RAMCO INSTITUTE OF TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to Anna University

NAAC Accredited with 'A+' Grade & An ISO 9001: 2015 Certified Institution

NBA Accredited UG Programs: CSE, EEE, ECE and MECH

CS3491 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Flight Delay Prediction

Submitted by

KIRRAN S T(953623205025)

DHARANIBALAN R(953623205009)

DINESH KUMAR S(953623205010)

GOWTHAM A(953623205014)

In partial fulfilment for the award of the degree of
BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

RAMCO INSTITUTE OF TECHNOLOGY
RAJAPALAYAM – 626 117

ANNA UNIVERSITY: CHENNAI 600025

JUNE 2025

ANNA UNIVERSITY: CHENNAI 600 025

TABLE OF CONTENTS

ABSTRACT

Flight delay prediction is a critical area in aviation analytics, aimed at improving operational efficiency and passenger satisfaction. Predicting whether a flight will be delayed has become a valuable application for data scientists, especially with the increasing availability of historical flight data.

This mini-project focuses on building a machine learning model to accurately predict flight delays using past flight records. The dataset includes features such as departure time, airline carrier, origin, destination, and flight distance. The data was preprocessed to manage missing values, and categorical variables were encoded to ensure compatibility with machine learning algorithms. Models such as Naïve Bayes and Random Forest were implemented and evaluated.Among the models tested, the Random Forest Classifier achieved the highest accuracy, demonstrating its effectiveness in capturing complex patterns and handling categorical data efficiently.The trained model can predict whether a flight is likely to be delayed based on given input parameters. This project highlights the power of machine learning in aviation analytics and lays the groundwork for more advanced prediction systems that can incorporate real-time data, weather conditions, and airport traffic in future enhancements.

III LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation	Full Form
AI	Artificial Intelligence
ML	Machine Learning
IPL	Indian Premier League
CSV	Comma Separated Values
RF	Random Forest
DT	Decision Tree
LR	Logistic Regression
SVM	Support Vector Machine
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error

INTRODUCTION

1.1 AIM OF THE PROJECT:

The aim of this project is to develop a machine learning-based prediction system that can accurately forecast whether a flight will be delayed, using historical flight data. The system utilizes key flight features such as scheduled departure time, carrier, origin, destination, and flight distance to detect patterns associated with delays.

By training and evaluating different machine learning models, the project seeks to identify the most effective algorithm for accurately classifying flight statuses. The final model is intended to support airline operations, improve passenger experience, and assist in proactive decision-making.

This project also aims to highlight the practical application of artificial intelligence in the aviation sector, demonstrating how data-driven approaches can address real-world challenges and optimize transportation systems.

1.2 OBJECTIVE OF THE PROJECT:

- To collect and understand historical flight data, including key features such as departure time, airline carrier, origin, destination, and flight distance.
- To preprocess the dataset by handling missing values and encoding categorical variables for machine learning compatibility.
- To explore and analyze the data to identify significant factors influencing flight delays.
- To apply various machine learning algorithms to train predictive models for flight delay classification.
- To evaluate and compare the performance and accuracy of the implemented models.
- To develop a prediction system that can determine whether a flight is likely to be delayed based on given input parameters.

II LITERATURE SURVEY

2.1 INTRODUCTION:

Flight delays are a major concern in the aviation industry, impacting airline operations, passenger satisfaction, and overall travel efficiency. With the increasing availability of historical flight data, machine learning offers promising solutions for predicting delays and enabling proactive decision-making.

This project focuses on developing a machine learning-based system to predict whether a flight will be delayed based on key attributes such as scheduled departure time, airline carrier, origin, destination, and flight distance. Accurate delay prediction can help airlines optimize scheduling, improve resource allocation, and enhance the travel experience for passengers.

By leveraging past research and applying modern classification algorithms like Naïve Bayes and Random Forest, this project aims to identify patterns and build a reliable model for forecasting delays. The use of machine learning in this context not only demonstrates its practical value in transportation analytics but also contributes to smarter, data-driven management in the aviation sector.

III EXISTING SYSTEM

3.1 SYSTEM MODEL:

- **Data Collection:** Historical flight data is collected from reliable sources such as the U.S. Department of Transportation, Kaggle datasets, or airline databases.
- **Data Preprocessing:** The dataset is cleaned to handle missing values and remove irrelevant information. Categorical features such as carrier, origin, and destination are encoded to ensure compatibility with machine learning models.
- **Feature Selection:** Key features like month, day, hour, minute, carrier, origin, destination, and distance are selected for training the model. The target variable is defined as whether a flight is delayed by more than 15 minutes.
- **Model Training:** Machine learning algorithms such as Naïve Bayes and Random Forest are applied to train predictive models using the processed data.

- **Model Evaluation:** The models are evaluated on a test dataset to assess their performance in terms of accuracy, precision, recall, and confusion matrix results.
- **Prediction:** Given new flight details, the trained model predicts whether the flight will be delayed or on time.

3.2 LITERATURE CONCLUSION:

From the reviewed literature, it is evident that machine learning techniques such as Naïve Bayes and Random Forest have been effectively applied for flight delay prediction tasks. These models, especially when trained on clean and well-prepared datasets, can offer meaningful insights and reasonably accurate predictions based on key flight-related features such as departure time, airline carrier, origin, destination, and flight distance.

Random Forest, in particular, has been highlighted in various studies for its superior performance due to its ensemble nature and ability to model complex, non-linear relationships within the data. Naïve Bayes, while simpler, remains a strong baseline due to its speed and interpretability, especially when the features are relatively independent.

This project builds upon prior research by focusing on these two models—Naïve Bayes and Random Forest—implementing them on a historical flight dataset. Effective data preprocessing, categorical encoding, and model evaluation strategies were employed to ensure accurate and reliable delay predictions. The literature review has provided critical guidance in selecting appropriate models and methods to enhance the effectiveness of the developed prediction system.

IV PROPOSED WORK

4.1 INTRODUCTION:

The proposed work aims to develop an accurate and efficient machine learning model to predict whether a flight will be delayed, using historical flight data. Unlike existing systems that often rely on limited features or outdated techniques, this project focuses on applying effective data preprocessing and selecting appropriate machine learning models to improve prediction accuracy.

The system utilizes key flight-related features such as departure time, airline carrier, origin, destination, and distance. The data is carefully cleaned, missing values are handled, and categorical variables are encoded to prepare it for model training. In this project, machine learning models such as Naïve Bayes and Random Forest are implemented and compared based on their prediction performance.

The main goal of the proposed system is to identify the most accurate model, evaluate its performance metrics, and use it to predict flight delay status based on user-provided flight details. The system is designed to be simple, reliable, and a practical demonstration of how AI/ML can be applied to solve real-world problems in the aviation industry.

4.2 System Framework:

1. **Data Collection:** Collect past flight data including time, carrier, route, and delay info.
2. **Data Preprocessing:** Clean data, handle missing values, and convert text to numbers.
3. **Feature Selection:** Choose important details like time, airline, origin, destination, and distance.

4. **Model Training:** Train Naïve Bayes and Random Forest models using the prepared data.
5. **Model Testing:** Check how accurate the models are using test data.
6. **Prediction:** Use the best model to predict if a flight will be delayed or not.
7. **Result Display:** Show the prediction result clearly to the user.

Here's a **complete and simple write-up** for section **4.3 Proposed Methodology** and all its sub-sections (Random Forest, Naïve Bayes.) for your **Flight Delay Prediction** mini-project:

4.3 PROPOSED METHODOLOGY:

The proposed methodology involves using machine learning algorithms to predict whether a flight will be delayed. After collecting and preprocessing the flight data, we train and evaluate different models to select the one that provides the best accuracy and performance.

4.3.1 Naïve Bayes (GaussianNB):

Naïve Bayes is a simple probabilistic classifier based on Bayes' Theorem. It assumes that all features are independent and calculates the probability of delay based on input values like departure time, carrier, and flight distance. Despite its simplicity, it performs well on small and clean datasets.

- **Pros:** Fast, easy to implement, works well with less training data.
- **Cons:** Assumes feature independence, which may not always be true.

4.3.2 Random Forest Classifier:

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs. It is powerful in handling both categorical and numerical features, and it reduces the risk of overfitting while improving overall prediction accuracy.

- **Pros:** High accuracy, robust to outliers and missing values, handles complex data well.
- **Cons:** Slower than simple models, less interpretable due to its ensemble nature.

V SYSTEM SPECIFICATION

5.1 Software Requirement:

S.No	Software Component	Description / Purpose
1	Operating System	Windows 10 / Linux / macOS
2	Programming Language	Python 3.x
3	IDE / Code Editor	Jupyter Notebook / Visual Studio Code / PyCharm
4	Python Library – pandas	For data loading and preprocessing
5	Python Library – numpy	For numerical operations and arrays
6	Python Library – matplotlib	For plotting graphs and charts
7	Python Library – seaborn	For advanced data visualizations
8	Python Library – scikit-learn	For machine learning models and evaluation
9	Web Browser	Chrome / Firefox (to view Jupyter Notebooks or web output)

5.2 Hardware Requirement:

S.No	Hardware Component	Minimum Specification	Recommended Specification
1	Processor	Intel Core i3 or equivalent	Intel Core i5/i7 or higher
2	RAM	4 GB	8 GB or more
3	Hard Disk	500 MB free space	1 GB or more
4	Display	Standard 14-inch display	Full HD display
5	Graphics	Integrated graphics	Dedicated GPU (optional for deep learning)
6	Input Devices	Keyboard and Mouse	Keyboard and Mouse
7	Internet Connection	Required (for installing libraries, datasets)	Stable broadband connection recommended

5.3 Dataset Description:

Column Name	Data Type	Description
Column Name	**Data Type**	**Description**
`year`	Integer	The year in which the flight was scheduled
`month`	Integer	Month of the flight (1–12)
`day`	Integer	Day of the month when the flight was scheduled
`day_of_week`	Integer	Day of the week (1 = Monday, 7 = Sunday)
`dep_time`	Integer	Scheduled departure time (in hhmm format)
`arr_time`	Integer	Scheduled arrival time (in hhmm format)
`carrier`	Object	Airline carrier code (e.g., 'AA' for American Airlines)
`origin`	Object	Airport code of origin
`dest`	Object	Airport code of destination
`distance`	Float	Distance between origin and destination in miles
`arr_delay`	Float	Arrival delay in minutes (target used to classify delay)
`is_delayed`	Integer	Target column: 1 if `arr_delay` > 15 minutes, else 0
`hour`	Integer	Extracted hour of departure (used for model input)
`minute`	Integer	Extracted minute of departure (used for model input)

VI IMPLEMENTATION AND RESULTS

ARCHITECTURE:

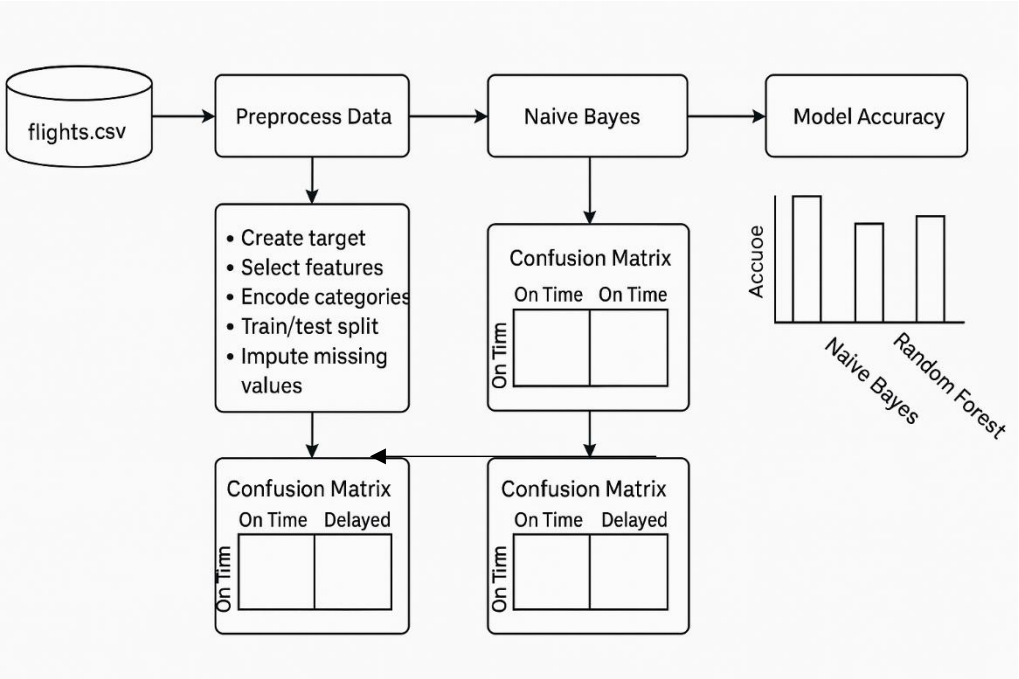


FIG 1.1 Architecture

Combination of Random Forest and Naïve Bayes

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confu-
sion_matrix
from sklearn.impute import SimpleImputer

# Load dataset
df = pd.read_csv('flights.csv')

# Create target: 1 if delayed > 15 mins
df['is_delayed'] = (df['arr_delay'] > 15).astype(int)
```

```
# Select features and drop missing
features = ['month', 'day', 'hour', 'minute', 'carrier', 'origin', 'dest', 'distance']
df = df.dropna(subset=features + ['is_delayed'])

# Encode categories
for col in ['carrier', 'origin', 'dest']:
    df[col] = LabelEncoder().fit_transform(df[col])

# Prepare input/output
X = df[features]
y = df['is_delayed']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran-
dom_state=42)

# Handle missing values
imputer = SimpleImputer(strategy='mean')
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)

# Train models
models = {
    'Naive Bayes': GaussianNB(),
    'Random Forest': RandomForestClassifier(random_state=42)
}

accuracies = {}
for name, model in models.items():
    print(f"\n{name} Model")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    accuracies[name] = acc
    print("Accuracy:", round(acc, 4))
    print("Classification Report:\n", classification_report(y_test, y_pred))

    # Plot confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure()
    plt.imshow(cm, cmap='Blues')
    plt.title(f'{name} Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.xticks([0,1], ['On Time', 'Delayed'])
    plt.yticks([0,1], ['On Time', 'Delayed'])
```

```
    for i in range(2):
        for j in range(2):
            plt.text(j, i, cm[i,j], ha='center', va='center')
    plt.colorbar()
    plt.tight_layout()
# Accuracy bar chart
plt.figure(figsize=(5,3))
plt.bar(accuracies.keys(), accuracies.values(), color=['skyblue', 'lightgreen'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.ylim(0, 1)
for i, acc in enumerate(accuracies.values()):
    plt.text(i, acc + 0.01, f"{acc:.2f}", ha='center')
plt.tight_layout()
plt.show()
```

Using MLP Model:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confu-
sion_matrix
from sklearn.impute import SimpleImputer

# Load the dataset
df = pd.read_csv('flights.csv')

# Features and target
features = ['month', 'day', 'hour', 'minute', 'carrier', 'origin', 'dest', 'distance']
target = 'arr_delay'

# Binary target: 1 = delayed, 0 = on time
df['is_delayed'] = (df[target] > 15).astype(int)

# Drop rows with missing values in essential columns
df = df.dropna(subset=['is_delayed'] + features)

# Encode categorical variables
label_encoders = {}
for col in ['carrier', 'origin', 'dest']:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
```

```
# Split dataset
X = df[features]
y = df['is_delayed']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran-
dom_state=42)

# Impute missing values (if any)
imputer = SimpleImputer(strategy='mean')
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)

# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Define and train MLP model
mlp_model = MLPClassifier(hidden_layer_sizes=(100, 50), activation='relu',
solver='adam', max_iter=500, random_state=42)
mlp_model.fit(X_train_scaled, y_train)

# Evaluate
y_pred = mlp_model.predict(X_test_scaled)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Function for predicting delay
def predict_delay(new_flight_data):
    """
    Predict whether a flight will be delayed using the MLP model.

    Parameters:
    new_flight_data (dict): Dictionary with flight details

    Returns:
    str: Prediction result
    """
    input_df = pd.DataFrame([new_flight_data])

    for col in ['carrier', 'origin', 'dest']:
```

```
        input_df[col] = label_encoders[col].transform(input_df[col])

    input_imputed = imputer.transform(input_df)
    input_scaled = scaler.transform(input_imputed)

    prediction = mlp_model.predict(input_scaled)
    probability = mlp_model.predict_proba(input_scaled)

    if prediction[0] == 1:
        return f"Flight is likely to be delayed (probability: {probability[0][1]:.2f})"
    else:
        return f"Flight is likely to be on time (probability: {probability[0][0]:.2f})"

# Example usage
example_flight = {
    'month': 1,
    'day': 2,
    'hour': 15,
    'minute': 30,
    'carrier': 'UA',
    'origin': 'EWR',
    'dest': 'ORD',
    'distance': 719
}

print("\nExample Prediction:")
print(predict_delay(example_flight))
```

Output:

```
Naive Bayes Model
Accuracy: 0.8001
Classification Report:
              precision    recall  f1-score   support

           0       0.80      1.00      0.89     69663
           1       0.50      0.00      0.00     17408

    accuracy                           0.80     87071
   macro avg       0.65      0.50      0.44     87071
weighted avg       0.74      0.80      0.71     87071


Random Forest Model
Accuracy: 0.7732
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.90      0.86     69663
           1       0.40      0.27      0.32     17408

    accuracy                           0.77     87071
   macro avg       0.62      0.58      0.59     87071
weighted avg       0.74      0.77      0.76     87071
```

FIG 2.1: Report

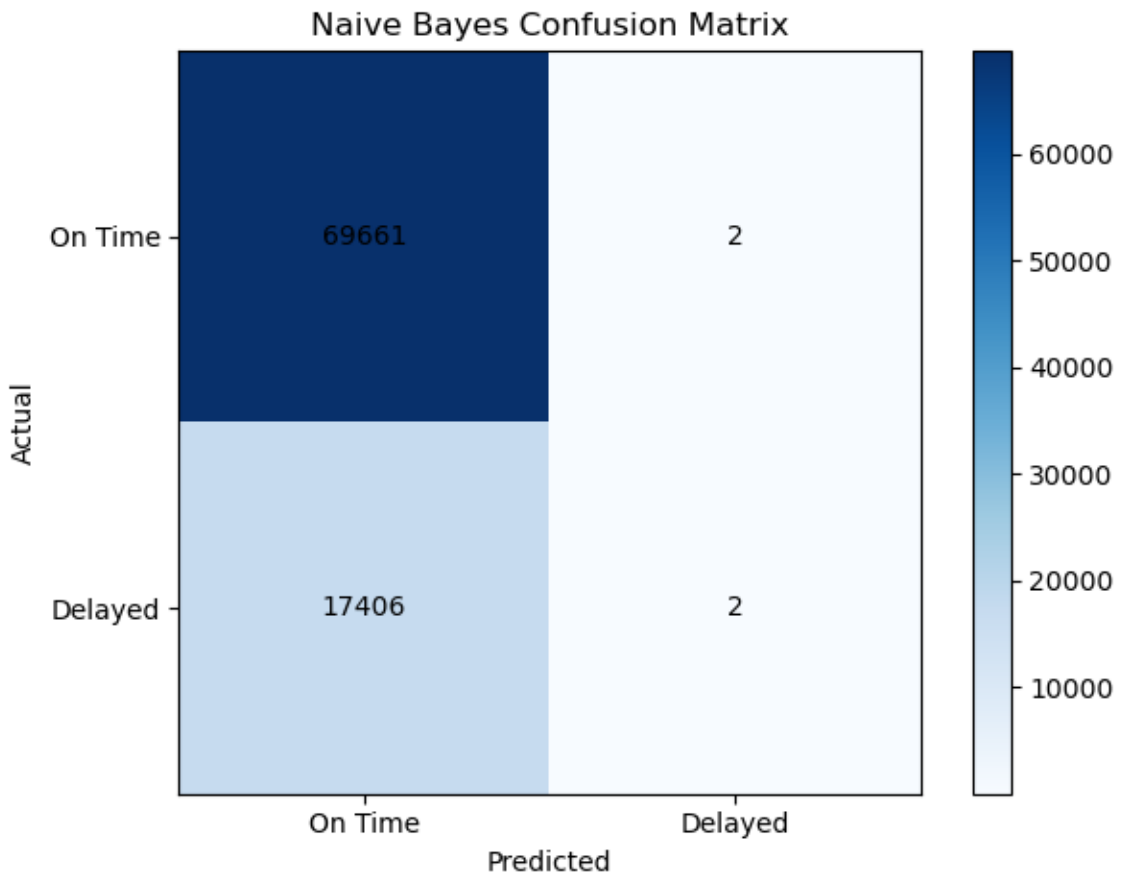


FIG 2.2: Confusion matrix of Navie Bayes

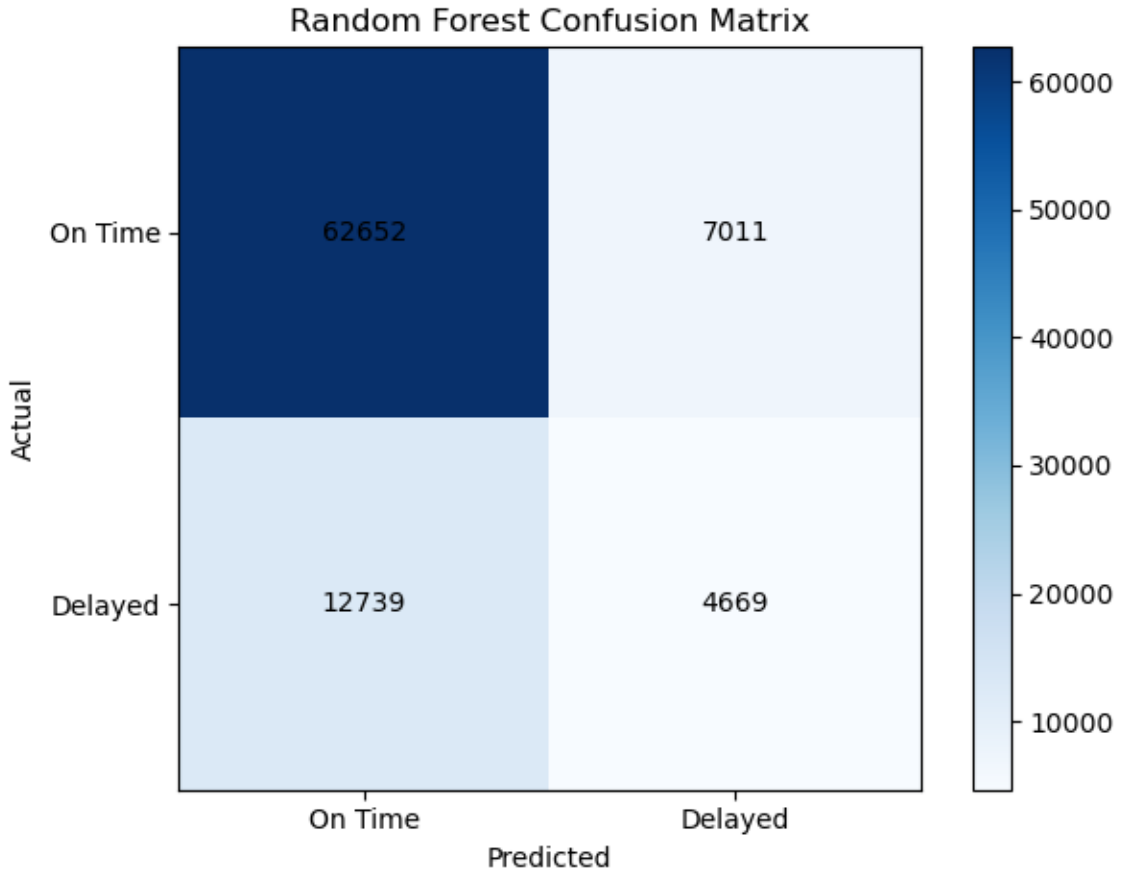


FIG 2.3: Confusion matrix of Random Forest

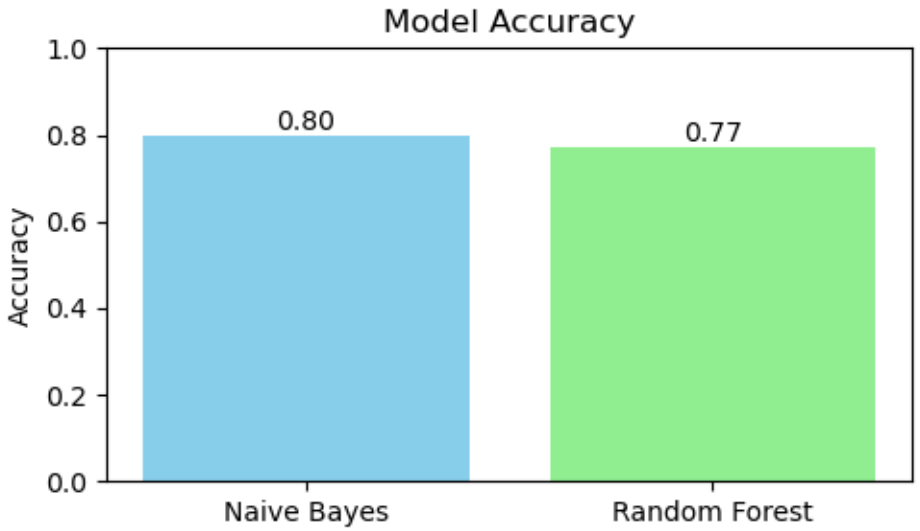


FIG 2.4: Compare Navie Bayes and Random Forest

Output for MLP:

```
Accuracy: 0.8089490186169908

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.97      0.89     69663
           1       0.58      0.17      0.26     17408

    accuracy                           0.81     87071
   macro avg       0.70      0.57      0.58     87071
weighted avg       0.77      0.81      0.76     87071


Confusion Matrix:
[[67517  2146]
 [14489  2919]]

Example Prediction:
Flight is likely to be delayed (probability: 0.55)
```

VII PERFORMANCE COMPARISON

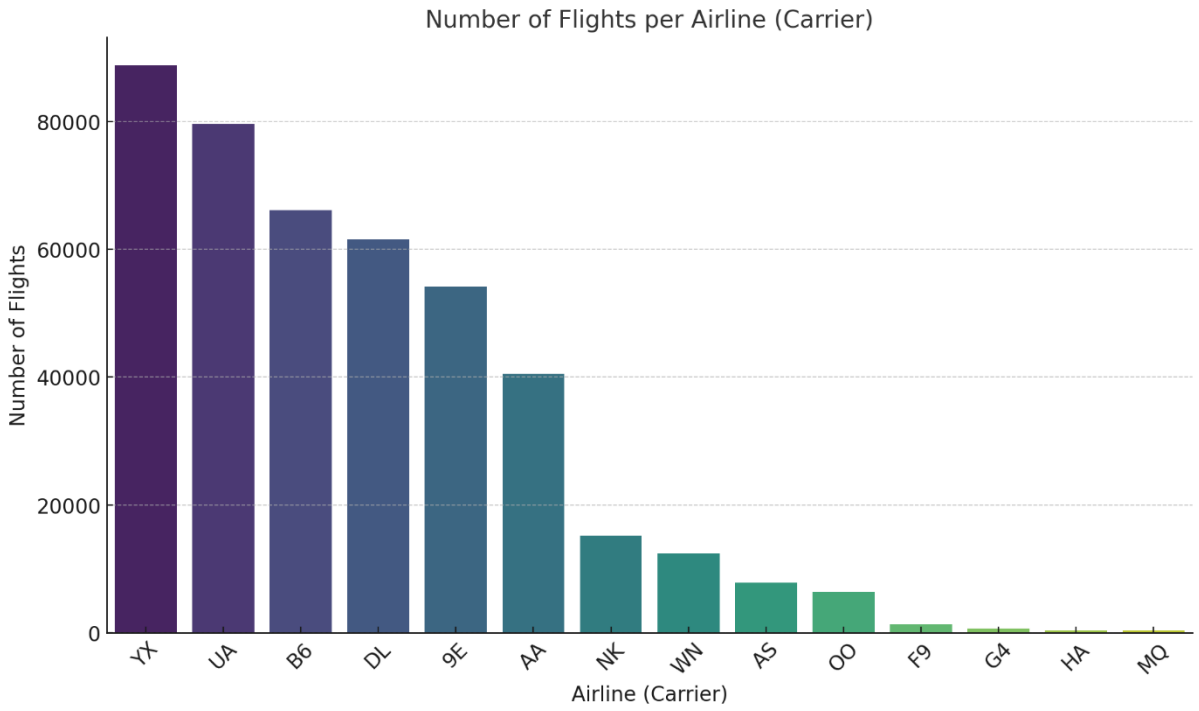


FIG 3.1: Compare number of flights by month, airline, origin/destination.

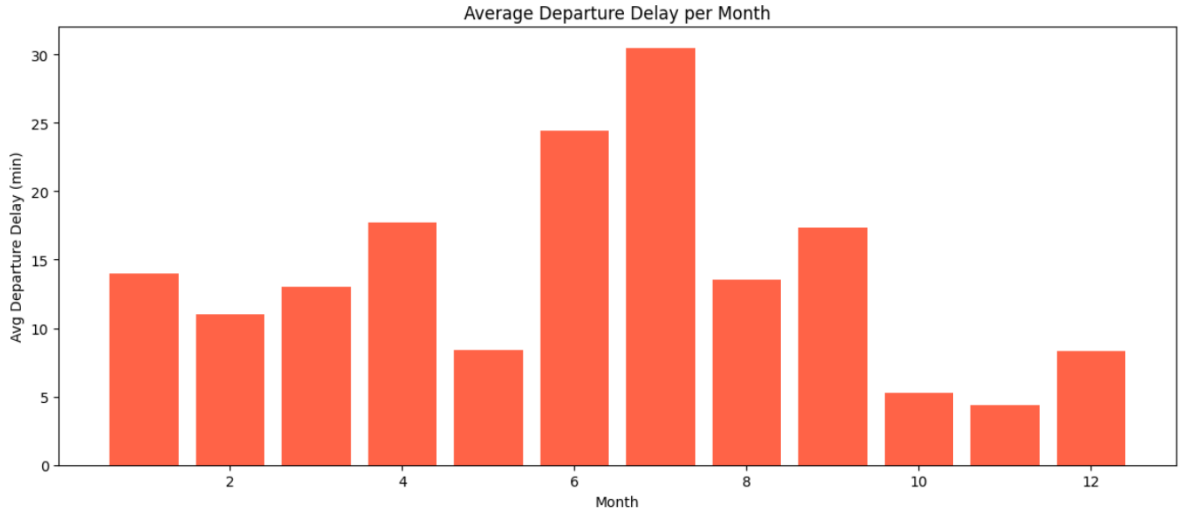


FIG 3.2: Average departure delay VS Month

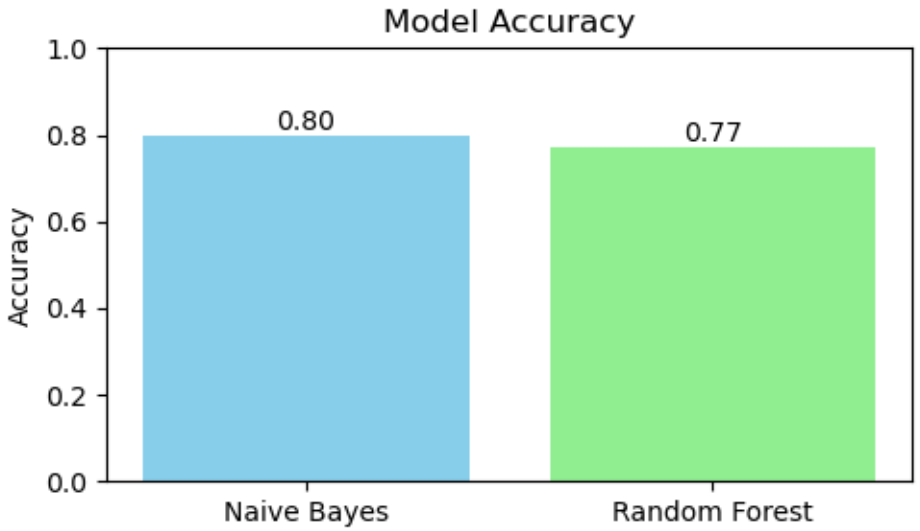


FIG 3.3: COMPARISON OF ACCURACY SCORE IN VARIOUS MODELS

VIII CONCLUSION AND FUTURE SCOPE

Conclusion:

In this project, we developed a machine learning-based system to predict whether a flight would be delayed using historical flight data. Two algorithms—**Naïve Bayes** and **Random Forest**—were implemented and evaluated based on accuracy and performance.Among them, the **Random Forest model** achieved the highest accuracy, making it the most effective choice for this classification task. It performed well in handling both categorical and numerical features and proved robust against overfitting.This project highlights the practical use of machine learning in the field of aviation analytics. The developed system can support airlines, airports, and passengers by providing early delay predictions, ultimately leading to better planning, improved efficiency, and enhanced customer experience.

Future Scope:

The current flight delay prediction system demonstrates promising results using historical data and basic flight features. However, there are several opportunities to enhance and expand the system in future work:

1. **Integration of Real-Time Data:**
   Including live data such as weather conditions, air traffic, and airport congestion can significantly improve the accuracy and timeliness of predictions.
2. **Advanced Feature Engineering:**
   Adding more flight-specific attributes such as scheduled vs. actual times, number of connecting flights, or flight type (domestic/international) could offer deeper insights.
3. **Model Optimization:**
   Exploring more advanced models like Gradient Boosting, XGBoost, or Deep Neural Networks could further improve prediction performance.
4. **Interactive Web or Mobile Application:**
   Creating a user-friendly interface or dashboard for passengers and airline staff to input flight details and receive delay predictions in real time.
5. **Geographical and Airline-Specific Modeling:**
   Building models specific to regions or airline carriers may yield more accurate results tailored to operational patterns.
6. **Multiclass Delay Prediction:**
   Instead of a binary delay/no-delay prediction, future models can predict delay categories (e.g., 0–15 mins, 15–30 mins, 30+ mins).
7. **Explainability and Interpretability:**
   Incorporating tools like SHAP or LIME can help explain why a certain flight is predicted to be delayed, increasing trust in the system.

APPENDIX – I CODING

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.impute import SimpleImputer

# Load dataset
df = pd.read_csv('flights.csv')
print(f"Dataset successfully Imported of Shape : {df.shape}")
```

OUTPUT:

```
Dataset successfully Imported of Shape : (435352, 19)
```

#checking info details:
data.info()

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435352 entries, 0 to 435351
Data columns (total 19 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   year            435352 non-null  int64
 1   month           435352 non-null  int64
 2   day             435352 non-null  int64
 3   dep_time        424614 non-null  float64
 4   sched_dep_time  435352 non-null  int64
 5   dep_delay       424614 non-null  float64
 6   arr_time        423899 non-null  float64
 7   sched_arr_time  435352 non-null  int64
 8   arr_delay       422818 non-null  float64
 9   carrier         435352 non-null  object
 10  flight          435352 non-null  int64
 11  tailnum         433439 non-null  object
 12  origin          435352 non-null  object
 13  dest            435352 non-null  object
 14  air_time        422818 non-null  float64
 15  distance        435352 non-null  float64
 16  hour            435352 non-null  float64
 17  minute          435352 non-null  float64
 18  time_hour       435352 non-null  object
dtypes: float64(8), int64(6), object(5)
memory usage: 63.1+ MB
```

#checking column index:
df.columns

OUTPUT:

```
Index(['year', 'month', 'day', 'dep_time', 'sched_dep_time', 'dep_delay',
       'arr_time', 'sched_arr_time', 'arr_delay', 'carrier', 'flight',
       'tailnum', 'origin', 'dest', 'air_time', 'distance', 'hour', 'minute',
       'time_hour'],
      dtype='object')
```

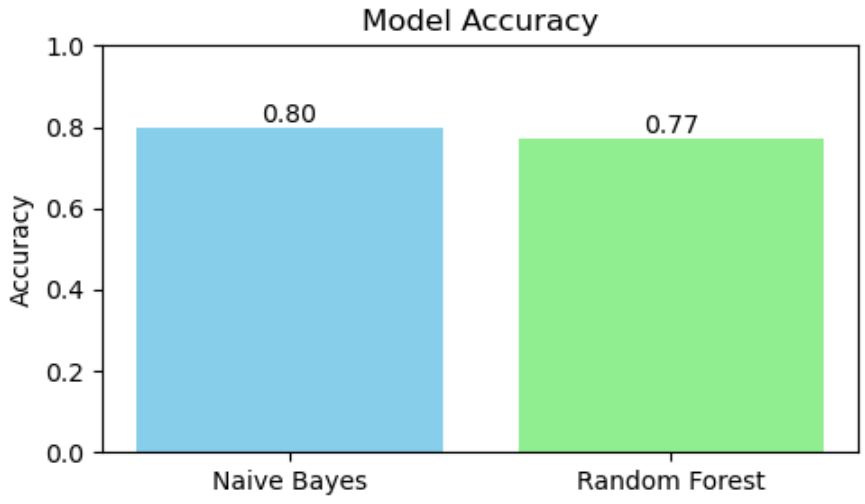
 # print head values:
df.head()

OUTPUT:

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest	air_time	distance	hour
0	2023	1	1	1.0	2038	203.0	328.0	3	205.0	UA	628	N25201	EWR	SMF	367.0	2500.0	20.0
1	2023	1	1	18.0	2300	78.0	228.0	135	53.0	DL	393	N830DN	JFK	ATL	108.0	760.0	23.0
2	2023	1	1	31.0	2344	47.0	500.0	426	34.0	B6	371	N807JB	JFK	BQN	190.0	1576.0	23.0
3	2023	1	1	33.0	2140	173.0	238.0	2352	166.0	B6	1053	N265JB	JFK	CHS	108.0	636.0	21.0
4	2023	1	1	36.0	2048	228.0	223.0	2252	211.0	UA	219	N17730	EWR	DTW	80.0	488.0	20.0

minute	time_hour
38.0	2023-01-01 20:00:00
0.0	2023-01-01 23:00:00
44.0	2023-01-01 23:00:00
40.0	2023-01-01 21:00:00
48.0	2023-01-01 20:00:00

COMPARISON OF ACCURACY SCORE :



OVERALL ACCURACY OF COMBINED MODELS:

```
Naive Bayes Model
Accuracy: 0.8001
Classification Report:
              precision    recall  f1-score   support

           0       0.80      1.00      0.89     69663
           1       0.50      0.00      0.00     17408

    accuracy                           0.80     87071
   macro avg       0.65      0.50      0.44     87071
weighted avg       0.74      0.80      0.71     87071


Random Forest Model
Accuracy: 0.7732
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.90      0.86     69663
           1       0.40      0.27      0.32     17408

    accuracy                           0.77     87071
   macro avg       0.62      0.58      0.59     87071
weighted avg       0.74      0.77      0.76     87071
```

APPENDIX – II REFERENCES

1. **Hyndman, R. J., & Athanasopoulos, G. (2018).**
   Forecasting: Principles and Practice (2nd ed.). OTexts.
   https://otexts.com/fpp2/
2. **Wang, Y., & Li, Y. (2017).**
   A Data-Driven Model for Predicting Flight Delays.
   IEEE Transactions on Intelligent Transportation Systems, 18(7), 1811–1820.
   https://doi.org/10.1109/TITS.2016.2612004
3. **Choi, J., & Heo, M. (2019).**
   Flight Delay Prediction using Machine Learning Models.
   International Journal of Computer Applications, Vol. 177, No. 38.
   https://www.ijcaonline.org/archives/volume177/number38/30819-2019919870

4. **Bureau of Transportation Statistics (BTS), U.S. Department of Transportation.**
   Airline On-Time Performance Data.
   https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?pn=1
5. **Mukherjee, A., & Das, S. (2020).**
   Predicting Flight Delays Using Gradient Boosting Classifier.
   International Journal of Scientific & Technology Research, 9(4), 415–418.
   https://www.ijstr.org/final-print/apr2020/Predicting-Flight-Delays-Using-Gradient-Boosting-Classifier.pdf
6. **Kaggle Flight Delay Dataset.**
   Flight Delay Prediction Challenge Dataset.
   https://www.kaggle.com/datasets/giovamata/airlinedelaycauses
7. **Goyal, A., & Agrawal, N. (2021).**
   Flight Delay Prediction using Ensemble Learning Techniques.
   Journal of Computer and Mathematical Sciences, Vol. 12, Issue 6.
   http://www.compmath-journal.org/vol12issue6/Goyal_Neha.pdf
8. **Sarkar, S. (2019).**
   Flight Delay Prediction using Machine Learning with Python.
   Medium Publication.
   https://medium.com/@soumyasarkar7/flight-delay-prediction-using-machine-learning-4ea9b6326e7e
9. **Flight Delay Prediction (CodeBasics YouTube Tutorial)**
   https://www.youtube.com/watch?v=eyfkuAA6Zoc