**Government Arts And Science College**

**Sankarankovil**

**Tenkasi (dt)-627756.**

April - 2023

**Flight Delay Prediction For Aviation Industry Using**

**Machine Learning in Python**

**Team ID :** NM2023TMID15981

**Team Leader :** Pavinkumar S

**Reg No :** 20201361506117

**Team Member :** Murugan V

**Reg No :** 20201361506116

**Team Member :** Sakthikannan A

**Reg No :** 20201361506119

**Team Member :** Rajan M

**Reg No :** 20201361506118

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No:** | **TITLE** | **PG.NO:** |
| 01 | Introduction | 01 |
| 02 | Problem Definition & Design Thinking | 03 |
| 03 | Result | 07 |
| 04 | Advantages & Disadvantages | 11 |
| 05 | Application | 15 |
| 06 | Conclusion | 18 |
| 07 | Future Scope | 24 |
| 08 | Appendix | 26 |

**INTRODUCTION:**

The aviation industry has a tremendous impact on global commerce and tourism. It facilitates the movement of people and goods around the world, connecting distant locations and enabling business and personal travel. However, one of the biggest challenges the industry faces is flight delays. Flight delays can cause significant disruptions and can be frustrating for passengers, resulting in dissatisfaction and potentially lost revenue for airlines.

In recent years, machine learning has emerged as a powerful tool for predicting flight delays. By analyzing vast amounts of historical data on flights, weather patterns, air traffic, and other relevant factors, machine learning algorithms can identify patterns and correlations that can help predict future delays. These predictions can be used by airlines and airports to optimize operations, reduce delays, and improve the overall passenger experience.

The ability to accurately predict flight delays has significant implications for the aviation industry. By knowing when and where delays are likely to occur, airlines can adjust their schedules and allocate resources more effectively. They can also proactively inform passengers about potential delays and offer alternative travel arrangements, which can help reduce frustration and improve customer satisfaction.

Moreover, machine learning models can help airports and airlines identify the root causes of delays, allowing them to take corrective action and minimize the likelihood of future delays. For example, if a particular airport consistently experiences delays due to congestion or weather patterns, airlines can adjust their schedules to avoid these times, or the airport can invest in infrastructure improvements to alleviate congestion.

In summary, the use of machine learning for predicting flight delays is a promising development for the aviation industry. By leveraging this technology, airlines and airports can improve their operations, reduce delays, and enhance the overall passenger experience. As the amount of data and computing power available continues to grow, machine learning models will become even more accurate and effective in predicting flight delays, enabling the industry to operate more efficiently and reliably.

**Overview of Flight Delay Predictions :**

**Data collection:**

Historical flight data, which includes information such as flight departure and arrival times, weather conditions, airline, aircraft type, and other relevant features, is collected from various sources, such as airline databases, aviation authorities, weather agencies, and other publicly available data sources.

**Data preprocessing:**

The collected data is then cleaned, transformed, and preprocess to ensure its quality and relevance for machine learning algorithms. This may involve handling missing data, normalizing numerical features, encoding categorical variables, and handling outliers, among other tasks.

**Feature engineering:**

Relevant features that may impact flight delays, such as time of day, day of week, weather conditions, and airline information, are extracted or engineered from the raw data. This step is crucial as the quality and relevance of features can significantly impact the performance of machine learning models.

**Model training:**

The preprocess data is used to train machine learning algorithms, such as decision trees, random forests, support vector machines, or deep learning models, using various techniques such as supervised learning or time-series analysis. The data is typically split into training, validation, and testing sets to evaluate model performance and prevent overfitting.

**Model evaluation:**

The trained models are evaluated using various performance metrics, such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve, to assess their effectiveness in predicting flight delays. Different models and techniques are compared to select the best-performing model.

**Model deployment:**

Once a suitable model is identified, it can be deployed in a production environment, such as an airline's operational system, to make real-time predictions on incoming flights and provide alerts or notifications to relevant stakeholders, such as airlines, airports, and travelers.

**Model monitoring and maintenance:**

The deployed model needs to be continuously monitored and updated to ensure its accuracy and relevance over time. This may involve retraining the model with new data, fine-tuning hyper parameters, and handling concept drift or data drift to maintain its predictive performance.

**Purpose in flight delay:**

**Improved operational efficiency:**

Flight delays can have significant operational and financial impacts on airlines, airports, and other stakeholders. By accurately predicting flight delays, airlines can proactively manage their resources, such as crew scheduling, gate allocation, and maintenance, to minimize disruptions and optimize operations. This can lead to improved operational efficiency, cost savings, and better customer satisfaction.

**Enhanced customer experience:**

Flight delays can cause inconvenience and frustration to travelers. By providing accurate flight delay predictions, airlines, travel agencies, or travel platforms can notify passengers in advance and offer alternative options, such as rebooking or rerouting, to minimize travel disruptions and provide a better customer experience. This can lead to increased customer loyalty and retention.

**Risk management and decision-making:**

Accurate flight delay predictions can assist airlines, airports, and other stakeholders in risk management and decision-making processes. For example, airlines can use flight delay predictions to optimize their fleet allocation, route planning, and fuel management. Airports can use these predictions to better manage their resources, such as ground handling, gate allocation, and security staffing. Other stakeholders, such as insurance companies and travel agencies, can also use flight delay predictions to assess risks and make informed decisions.

**Research and analysis:**

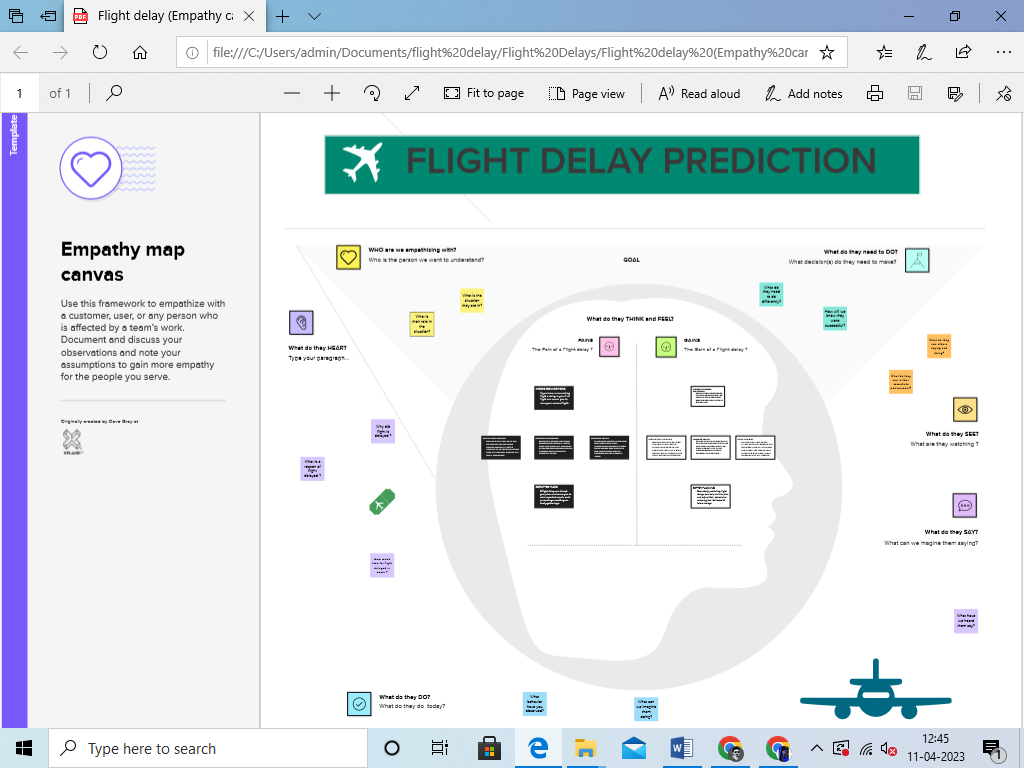
Flight delay prediction using machine learning can be used in research and analysis, such as studying the causes and patterns of flight delays, identifying trends, and understanding the impact of various factors, such as weather, air traffic, and airline operations, on flight delays. This can provide valuable insights for academic research, industry reports, policy-making, and strategic planning.

**Innovation and technological advancement:**

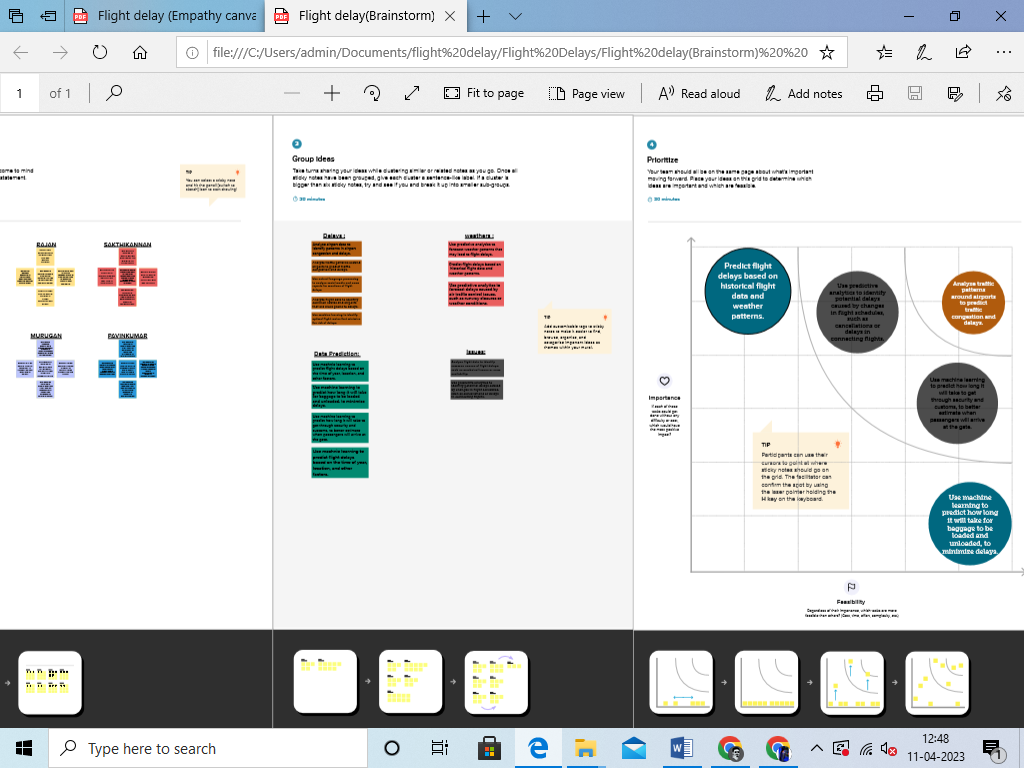
Incorporating flight delay prediction using machine learning in a project can be driven by a desire to explore and adopt cutting-edge technologies, leverage advanced analytics techniques, and promote innovation in the aviation industry. It can also serve as a proof-of-concept for demonstrating the potential of machine learning in solving real-world problems and driving technological advancements in the field of aviation.

**Problem definition & designing thinking:**

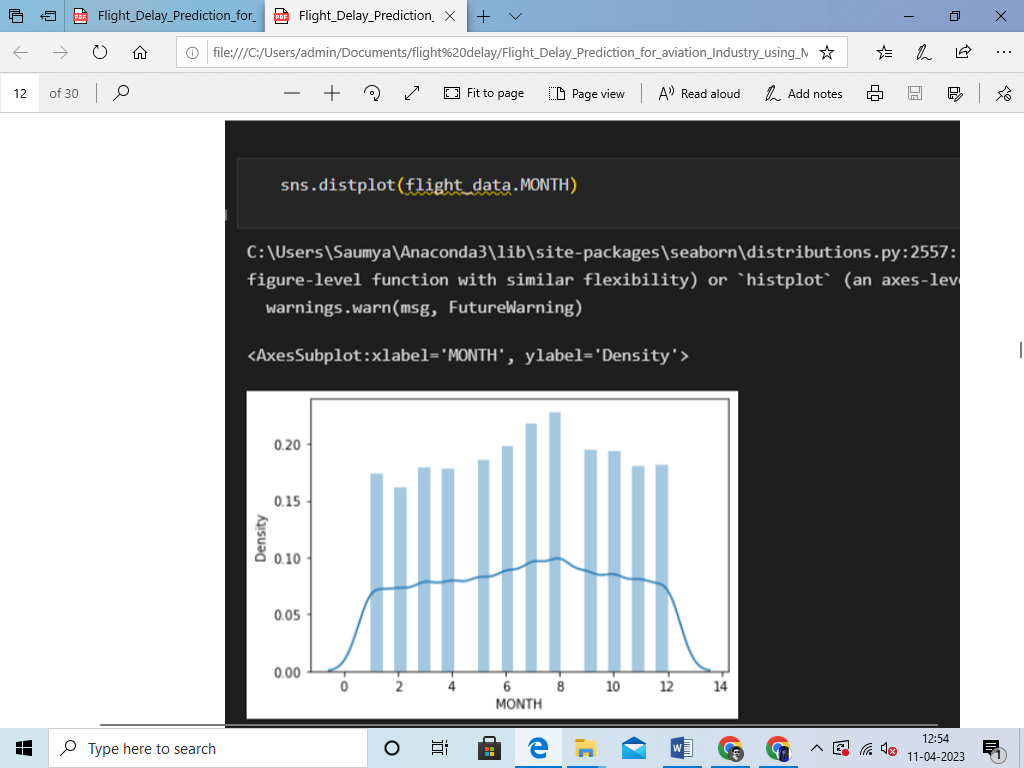
***Empathy map:***

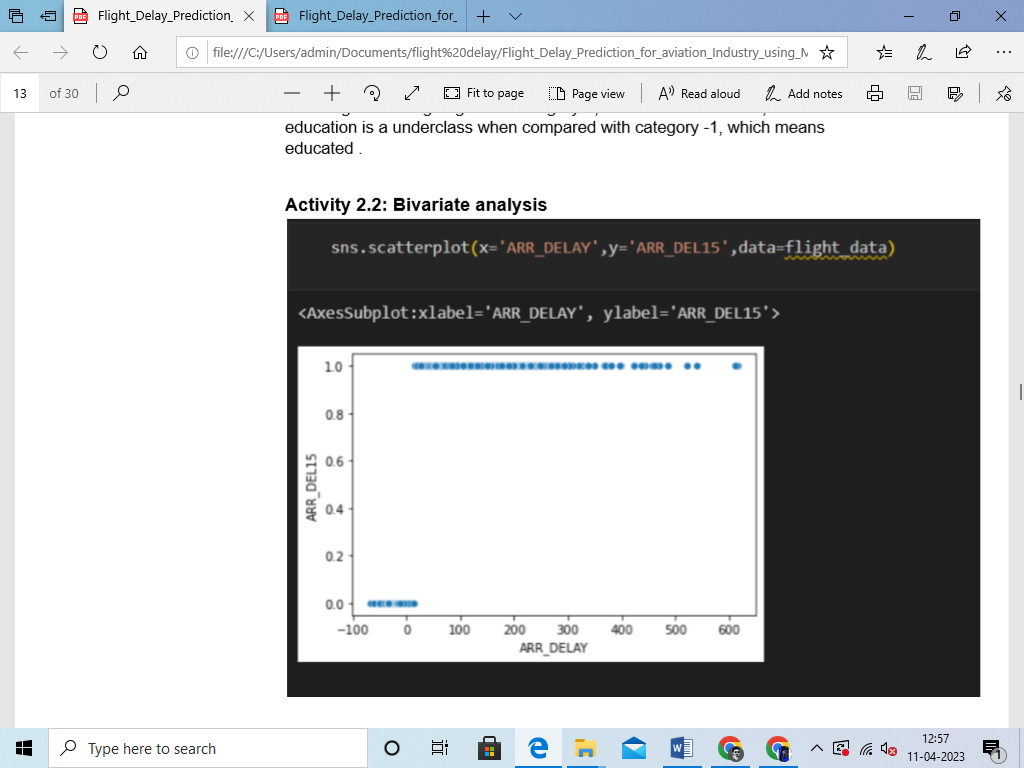


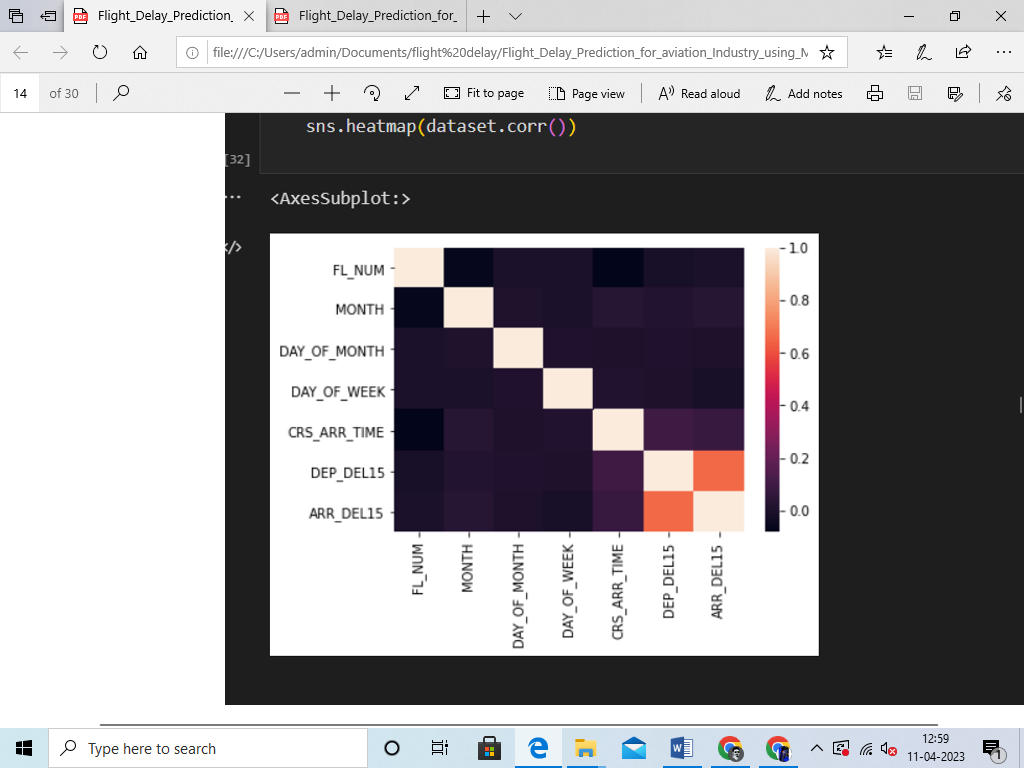
***Brainstroming Map:***



**Result:**







**Advantages:**

**Improved operational efficiency:**

Airlines, airports, and other stakeholders in the aviation industry can benefit from flight delay prediction by optimizing their operational planning. By accurately predicting flight delays, airlines and airports can proactively manage resources, such as aircraft, crew, gates, and ground handling services, to minimize disruptions and delays, leading to improved operational efficiency.

**Enhanced customer experience:**

Flight delays can result in inconvenience and frustration for travelers. By accurately predicting flight delays, airlines can provide proactive notifications to passengers, allowing them to make alternate travel arrangements or adjust their plans accordingly. This can improve the overall customer experience by reducing uncertainty and minimizing disruptions caused by unexpected flight delays.

**Cost savings:**

Flight delays can result in additional costs for airlines, such as compensation for passengers, crew expenses, and operational penalties. By accurately predicting flight delays, airlines can better manage these costs by proactively taking measures to mitigate delays, such as re-routing flights, rescheduling crew, or optimizing ground operations.

**Safety and security:**

Flight delays can have safety and security implications, especially in cases where connecting flights, crew availability, or regulatory requirements are impacted. By accurately predicting flight delays, airlines and airports can take proactive measures to ensure safety and security, such as adjusting schedules, managing crew assignments, or re-routing flights to avoid potential safety or security risks.

**Data-driven decision making:**

Flight delay prediction using machine learning enables airlines, airports, and other stakeholders to make data-driven decisions based on historical and real-time data. This can help in identifying patterns, trends, and factors that contribute to flight delays, leading to better decision making, resource allocation, and operational planning.

**Competitive advantage:**

Airlines and airports that can accurately predict flight delays and proactively manage disruptions can gain a competitive advantage by providing better customer service, reducing costs, and improving operational efficiency. This can result in increased customer loyalty, positive brand image, and improved market competitiveness.

**Improved overall performance:**

By accurately predicting flight delays and taking proactive measures to mitigate them, airlines and airports can improve their overall performance metrics, such as on-time performance (OTP), customer satisfaction, and operational efficiency. This can lead to better performance rankings, regulatory compliance, and business outcome.

**Disadvantages:**

**Data limitations:**

Accurate prediction of flight delays requires large amounts of historical data related to flights, weather conditions, air traffic, and other relevant factors. However, obtaining comprehensive and reliable data can be challenging, as it may not always be readily available, or it may be incomplete or inaccurate.

**Complexity and model interpretability:**

Machine learning models used for flight delay prediction are often complex and may involve multiple algorithms and techniques. As a result, it can be difficult to interpret the reasoning behind the predictions, making it challenging to explain the results to stakeholders, regulators, or customers, especially in the case of black-box models such as deep learning algorithms.

**Changing factors:**

Flight delays can be influenced by a wide range of factors, including weather conditions, air traffic, airport operations, and airline scheduling. However, these factors are subject to change, and new factors may emerge over time, which can impact the accuracy of the machine learning models if they are not regularly updated and adapted.

**Uncertainty and variability:**

Flight delay prediction is inherently uncertain, as it depends on various factors that are subject to variability, such as weather conditions, air traffic, and airline operations. Machine learning models may not always be able to capture this uncertainty accurately, leading to inaccurate predictions, especially in situations where there are unforeseen events or rapid changes in the operating environment.

**Regulatory and ethical considerations:**

The use of machine learning for flight delay prediction raises regulatory and ethical concerns, such as data privacy, security, and fairness. For example, the use of passenger data for prediction purposes may raise privacy concerns, and biases in the data or models may result in unfair treatment of certain groups of passengers.

**Cost and resource requirements:**

Developing and maintaining machine learning models for flight delay prediction can be resource-intensive, requiring significant investment in data collection, data storage, computing power, and expertise in machine learning. Smaller airlines or airports with limited resources may face challenges in implementing and maintaining such systems.

**Overreliance on technology:**

Overreliance on machine learning for flight delay prediction may result in decreased reliance on human expertise, experience, and decision-making. This may lead to potential risks, as human intervention may be necessary in certain situations where machine learning models may not be able to fully capture complex scenarios or unexpected events.

**Applications:**

**Operations optimization:**

Airlines, airports, and other aviation stakeholders can use flight delay prediction to optimize their operations. By accurately predicting flight delays, airlines and airports can proactively adjust schedules, allocate resources, and manage personnel to minimize disruptions and improve operational efficiency. For example, airlines can optimize crew assignments, gate allocations, and aircraft rotations, while airports can manage ground handling, baggage handling, and other operations more effectively.

**Passenger communication and service management:**

Flight delay prediction can help airlines and airports better communicate with passengers and manage their services. By providing accurate and timely information about flight delays, airlines can proactively notify passengers, manage rebookings or accommodations, and provide alternative options to minimize inconvenience and improve customer satisfaction. Passengers can be informed in advance about potential delays, allowing them to make alternate travel arrangements or adjust their plans accordingly.

M**aintenance and safety management:**

Flight delay prediction can be used in aircraft maintenance and safety management. By predicting potential delays, airlines can proactively schedule and prioritize maintenance activities, reducing the risk of unscheduled maintenance events that can cause delays. Additionally, by predicting adverse weather conditions or other safety-related factors, airlines can take appropriate measures to ensure the safety of passengers, crew, and aircraft.

**Resource planning and optimization:**

Flight delay prediction can assist airlines and airports in optimizing their resource planning. By accurately predicting flight delays, airlines and airports can better plan and allocate resources such as aircraft, crew, ground handling equipment, and gates. This can help minimize resource wastage, improve resource utilization, and reduce costs.

**Analytics and insights:**

Flight delay prediction can provide valuable insights for airlines and airports to analyze and understand the underlying causes of delays. By analyzing historical data on flight delays, weather conditions, air traffic, and other relevant factors, machine learning models can uncover patterns, trends, and correlations that can help aviation stakeholders gain insights and make data-driven decisions for operational improvements.

**Conclusion:**

In conclusion, flight delay prediction using machine learning has the potential to bring significant benefits to the aviation industry. By leveraging historical data, advanced algorithms, and predictive analytics, machine learning models can accurately forecast flight delays, allowing airlines, airports, and other stakeholders to proactively manage operations, communicate with passengers, optimize resources, and make data-driven decisions.

However, it's important to recognize that there are also limitations and challenges associated with flight delay prediction using machine learning, such as data limitations, model complexity, changing factors, uncertainty, regulatory and ethical considerations, and resource requirements. These limitations need to be carefully considered and addressed to ensure the accuracy, reliability, and ethical use of flight delay prediction models.

Despite these challenges, flight delay prediction using machine learning has already been widely adopted in the aviation industry, and its potential for improving operational efficiency, passenger experience, safety, and decision-making cannot be overstated. As technology advances and more data becomes available, machine learning models for flight delay prediction are likely to become even more sophisticated and accurate, further enhancing their applications in the aviation industry.

Overall, flight delay prediction using machine learning has proven to be a valuable tool for aviation stakeholders to mitigate disruptions, improve operations, and enhance customer satisfaction. When properly implemented, validated, and monitored, machine learning-based flight delay prediction systems can offer significant advantages in managing the complexities of air travel and contribute to more efficient and reliable air transportation systems.

**Future Scope:**

**Improved prediction accuracy:**

As more data becomes available, including real-time data on weather, air traffic, and other relevant factors, machine learning models for flight delay prediction can become even more accurate. Advancements in machine learning algorithms, such as deep learning and ensemble methods, can also lead to improved prediction performance, allowing for more precise and reliable flight delay predictions.

**Enhanced data integration and feature engineering:**

Integrating diverse data sources and leveraging advanced feature engineering techniques can further enhance the accuracy and robustness of flight delay prediction models. This can include incorporating additional data sources, such as social media, airport infrastructure data, or airline operations data, to capture more comprehensive and nuanced factors that influence flight delays.

**Real-time and dynamic prediction:**

Real-time prediction of flight delays can enable airlines and airports to respond quickly to changing conditions and minimize disruptions. Advances in machine learning techniques, such as online learning and incremental learning, can enable models to adapt and update in real-time, providing more accurate and up-to-date predictions based on the latest data.

**Explainable AI and interpretability:**

Explainable AI, which enables the understanding and interpretation of machine learning models, can provide insights into the reasons behind flight delays, helping aviation stakeholders better understand the underlying causes and take appropriate actions. Interpretable models and visualization techniques can help improve trust, transparency, and accountability in flight delay prediction systems.

**Personalized and context-aware predictions:**

Machine learning models can be tailored to individual airlines, airports, or even specific flight routes, taking into account their unique characteristics, operations, and historical data. Context-aware predictions that consider the specific circumstances of a flight, such as the time of day, season, and location, can further enhance prediction accuracy and relevance, leading to more personalized and contextually relevant insights for aviation stakeholders.

**Integration with other aviation systems:**

Flight delay prediction can be integrated with other aviation systems, such as airline operations, crew management, revenue management, and customer service, to create a more holistic and integrated approach to managing flight delays. This can lead to more coordinated and optimized operations, resulting in improved efficiency and customer satisfaction.

**Appendix :**

Source Code :

import·pandas·as·pd

import numpy as np

import pickle

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import sklearn

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.model\_selection import RandomizedSearchCV

import imblearn

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, f1\_sc

import csv

from google.colab import file

uploaded = files.upload()

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Choose Files

Saving flightdata.csv to flightdata.csv

dataset=pd.read\_csv("flightdata.csv")

dataset.head()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** | **FL\_N** |
| **0** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 13 |
| **1** 2016 | 1 | 1 | 1 | 5 | DL | N964DN | 14 |
| **2** 2016 | 1 | 1 | 1 | 5 | DL | N813DN | 15 |
| **3** 2016 | 1 | 1 | 1 | 5 | DL | N587NW | 17 |
| **4** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 18 |

5 rows × 26 columns

dataset.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 11231 entries, 0 to 11230 Data columns (total 26 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | YEAR | 11231 | non-null |  | int64 |
| 1 |  | QUARTER | 11231 | non-null |  | int64 |
| 2 |  | MONTH | 11231 | non-null |  | int64 |
| 3 |  | DAY\_OF\_MONTH | 11231 | non-null |  | int64 |
| 4 |  | DAY\_OF\_WEEK | 11231 | non-null |  | int64 |
| 5 |  | UNIQUE\_CARRIER | 11231 | non-null |  | object |
| 6 |  | TAIL\_NUM | 11231 | non-null |  | object |
| 7 |  | FL\_NUM | 11231 | non-null |  | int64 |
| 8 |  | ORIGIN\_AIRPORT\_ID | 11231 | non-null |  | int64 |
| 9 |  | ORIGIN | 11231 | non-null |  | object |
| 10 |  | DEST\_AIRPORT\_ID | 11231 | non-null |  | int64 |
| 11 |  | DEST | 11231 | non-null |  | object |
| 12 |  | CRS\_DEP\_TIME | 11231 | non-null |  | int64 |
| 13 |  | DEP\_TIME | 11124 | non-null |  | float64 |
| 14 |  | DEP\_DELAY | 11124 | non-null |  | float64 |

dataset.isnull()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** |
| **0** | False | False | False | False | False | False | False |
| **1** | False | False | False | False | False | False | False |
| **2** | False | False | False | False | False | False | False |
| **3** | False | False | False | False | False | False | False |
| **4** | False | False | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **11226** | False | False | False | False | False | False | False |
| **11227** | False | False | False | False | False | False | False |
| **11228** | False | False | False | False | False | False | False |
| **11229** | False | False | False | False | False | False | False |
| **11230** | False | False | False | False | False | False | False |

11231 rows × 26 columns

Double-click (or enter) to edit

dataset

# YEAR QUARTER MONTH DAY\_OF\_MONTH DAY\_OF\_WEEK UNIQUE\_CARRIER TAIL\_NUM

**0** 2016 1 1 1 5 DL N836DN

**1** 2016 1 1 1 5 DL N964DN

**2** 2016 1 1 1 5 DL N813DN

**3** 2016 1 1 1 5 DL N587NW

**4** 2016 1 1 1 5 DL N836DN

dataset=dataset.drop('Unnamed: 25', axis=1)

**...** ... ... ... ... ... ... ...

dataset.isnull().sum()

**11226** 2016 4 12 30 5 DL N940DL

**11227** 2016 4 12 30 5 DL N836DN

**11228** 2016 4 12 30 5 DL N583NW

**11229** 2016 4 12 30 5 DL N554NW

**11230** 2016 4 12 30 5 DL N843DN

11231 rows × 26 columns

|  |  |
| --- | --- |
| YEAR | 0 |
| QUARTER | 0 |
| MONTH | 0 |
| DAY\_OF\_MONTH | 0 |
| DAY\_OF\_WEEK | 0 |
| UNIQUE\_CARRIER | 0 |
| TAIL\_NUM | 0 |
| FL\_NUM | 0 |
| ORIGIN\_AIRPORT\_ID | 0 |
| ORIGIN | 0 |
| DEST\_AIRPORT\_ID | 0 |
| DEST | 0 |
| CRS\_DEP\_TIME | 0 |
| DEP\_TIME | 107 |
| DEP\_DELAY | 107 |
| DEP\_DEL15 | 107 |
| CRS\_ARR\_TIME | 0 |
| ARR\_TIME | 115 |
| ARR\_DELAY | 188 |
| ARR\_DEL15 | 188 |
| CANCELLED | 0 |
| DIVERTED | 0 |
| CRS\_ELAPSED\_TIME | 0 |
| ACTUAL\_ELAPSED\_TIME | 188 |
| DISTANCE | 0 |
| dtype: int64 |  |

dataset = dataset['Index']

KeyError Traceback (most recent call last)

/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get\_loc(self, key, method, tolerance)

3628 try:

-> 3629 return self.\_engine.get\_loc(casted\_key) 3630 except KeyError as err:

 4 frames

pandas/\_libs/hashtable\_class\_helper.pxi in

pandas.\_libs.hashtable.PyObjectHashTable.get\_item()

pandas/\_libs/hashtable\_class\_helper.pxi in

pandas.\_libs.hashtable.PyObjectHashTable.get\_item()

KeyError: 'Index'

dataset.isnull().sum()

The above exception was the direct cause of the following exception:

KeyError Traceback (most recent call last)

/usr/local/lib/python3.9/dist-packages/pandas/core/indexes/base.py in get\_loc(self, key, method, tolerance)

3629 return self.\_engine.get\_loc(casted\_key) 3630 except KeyError as err:

-> 3631 raise KeyError(key) from err 3632 except TypeError:

3633 # If we have a listlike key, \_check\_indexing\_error will

raise

|  |  |
| --- | --- |
| YEAR | 0 |
| QUARTER | 0 |
| MONTH | 0 |
| DAY\_OF\_MONTH | 0 |
| DAY\_OF\_WEEK | 0 |
| UNIQUE\_CARRIER | 0 |
| TAIL\_NUM | 0 |
| FL\_NUM | 0 |
| ORIGIN\_AIRPORT\_ID | 0 |
| ORIGIN | 0 |
| DEST\_AIRPORT\_ID | 0 |
| DEST | 0 |
| CRS\_DEP\_TIME | 0 |
| DEP\_TIME | 107 |
| DEP\_DELAY | 107 |
| DEP\_DEL15 | 107 |
| CRS\_ARR\_TIME | 0 |
| ARR\_TIME | 115 |
| ARR\_DELAY | 188 |
| ARR\_DEL15 | 188 |
| CANCELLED | 0 |
| DIVERTED | 0 |
| CRS\_ELAPSED\_TIME | 0 |
| ACTUAL\_ELAPSED\_TIME | 188 |
| DISTANCE  dtype: int64 | 0 |

dataset[dataset.isnull().any(axis=1)].head(10)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** | **FL** |
| **177** | 2016 | 1 | 1 | 9 | 6 | DL | N3743H |  |
| **179** | 2016 | 1 | 1 | 10 | 7 | DL | N924DN |  |
| **184** | 2016 | 1 | 1 | 10 | 7 | DL | N922DX |  |
| **210** | 2016 | 1 | 1 | 10 | 7 | DL | N951DN |  |
| **478** | 2016 | 1 | 1 | 22 | 5 | DL | N387DA |  |
| **481** | 2016 | 1 | 1 | 22 | 5 | DL | N960AT |  |
| **491** | 2016 | 1 | 1 | 22 | 5 | DL | N972AT |  |

dataset['DEP\_DEL15'].mode()

**499** 2016 1 1 23 6 DL N321NB

**500** 2016 1 1 23 6 DL N948AT

**501** 2016 1 1 23 6 DL N712TW

0 0.0

Name: DEP\_DEL15, dtype: float64

10 rows × 25 columns

dataset= dataset.fillna({'ARR\_DEL15': 1})

dataset= dataset.fillna({'DEP\_DEL15': 0})

dataset.iloc[177:185]

# YEAR QUARTER MONTH DAY\_OF\_MONTH DAY\_OF\_WEEK UNIQUE\_CARRIER TAIL\_NUM FL

**177** 2016 1 1 9 6 DL N3743H

**178** 2016 1 1 9 6 DL N975AT

**179** 2016 1 1 10 7 DL N924DN

**180** 2016 1 1 10 7 DL N671DN

**181** 2016 1 1 10 7 DL N319NB

**182** 2016 1 1 10 7 DL N587NW

**183** 2016 1 1 10 7 DL N813DN

**184** 2016 1 1 10 7 DL N922DX

8 rows × 25 columns

import math

for index, row in dataset.iterrows():

dataset.loc[index, 'CRS\_ARR\_TIME']= math.floor(row['CRS\_ARR\_TIME']/100)

dataset.head()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** | **FL\_N** |
| **0** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 13 |
| **1** 2016 | 1 | 1 | 1 | 5 | DL | N964DN | 14 |
| **2** 2016 | 1 | 1 | 1 | 5 | DL | N813DN | 15 |
| **3** 2016 | 1 | 1 | 1 | 5 | DL | N587NW | 17 |
| **4** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 18 |

5 rows × 25 columns

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

dataset['DEST']=le.fit\_transform(dataset['DEST'])

dataset['ORIGIN']=le.fit\_transform(dataset['ORIGIN'])

dataset.head()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** | **FL\_N** |
| **0** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 13 |
| **1** 2016 | 1 | 1 | 1 | 5 | DL | N964DN | 14 |
| **2** 2016 | 1 | 1 | 1 | 5 | DL | N813DN | 15 |
| **3** 2016 | 1 | 1 | 1 | 5 | DL | N587NW | 17 |
| **4** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 18 |

5 rows × 25 columns

dataset['ORIGIN'].unique()

array([0, 1, 4, 3, 2])

dataset= pd.get\_dummies(dataset, columns=['ORIGIN','DEST'])

dataset.head()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **YEAR** | **QUARTER** | **MONTH** | **DAY\_OF\_MONTH** | **DAY\_OF\_WEEK** | **UNIQUE\_CARRIER** | **TAIL\_NUM** | **FL\_N** |
| **0** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 13 |
| **1** 2016 | 1 | 1 | 1 | 5 | DL | N964DN | 14 |
| **2** 2016 | 1 | 1 | 1 | 5 | DL | N813DN | 15 |
| **3** 2016 | 1 | 1 | 1 | 5 | DL | N587NW | 17 |
| **4** 2016 | 1 | 1 | 1 | 5 | DL | N836DN | 18 |

5 rows × 33 columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x= dataset.iloc[:,  y= dataset.iloc[:,  x | 0:8].values  8:9].values | | |  | | | |
| array([[2016, | 1, 1, ..., | | | 'DL', | 'N836DN', | 1399], | |
| [2016, | 1, 1, ..., | | | 'DL', | 'N964DN', | 1476], | |
| [2016,  ..., | 1, 1, ..., | | | 'DL', | 'N813DN', | 1597], | |
| [2016, | 4, | 12, | ..., | 'DL', | 'N583NW', | 1823], |  |
| [2016, | 4, | 12, | ..., | 'DL', | 'N554NW', | 1901], |  |
| [2016, | 4, | 12, | ..., | 'DL', | 'N843DN', | 2005]], | dtype=object) |

from sklearn.preprocessing import OneHotEncoder

oh = OneHotEncoder()

z=oh.fit\_transform(x[:,4:5]).toarray()

t=oh.fit\_transform(x[:,5:6]).toarray()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| z |  | | | | | | |
|  | array([[0., | 0., | 0., | ..., | 1., | 0., | 0.], |
|  | [0., | 0., | 0., | ..., | 1., | 0., | 0.], |
|  | [0., | 0., | 0., | ..., | 1., | 0., | 0.], |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ...,  [0., | 0., | 0., | ..., | 1., | 0., | 0.], |
| [0., | 0., | 0., | ..., | 1., | 0., | 0.], |
| [0., | 0., | 0., | ..., | 1., | 0., | 0.]]) |
| t |  |  | | | | | |
|  | array([[1.], |
|  | [1.], |
|  | [1.], |

...,

[1.],

[1.],

[1.]])

x=np.delete(x,[4,5],axis=1)

flight\_data.describe()

sns.distplot(flight\_data.MONTH)

sns.scatterplot(x='ARR\_DELAY',y='ARR\_DEL15',data=flight\_data)

sns.xatplot(x="ARR\_DEL15",y="ARR\_DELAY",kind='bar',data=flight\_data)

sns.heatmap(dataset.corr())

dataset=pd.get\_dummies(dataset, columns=['ORIGIN', 'DEST'])

dataset.head()

x=dataset.iloc[:, 0:8].values

y=dataset.iloc[:, 8:9].values

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=0)

from sklearn.model\_selection import train\_test\_split

train\_x,test\_x,train\_y,test\_y = train\_test\_split(dataset.drop('ARR\_DEL15', axis=1), df['AR

x\_test.shape

x\_train.shape

y\_test.shape

y\_train.shape

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(random\_staste=0)

classifier.fit(x\_train,y\_train)

decisiontree = classifier.predict(x\_test)

decisiontree

from sklearn.metrics import accuracy\_score

desacc = accuracy\_score(y\_test,decisiontree)

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n\_estimators=10,criterion='entropy')

rfc.fit(x\_train,y\_train)

y\_predict = rfc.predict(x\_test)

import tensorflow

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

classification = Sequential()

classification.add(Dense(30,activation='relu'))

classification.add(Dense(128,activation='relu'))

classification.add(Dense(64,activation='relu'))

classification.add(Dense(32,activation='relu'))

classification.add(Dense(1,activation='sigmod'))

classification.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy'])

classification.fit(x\_train,y\_train,betch\_size=4,validation\_split=0.2,epochs=100)

y\_pred=classifier.predict([[129,99,1,0,0,1,0,1,1,1,0,1,1,1,1,1]])

print(y\_pred)

(y\_pred)

y\_pred= rfc.predict([[129,99,1,0,0,1,0,1,1,1,0,1,1,1,1,1]])

print(y\_pred)

(y\_pred)

classification.save('flight.h5')

y\_pred = classification.predict(x\_test)

(y\_pred)

(y\_pred) = (y\_pred > 0.5)

(y\_pred)

def predict\_exit(sample\_value):

sample\_value=np.array(sample\_value)

sample\_value=sample\_value.reshape(1,-1)

sample\_value=sc.transform(sample\_value)

return classifier.predict(sample\_value)

test=classification.predict([[1,1,121.000000,36.0,0,0,1,0,1,1,1,1,1,1,1,1]])

if test==1:

print('Prediction: Chance of delay')

else:

print('Prediction: No chance of delay.')

from sklearn import model\_selection

from sklearn.neural\_network import MLPClassifier

dfs=[]

models=[

('RF',RandomForestClassifier()),

('DecisionTree',DecisionTreeClassifier()), ('ANN',MLPClassifier())

]

results=[]

names=[]

scoring=['accuracy','precision\_weighted','recall\_weighted','f1\_weighter','roc\_auc']

target\_names=['no delay','delay']

for name, model in models:

kfold=model\_selection.KFold(n\_splits=5,shuffle=True,random\_state=90210)

cv\_results=model\_selection.cross\_validate(model,x\_train,y\_train,cv=kfold,scoring=scoring clf=model.fit(x\_train,y\_train)

y\_pred = clf.predict(x\_test) print(name)

print(classification\_report(y\_test,y\_pred,target\_names=targernames)) results.appenf(cv\_results)

names.append(name)

this\_df=pd.DataFrame(cv\_results) this\_df['model']= name

dfs.append(this\_df)

final=pd.concat(dfs,ignore\_index=True)

return final

print('Training accuracy:',accuracy\_score(y\_train,y\_predict\_train))

print('Testing accuracy:',accuracy\_score(y\_test,y\_predict))

from sklearn.metrics import confusion\_metrix cm= confusion\_metrix(y\_test,y\_predict)

cm

from sklearn.metrics import accuracy\_score desacc=accuracy\_score(y\_test,decisiontree)

desacc

from sklearn.metrics import confusion\_matrix cm=confusion\_matrix(y\_test,decisiontree)

cm

from sklearn.metrics import accuray\_score,classification\_report score=accuracy\_score(y\_pred,y\_test)

print('The accuracy for ANN model is: {}%'.forma(score\*100))

from sklearn.metrics import confusion\_matrix cm=confusion\_matrix(y\_test,y\_pred)

cm

parameters={

'n\_estimators':[1,20,30,55,68,74,90,120,155],

'criterion':['gini','entropy'],

'max\_features':["auto","sqrt","log2"],

'max\_depth':[2,5,8,10],'veerbose':[1,2,3,4,6,8,9,10]

}

RCV=RandomizedSearchCV(estimator=rf,param\_distributions=parameters,cv=10,n\_iter=4)

RCV.fit(x\_train,y\_train)

bt\_params=RCV.best\_params\_ bt\_score=RCV.best\_score\_

bt\_params

bt\_score

model= RandomForestClassifier(verbose=10,n\_estimators=120,max\_features='log2',max\_depth=10 RCV.fit(x\_train,y\_train)

y\_predict\_rf=RCV.predict(x\_test)

RFC=accuracy\_score(y\_test,y\_predict\_rd) RFC

import pickle

pickle.dump(RCV,open('flight.pkl','wb'))