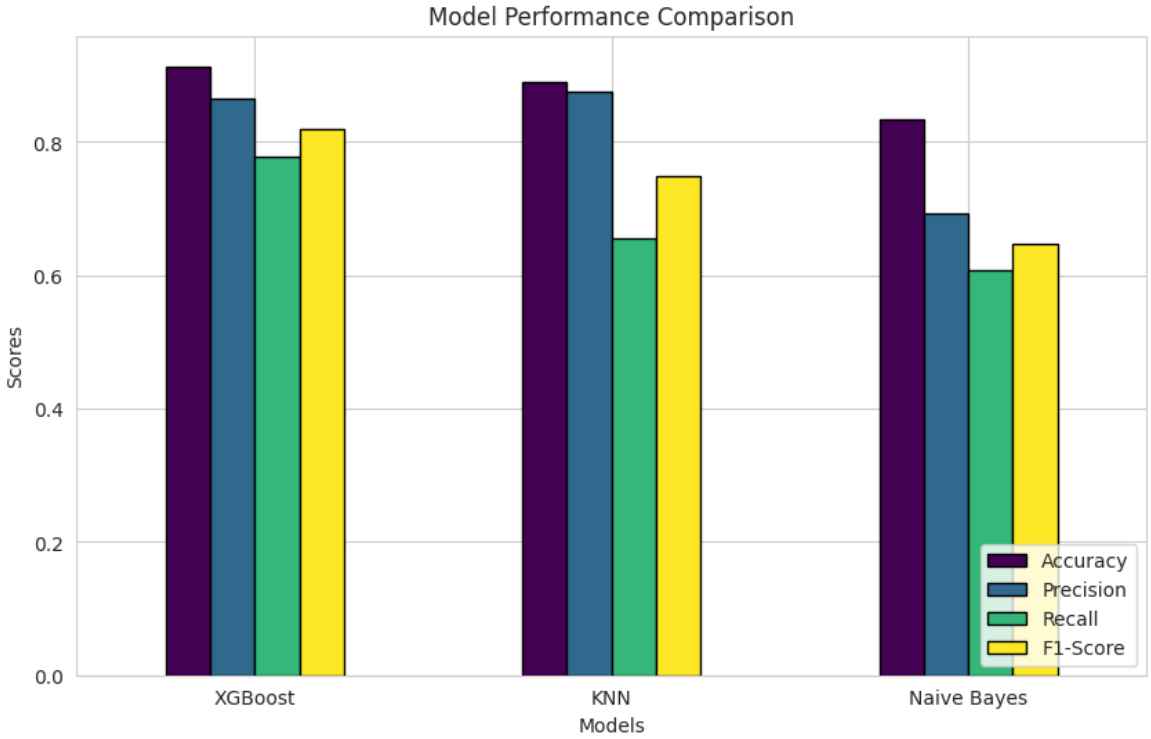
The performance metrics for the XGBoost model after hyperparameter tuning were as follows:

- **Accuracy**: 91.37%
- **Precision**: 86.53%
- **Recall**: 77.84%
- **F1-Score**: 81.95%

Comparison with Other Models:

For comparison, KNN and Naive Bayes classifiers were also tested using the same dataset. The results are summarized below:

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	91.37%	86.53%	77.84%	81.95%
KNN	88.99%	87.61%	65.50%	74.96%
Naïve Bayes	83.38%	69.40%	60.75%	64.79%

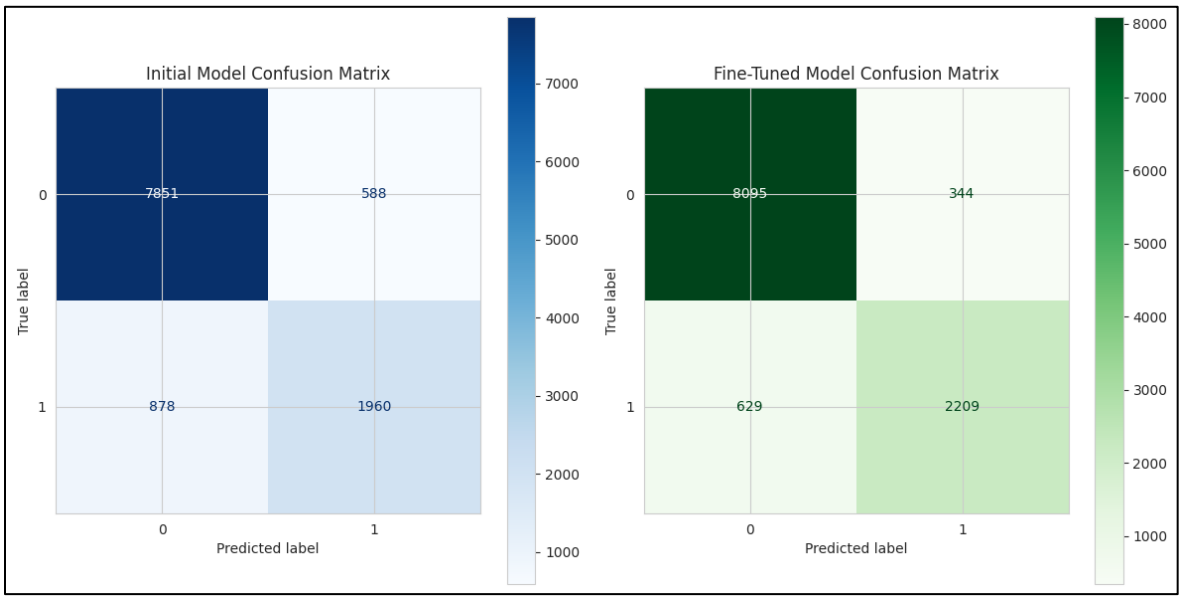


XGBoost outperformed both KNN and Naive Bayes in terms of accuracy, precision, and F1-score. Although KNN exhibited higher precision, its recall was significantly lower compared to XGBoost. Naive Bayes showed the lowest performance across all metrics.

Confusion Matrix & ROC Curve for XGBoost:

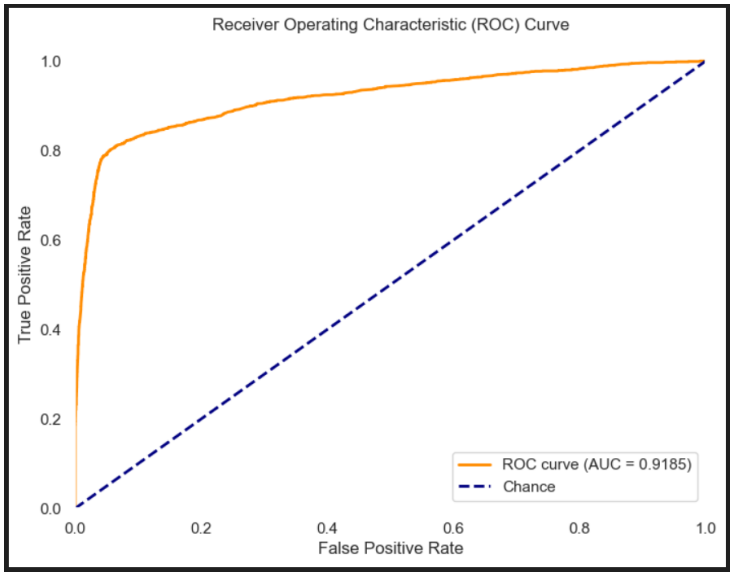
Confusion Matrix:

The confusion matrix for XGBoost shows the true positives, true negatives, false positives, and false negatives. This provides insights into how well the model distinguishes between delayed and non-delayed flights.



ROC Curve:

The Receiver Operating Characteristic (ROC) curve shows the trade-off between the true positive rate (sensitivity) and the false positive rate. XGBoost showed a strong performance with a high Area Under the Curve (AUC).



6 DISCUSSION

After fine-tuning, the XGBoost model was more accurate with regard to flight delay predictions compared to other classifiers like KNN and Naive Bayes. Hyperparameter tuning greatly improved the model's generalizability and resulted in a high value of accuracy, precision, and recall with the final configuration.

Regularization techniques used in XGBoost prevented overfitting, a common issue when developing machine learning models for working with such large data. One of the major strengths of XGBoost has been its ability to perform well with large numbers of features, especially when applied to complex data like this flight information.

However, the KNN model was also performing well with respect to precision, which means it might be more applicable in scenarios where the former is preferred over the latter. On the contrary, Naive Bayes was the worst performer and, therefore, had less precision and recall, thus possibly not being the most ideal model for this particular type of classification problem.

7 CONCLUSION

This work successfully applied XGBoost for predicting flight delays by further fine-tuning the model's hyperparameters for good performance. It demonstrated XGBoost to outperform other classifiers like KNN and Naive Bayes in predicting flight delays and thus its efficiency. Possible Future Work Further feature engineering. Incorporate more features flight-related such as weather information. Explore other ensemble methods to further improve the models.

REFERENCES

[1] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM Sigkdd international conference on knowledge discovery and data mining. 2016, p. 785–94. Available: https://arxiv.org/abs/1603.02754

[2] Analytics Vidhya, "Understanding the Math behind the XGBoost Algorithm," Analytics Vidhya, Sep. 06, 2018.
Available: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

[3] "Class XGBClassifier (1.27.0)," Google Cloud, 2024. Available: https://cloud.google.com/python/docs/reference/bigframes/latest/bigframes.ml.ensemble.XGBClassifier .

[4] "XGBoost for Regression," GeeksforGeeks, Aug. 29, 2020. Available: https://www.geeksforgeeks.org/xgboost-for-regression/

[5] "Get Started with XGBoost — xgboost 1.7.5 documentation," xgboost.readthedocs.io. Available: https://xgboost.readthedocs.io/en/stable/get_started.html

[6] Analytics Vidhya, "Understanding the Math behind the XGBoost Algorithm," Analytics Vidhya, Sep. 06, 2018.
Available :https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

[7] D. Agrawal, "January Flight Delay Prediction," Kaggle.com, 2020, doi: Available: https://www.kaggle.com/datasets/divyansh22/flight-delay-prediction/data