Kamarajar Government Arts and Science College,Surandai.

Tenkasi (dt)-627859

April - 2023

Flight Delay Prediction For Aviation Industry Using

Machine Learning in Python

Team ID : NM2023TMID15981

Team Leader : I.Mohammed Aneesh

Reg No : 20201061506227

Team Member : K.Manikandan

Reg No : 20201061506220

Team Member : S.Manikandan

Reg No : 20201061506221

Team Member : T.Prakash

Reg No : 20201061506236

Team Member **:** E.Sundar

Reg No **:** 20201061506246

INDEX

1.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:

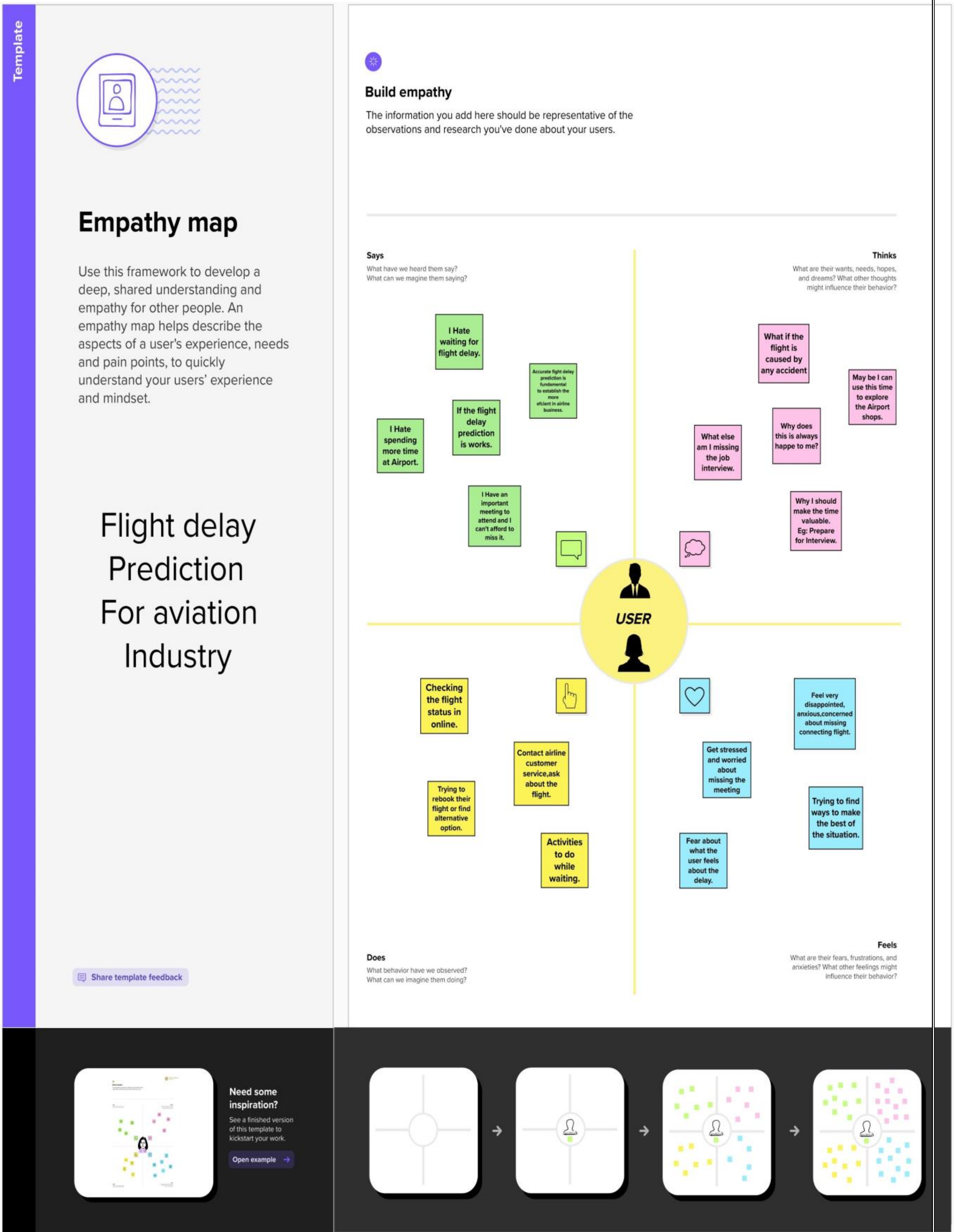
Flight delay prediction using machine learning can be used in research and analysis, such as studying the causes and patterns of flight delays, identifying trends, and understanding the impact of various factors, such as weather, air traffic, and airline operations, on flight delays. This can provide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

Innovation and technological advancement:

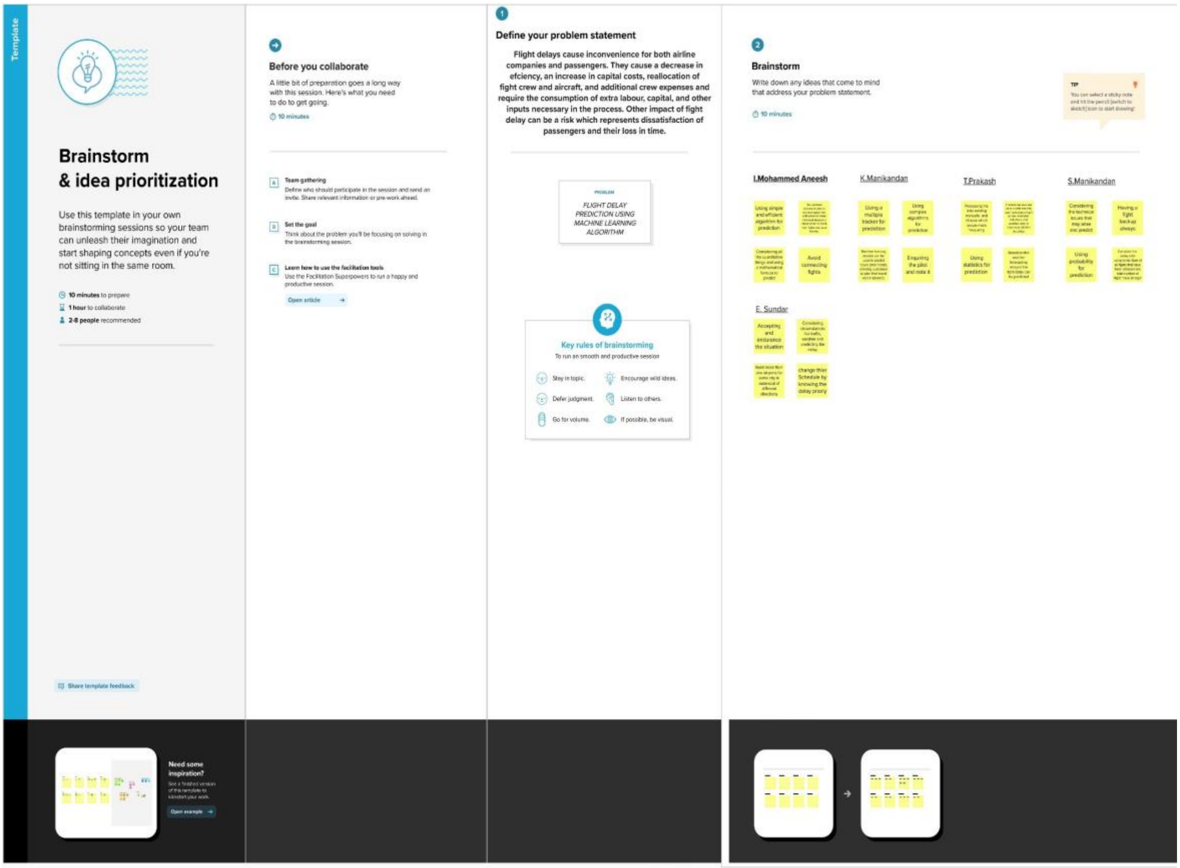
Incorporating flight delay prediction using machine learning in a project can be driven by a desire to explore and adopt cutting-edge technologies, leverage advanced analytics techniques, and promote innovation in the aviation industry. It can also serve as a proof-of-concept for demonstrating the potential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

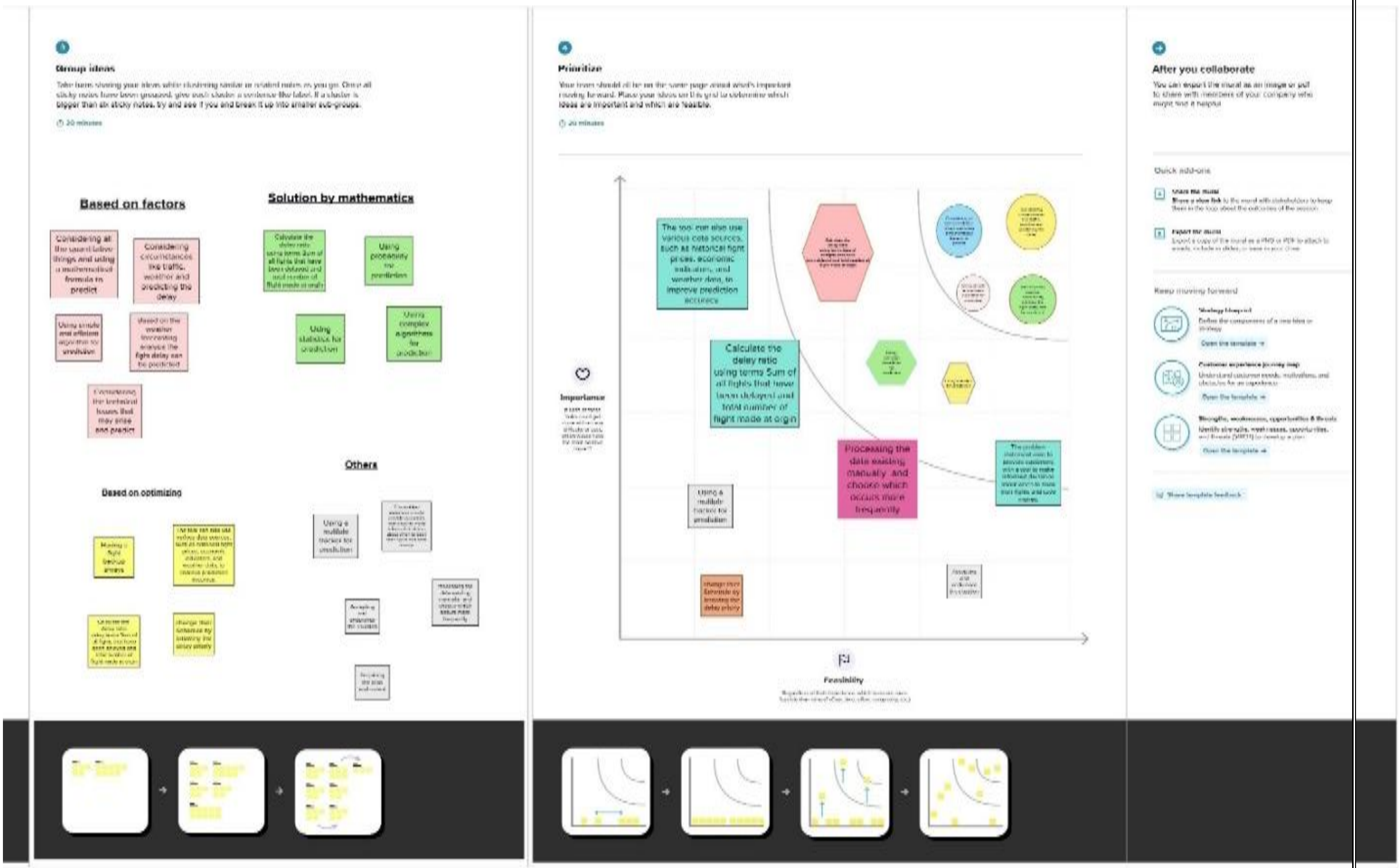
2.Problem definition & designing thinking:-

EMPATHYMAP:-

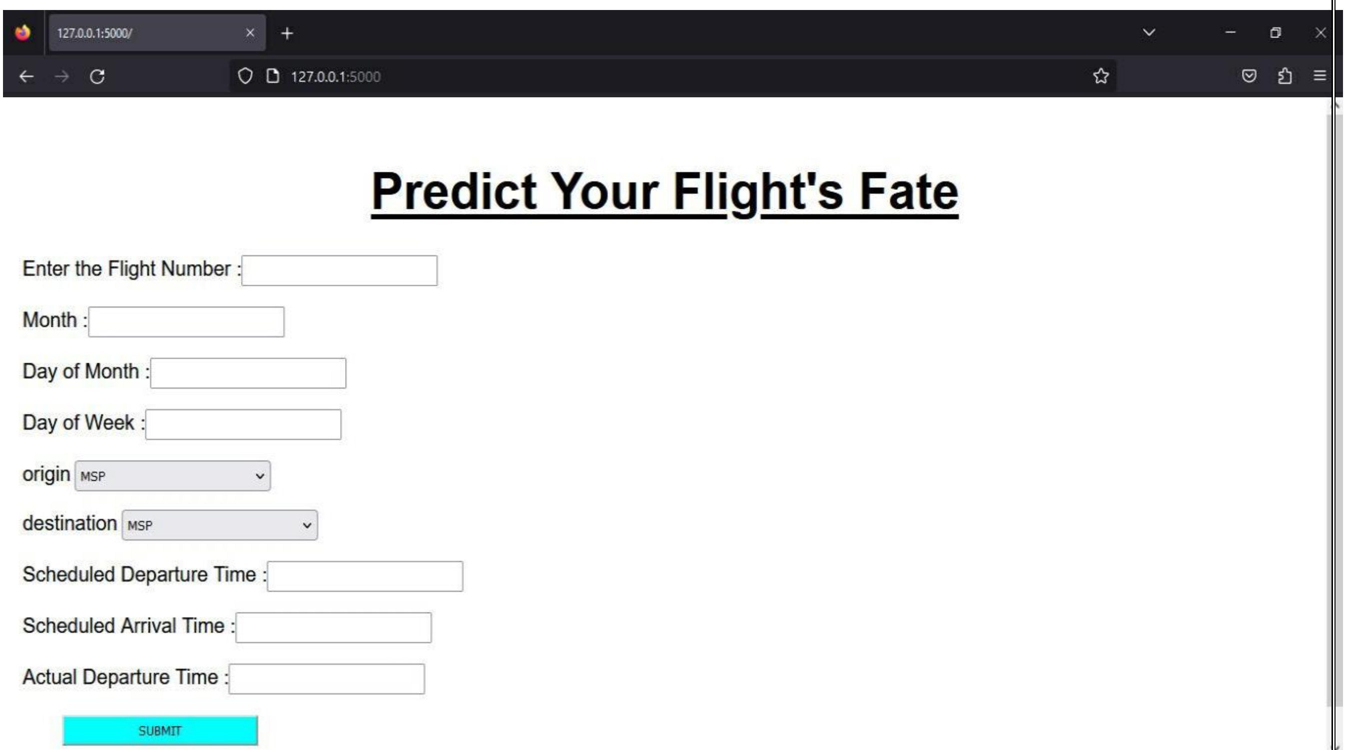

Template
Empathy map
Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.
Flight delay Prediction For aviation Industry
Share template feedback
Build empathy
The information you add here should be representative of the observations and research you've done about your users.
Says
What have we heard them say?
What can we magine them saying?
I Hate waiting for flight delay.
Accurate fight delay prediction is fundemental to establish the more efcient in airline business.
If the flight delay prediction is works.
I Hate spending more time at Airport.
I Have an important meeting to attend and I can't afford to miss it.
Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?
What if the flight is caused by any accident
May be I can use this time to explore the Airport shops.
Why does this is always happe to me?
What else am I missing the job interview.
Why I should make the time valuable. Eg: Prepare for Interview.
USER
Checking the flight status in online.
Contact airline customer service,ask about the flight.
Trying to rebook their flight or find alternative option.
Activities to do while waiting.
Feel very disappointed, anxious,concerned about missing connecting flight.
Get stressed and worried about missing the meeting
Trying to find ways to make the best of the situation.
Fear about what the user feels about the delay.
Does
What behavior have we observed?
What can we imagine them doing?
Feels
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?
Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

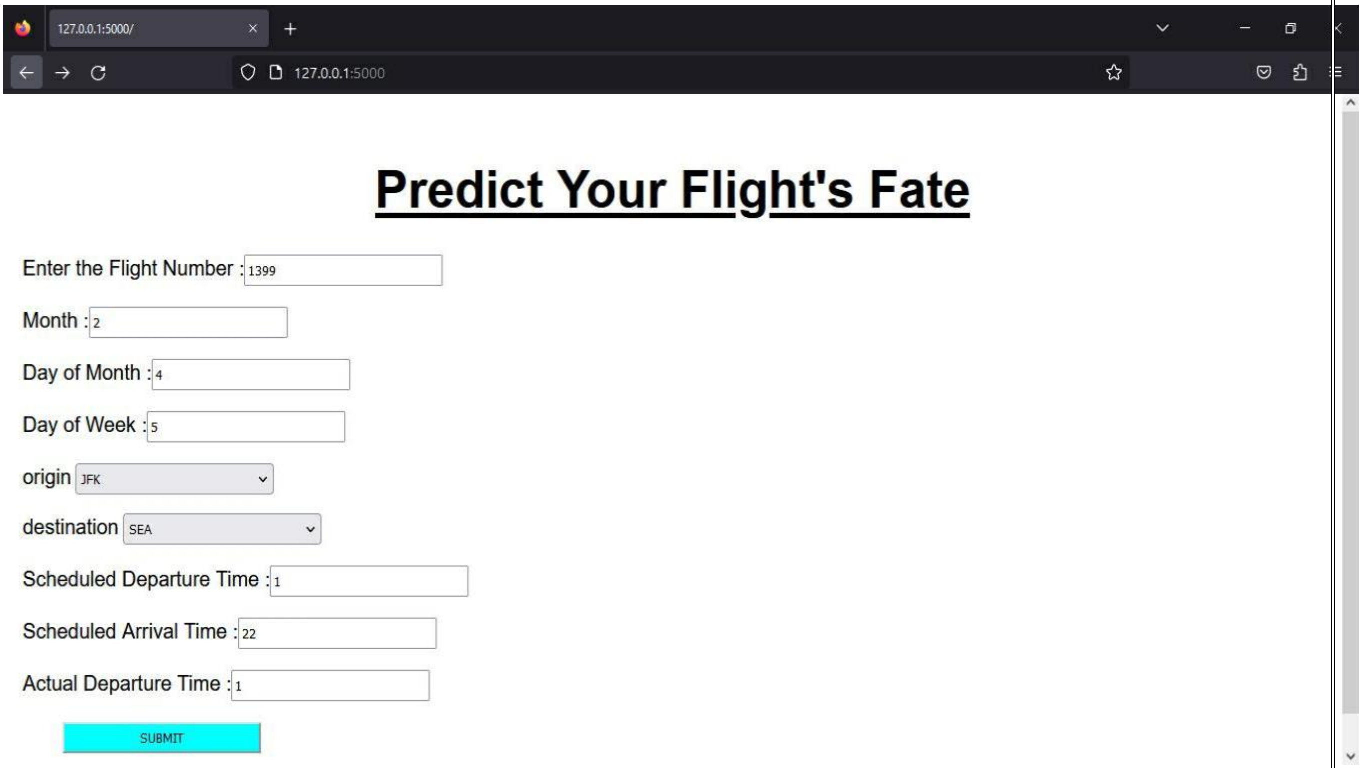
Brainstorming and idea prioritization :-

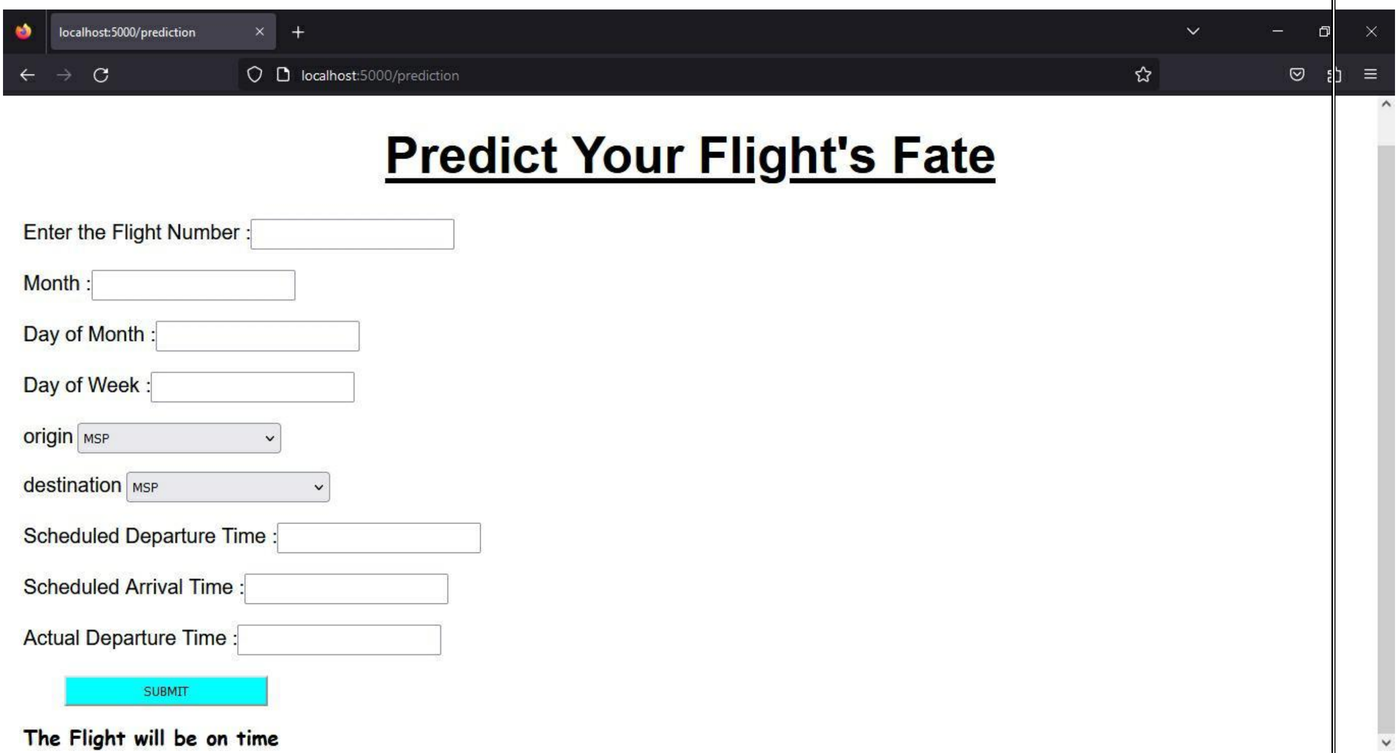


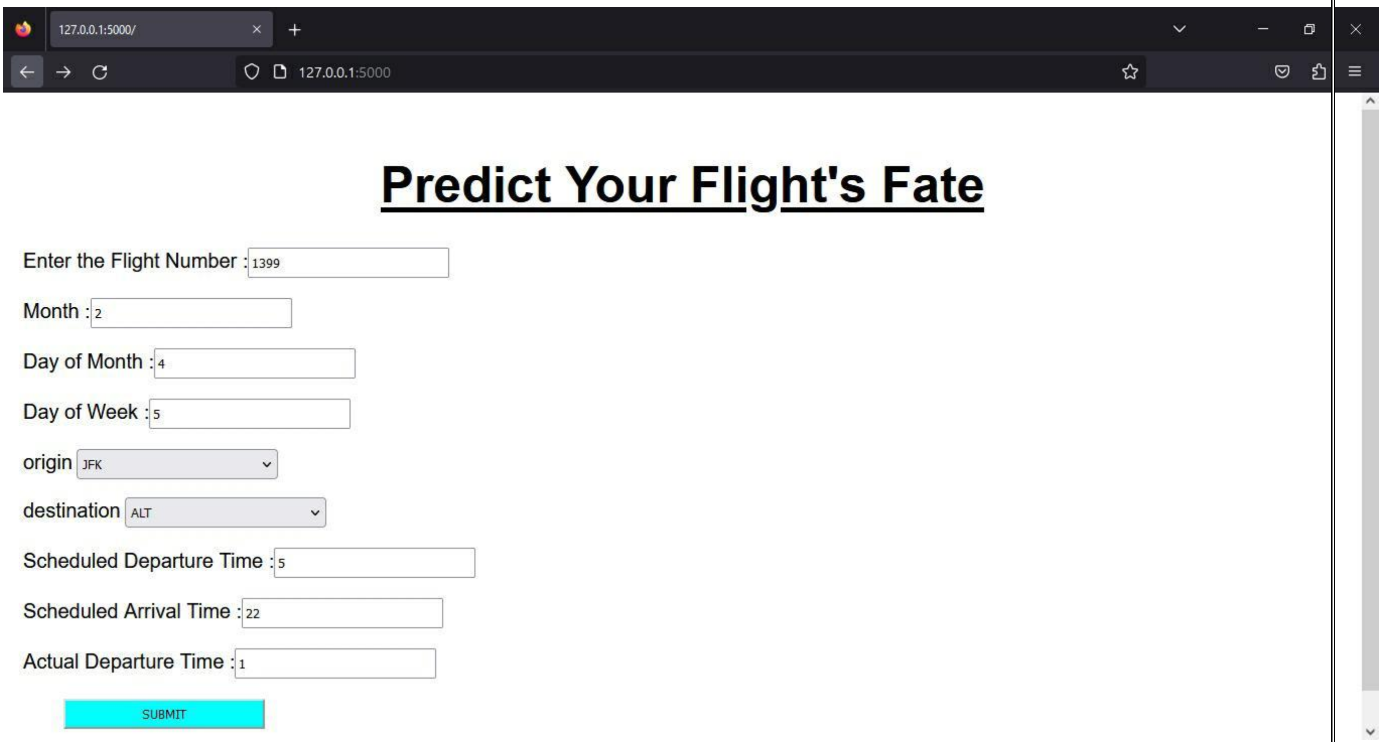

Group ideas
Prioritize
After you collaborate
Based on factors
Solution by mathematics
Others
Based on optimizing
Importance
Feasibility
The tool can also use various data sources, such as historical flight prices, economic indicators, and weather data, to improve prediction accuracy
Calculate the delay ratio using terms Sum of all flights that have been delayed and total number of flight made at origin
Processing the data existing manually and choose which occurs more frequently

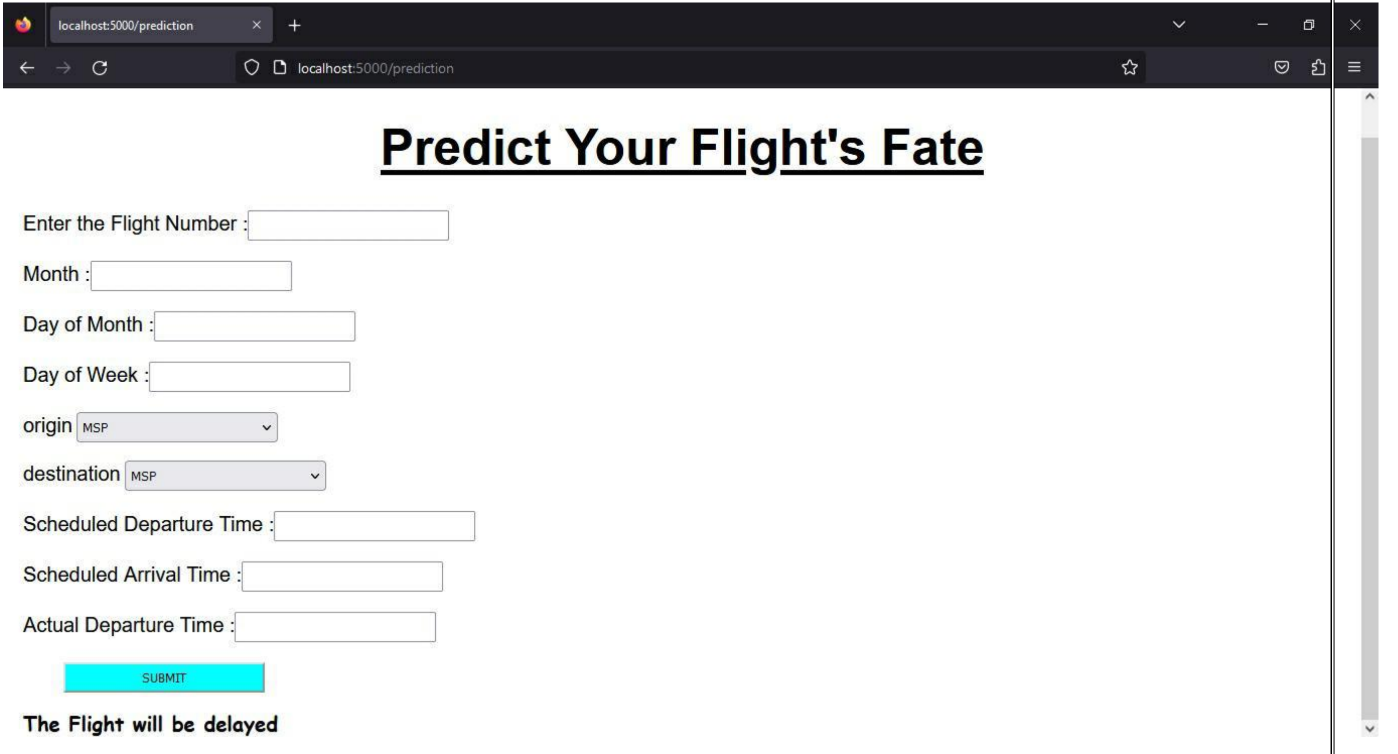
3.Result:-


127.0.0.1:5000/
127.0.0.1:5000
Predict Your Flight's Fate
Enter the Flight Number :
Month :
Day of Month :
Day of Week :
origin MSP
destination MSP
Scheduled Departure Time :
Scheduled Arrival Time :
Actual Departure Time :
SUBMIT


127.0.0.1:5000/
127.0.0.1:5000
Predict Your Flight's Fate
Enter the Flight Number : 1399
Month : 2
Day of Month : 4
Day of Week : 5
origin JFK
destination SEA
Scheduled Departure Time : 1
Scheduled Arrival Time : 22
Actual Departure Time : 1
SUBMIT


localhost:5000/prediction
localhost:5000/prediction
Predict Your Flight's Fate
Enter the Flight Number :
Month :
Day of Month :
Day of Week :
origin MSP
destination MSP
Scheduled Departure Time :
Scheduled Arrival Time :
Actual Departure Time :
SUBMIT
The Flight will be on time


127.0.0.1:5000/
127.0.0.1:5000
Predict Your Flight's Fate
Enter the Flight Number : 1399
Month : 2
Day of Month : 4
Day of Week : 5
origin JFK
destination ALT
Scheduled Departure Time : 5
Scheduled Arrival Time : 22
Actual Departure Time : 1
SUBMIT



4.Advantages and disadvantages :-

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.

5.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.

Maintenance 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.

6.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.

7.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:

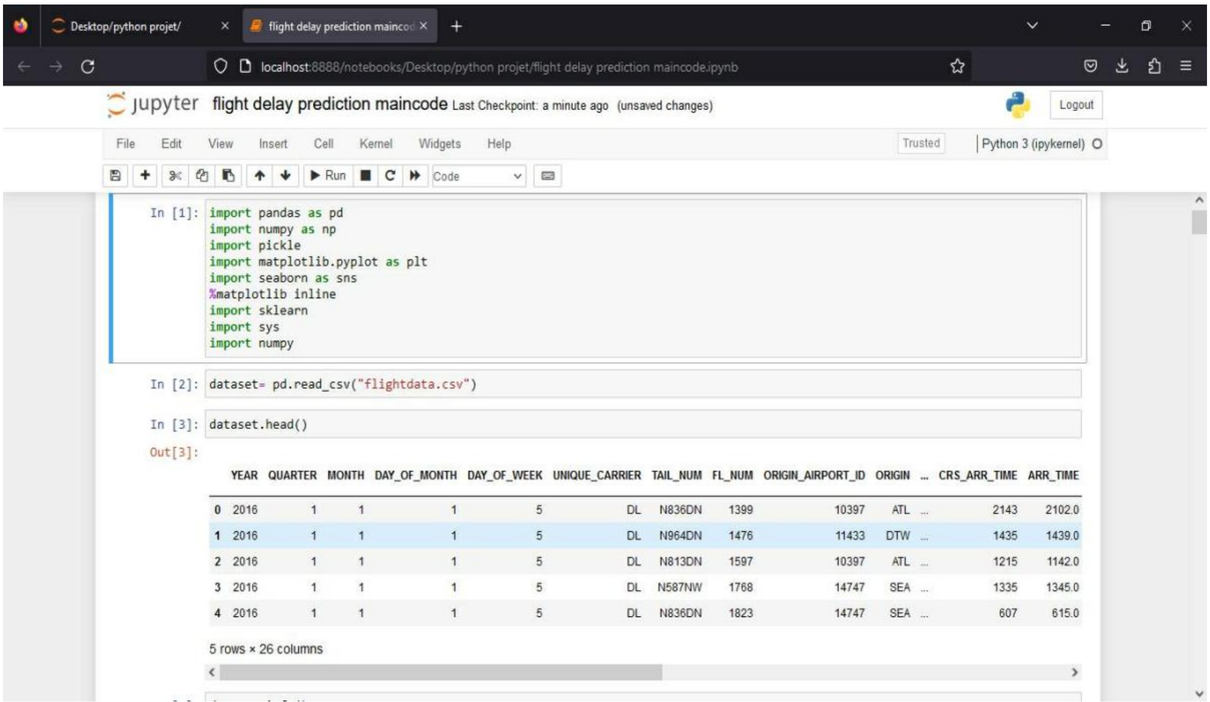
Machine learning models can be tailored to individual airlines, airports, or even specific flight routes, taking into account their unique characteristics, operations, and historical data. Context-aware predictions that consider the specific circumstances of a flight, such as the time of day, season, 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.

9.Appendix.:-

Source Code :


jupyter flight delay prediction maincode Last Checkpoint: a minute ago (unsaved changes)

In [1]:
```python
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import sklearn
import sys
import numpy
```

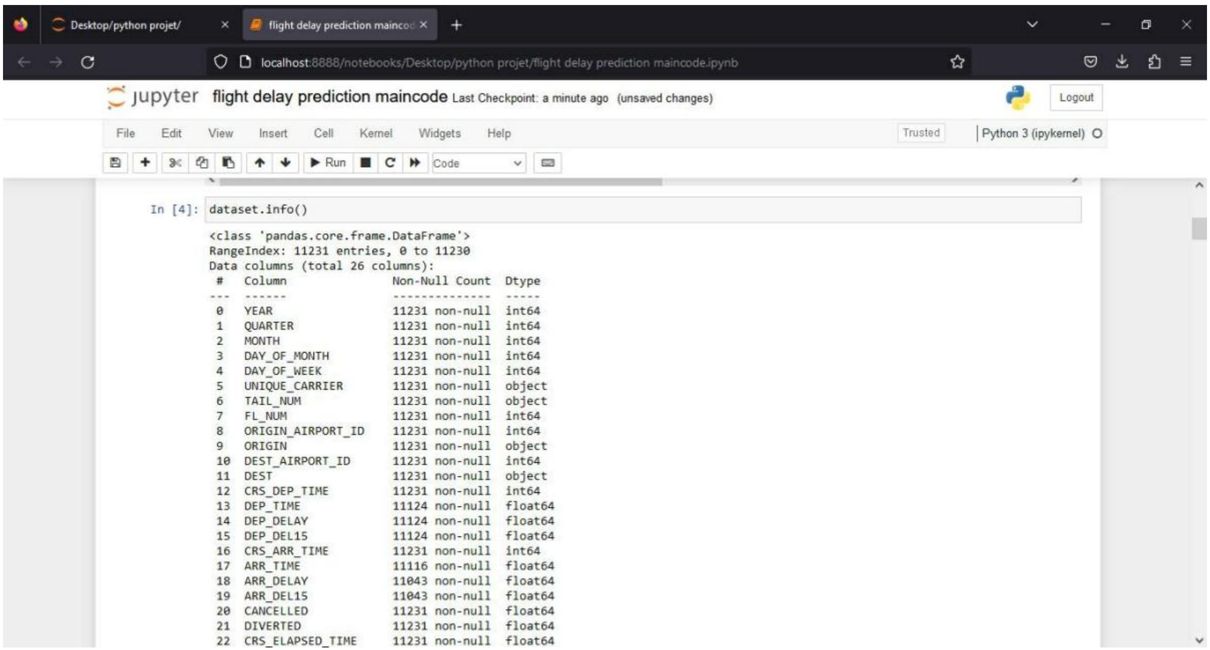
In [2]:
```python
dataset= pd.read_csv("flightdata.csv")
```

In [3]:
```python
dataset.head()
```

Out[3]:

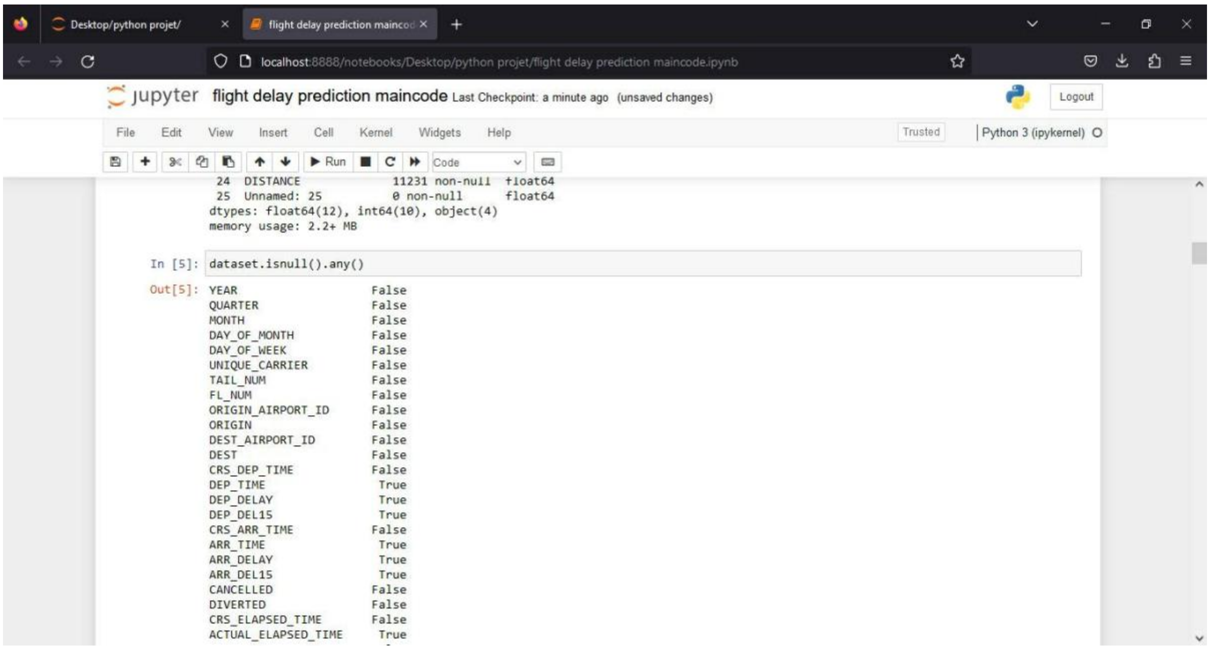
	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN	...	CRS_ARR_TIME	ARR_TIME
0	2016	1	1	1	5	DL	N836DN	1399	10397	ATL	...	2143	2102.0
1	2016	1	1	1	5	DL	N964DN	1476	11433	DTW	...	1435	1439.0
2	2016	1	1	1	5	DL	N813DN	1597	10397	ATL	...	1215	1142.0
3	2016	1	1	1	5	DL	N587NW	1768	14747	SEA	...	1335	1345.0
4	2016	1	1	1	5	DL	N836DN	1823	14747	SEA	...	607	615.0

5 rows × 26 columns


jupyter flight delay prediction maincode Last Checkpoint: a minute ago (unsaved changes)

In [4]:
```python
dataset.info()
```

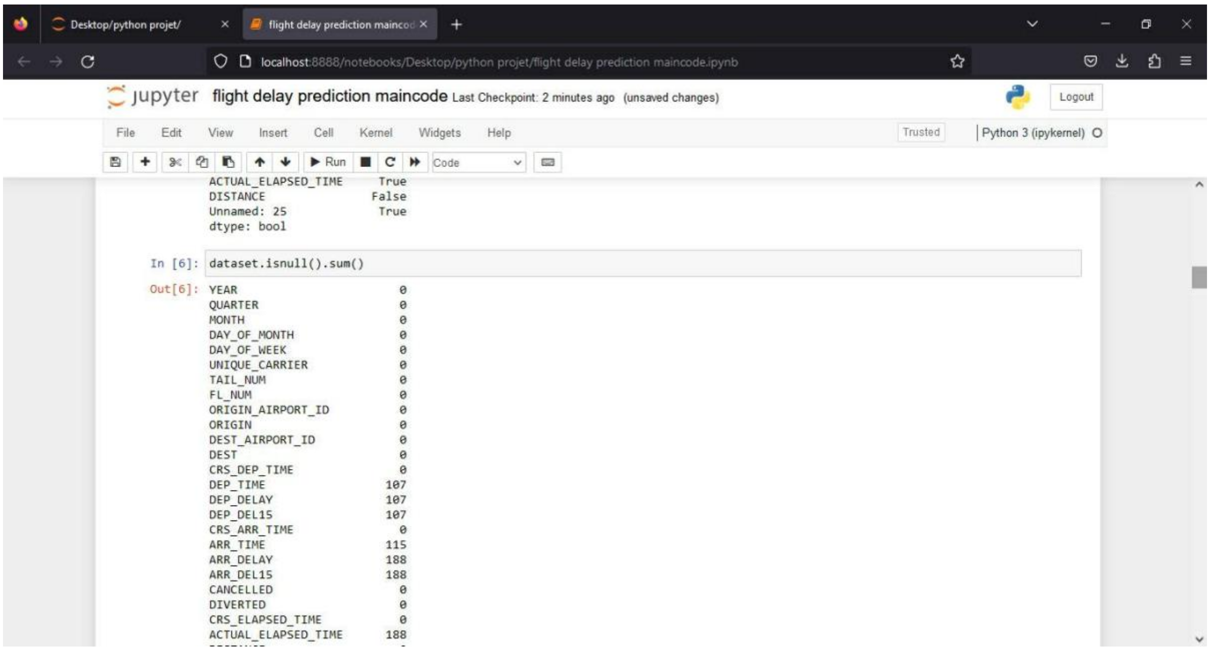
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11231 entries, 0 to 11230
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   YEAR               11231 non-null  int64
 1   QUARTER            11231 non-null  int64
 2   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
 15  DEP_DEL15          11124 non-null  float64
 16  CRS_ARR_TIME       11231 non-null  int64
 17  ARR_TIME           11116 non-null  float64
 18  ARR_DELAY          11043 non-null  float64
 19  ARR_DEL15          11043 non-null  float64
 20  CANCELLED          11231 non-null  float64
 21  DIVERTED           11231 non-null  float64
 22  CRS_ELAPSED_TIME   11231 non-null  float64
```




```
 24  DISTANCE              11231 non-null  float64
 25  Unnamed: 25           0 non-null      float64
dtypes: float64(12), int64(10), object(4)
memory usage: 2.2+ MB
```

```
In [5]: dataset.isnull().any()
```

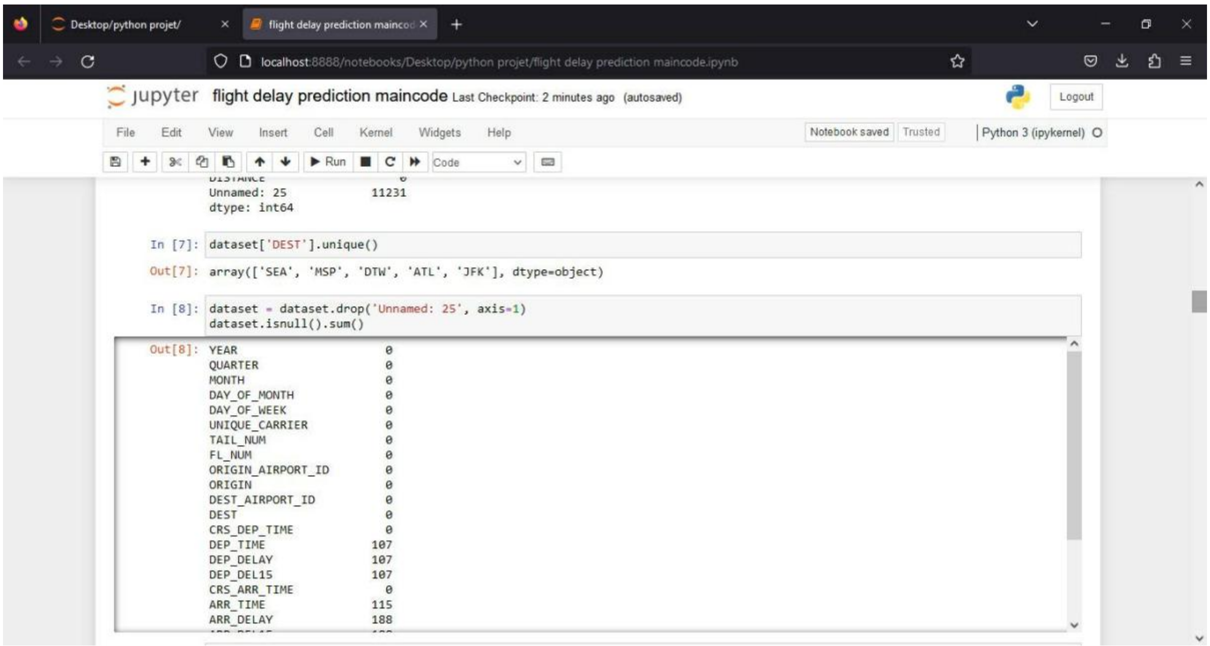
```
Out[5]: YEAR                   False
        QUARTER                False
        MONTH                  False
        DAY_OF_MONTH           False
        DAY_OF_WEEK            False
        UNIQUE_CARRIER         False
        TAIL_NUM               False
        FL_NUM                 False
        ORIGIN_AIRPORT_ID      False
        ORIGIN                 False
        DEST_AIRPORT_ID        False
        DEST                   False
        CRS_DEP_TIME           False
        DEP_TIME                True
        DEP_DELAY               True
        DEP_DEL15               True
        CRS_ARR_TIME           False
        ARR_TIME                True
        ARR_DELAY               True
        ARR_DEL15               True
        CANCELLED              False
        DIVERTED               False
        CRS_ELAPSED_TIME       False
        ACTUAL_ELAPSED_TIME     True
```




```
        ACTUAL_ELAPSED_TIME     True
        DISTANCE               False
        Unnamed: 25             True
        dtype: bool
```

```
In [6]: dataset.isnull().sum()
```

```
Out[6]: 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
```



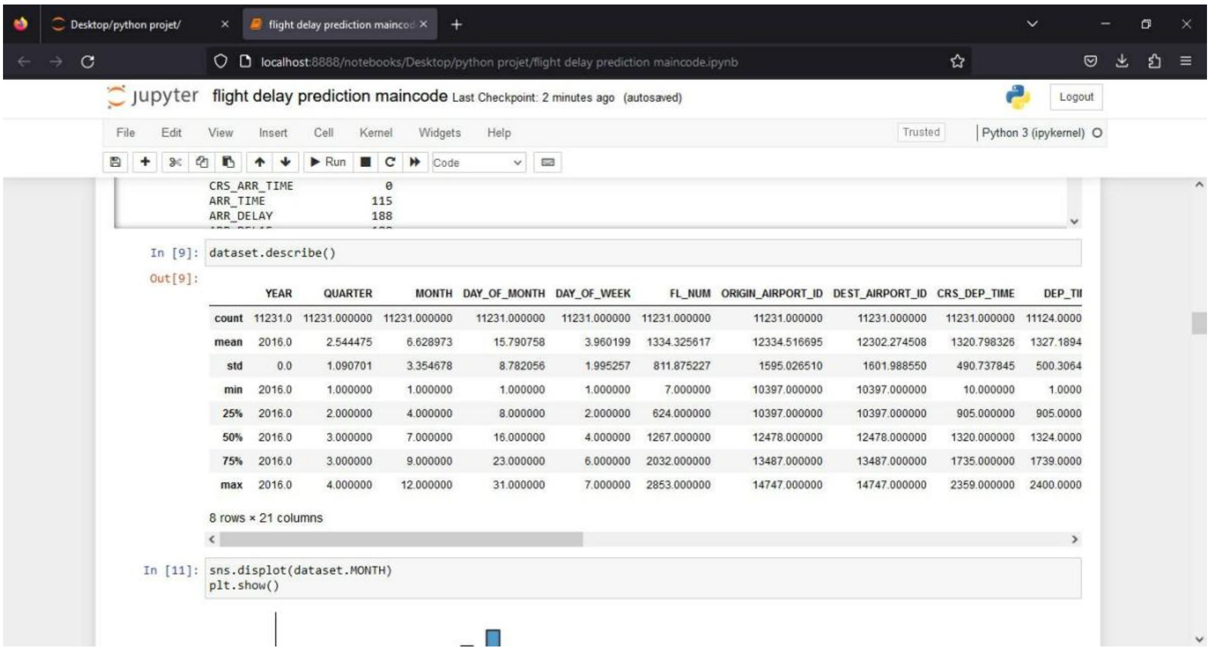
```
Unnamed: 25          11231
dtype: int64
```

```
In [7]: dataset['DEST'].unique()
```

```
Out[7]: array(['SEA', 'MSP', 'DTW', 'ATL', 'JFK'], dtype=object)
```

```
In [8]: dataset = dataset.drop('Unnamed: 25', axis=1)
        dataset.isnull().sum()
```

```
Out[8]: 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
```



```
CRS_ARR_TIME           0
ARR_TIME             115
ARR_DELAY            188
```

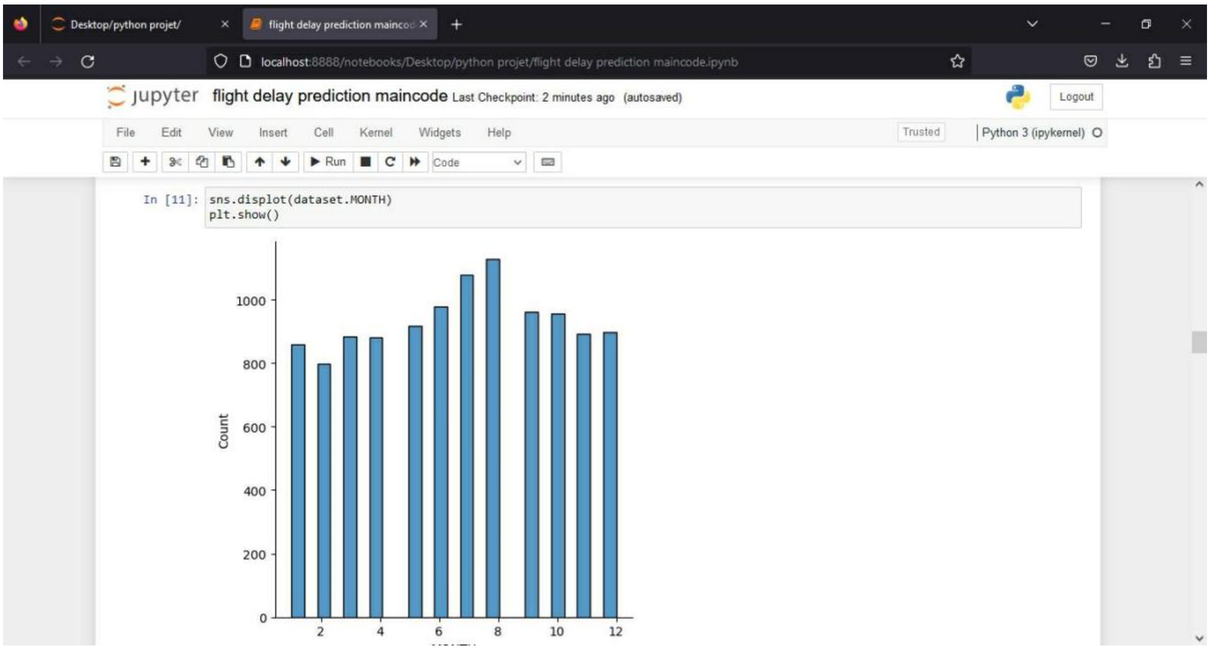
```
In [9]: dataset.describe()
```

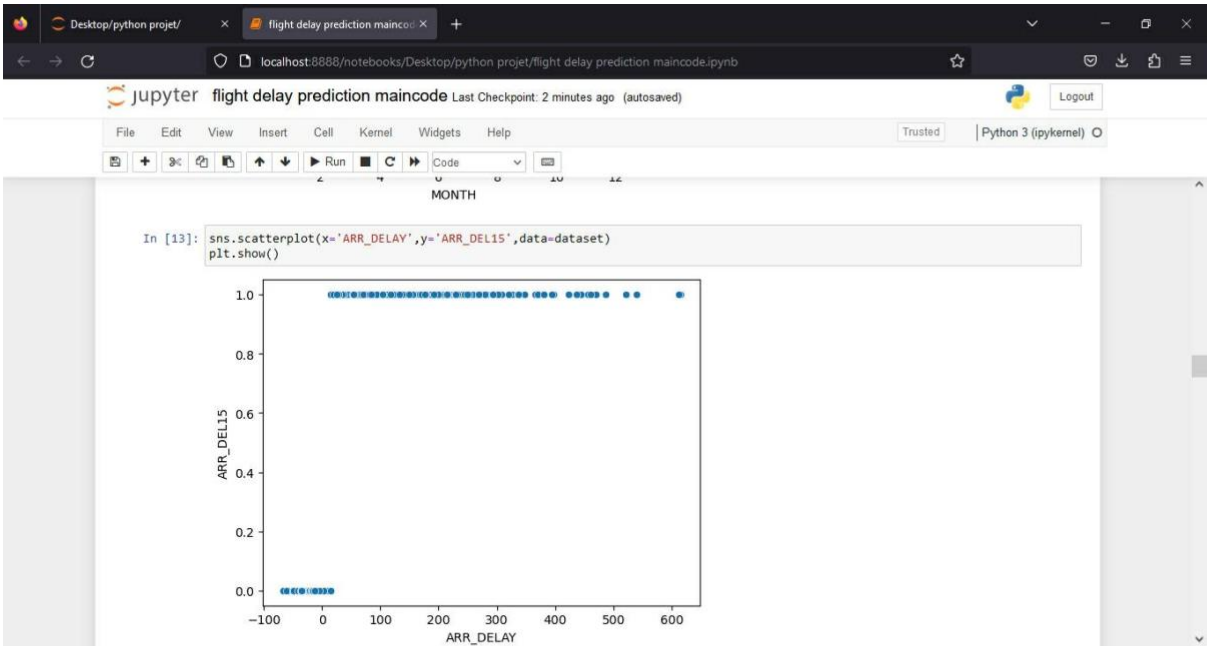
Out[9]:

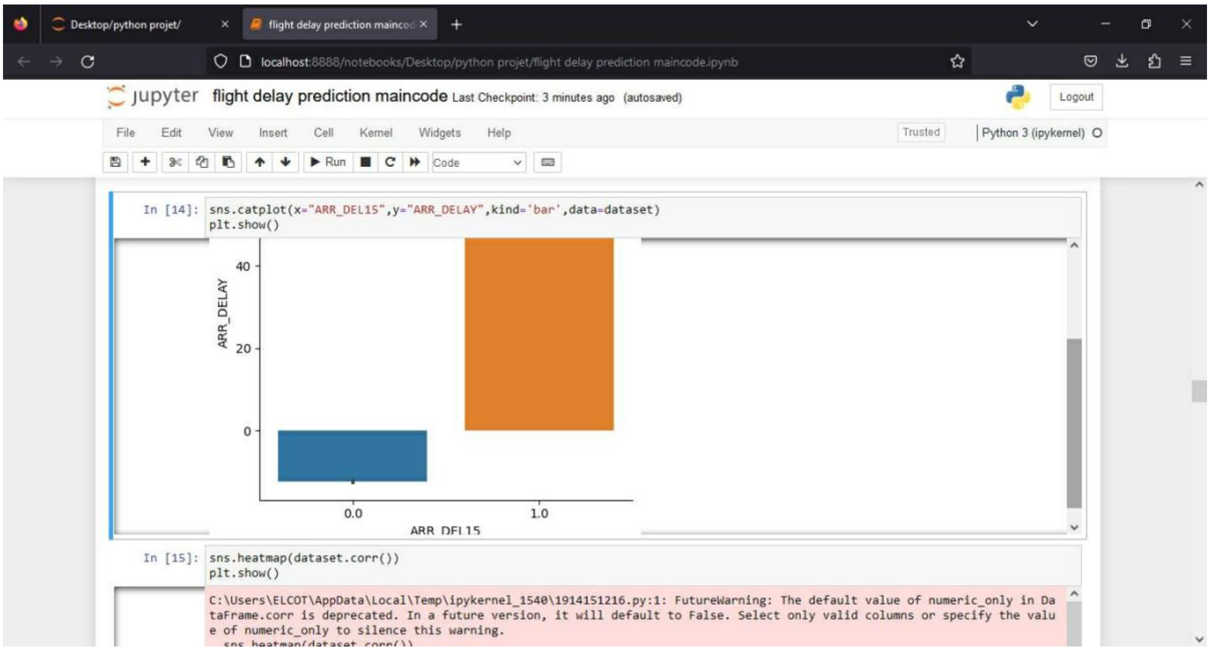
	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_NUM	ORIGIN_AIRPORT_ID	DEST_AIRPORT_ID	CRS_DEP_TIME	DEP_TI
count	11231.0	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000	11124.0000
mean	2016.0	2.544475	6.628973	15.790758	3.960199	1334.325617	12334.516695	12302.274508	1320.798326	1327.1894
std	0.0	1.090701	3.354678	8.782056	1.995257	811.875227	1595.026510	1601.988550	490.737845	500.3064
min	2016.0	1.000000	1.000000	1.000000	1.000000	7.000000	10397.000000	10397.000000	10.000000	1.0000
25%	2016.0	2.000000	4.000000	8.000000	2.000000	624.000000	10397.000000	10397.000000	905.000000	905.0000
50%	2016.0	3.000000	7.000000	16.000000	4.000000	1267.000000	12478.000000	12478.000000	1320.000000	1324.0000
75%	2016.0	3.000000	9.000000	23.000000	6.000000	2032.000000	13487.000000	13487.000000	1735.000000	1739.0000
max	2016.0	4.000000	12.000000	31.000000	7.000000	2853.000000	14747.000000	14747.000000	2359.000000	2400.0000

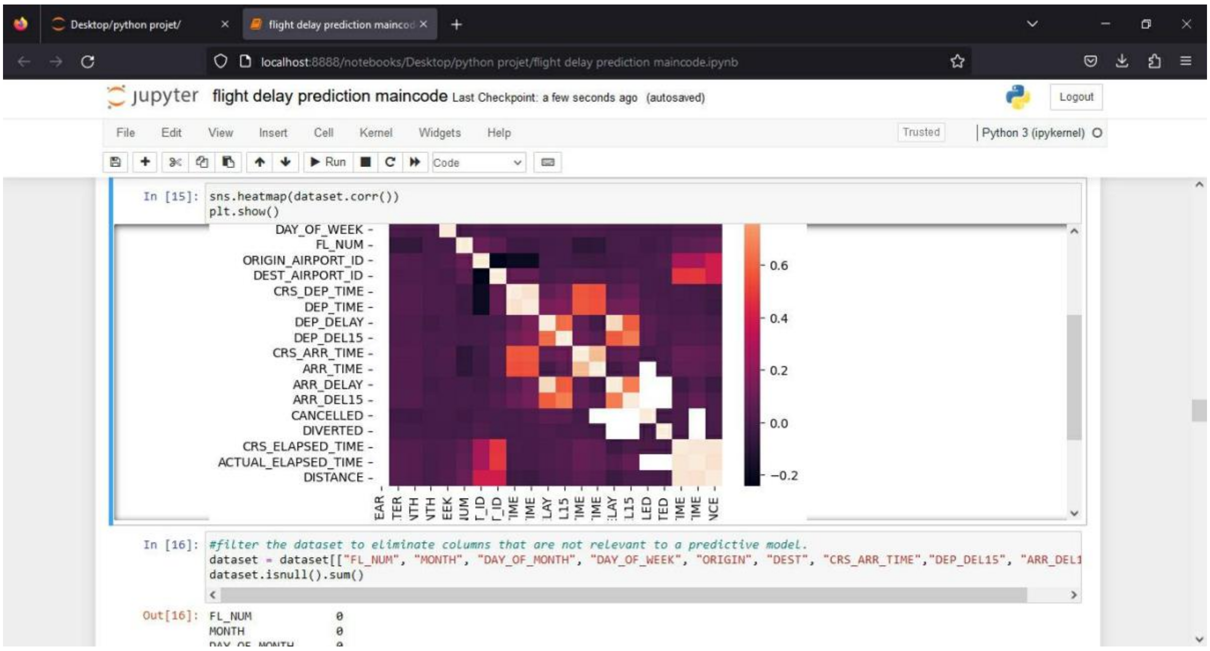
8 rows × 21 columns

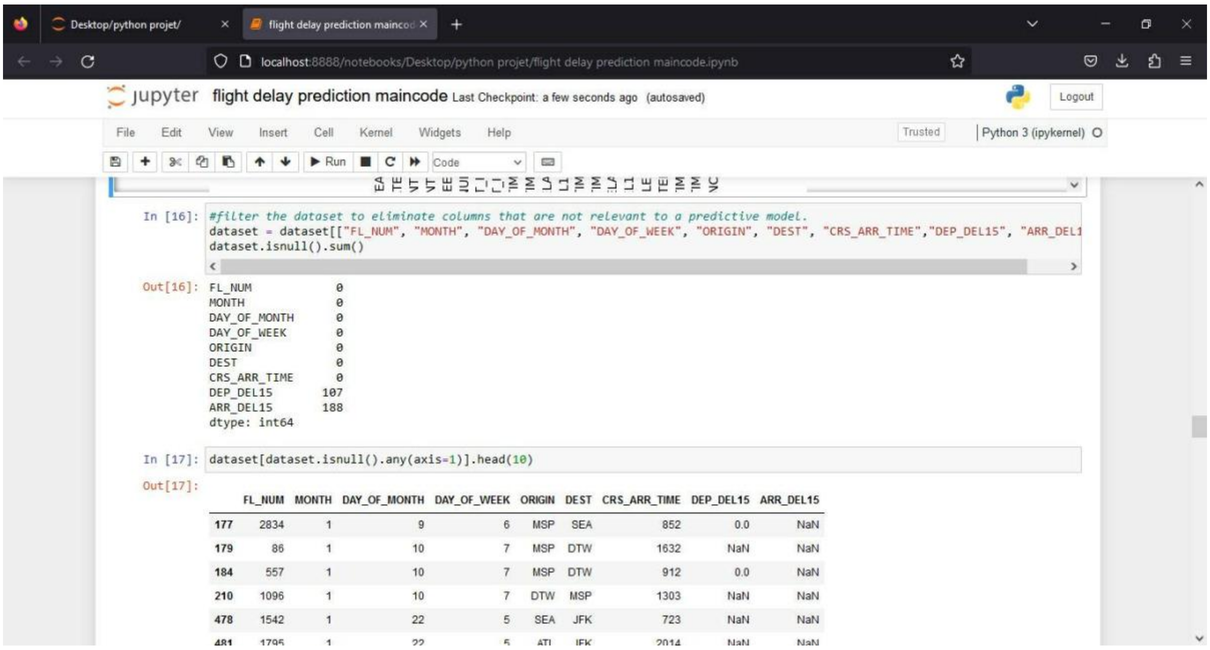
```
In [11]: sns.displot(dataset.MONTH)
         plt.show()
```


Desktop/python projet/
flight delay prediction mainco
localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb
jupyter flight delay prediction maincode Last Checkpoint: 2 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [11]: sns.displot(dataset.MONTH)
plt.show()
Count
1000
800
600
400
200
0
2
4
6
8
10
12


Desktop/python projet/
flight delay prediction mainco
localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb
jupyter flight delay prediction maincode Last Checkpoint: 2 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
MONTH
In [13]: sns.scatterplot(x='ARR_DELAY',y='ARR_DEL15',data=dataset)
plt.show()
ARR_DEL15
1.0
0.8
0.6
0.4
0.2
0.0
−100
0
100
200
300
400
500
600
ARR_DELAY


Desktop/python projet/
flight delay prediction maincod
localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb
jupyter flight delay prediction maincode Last Checkpoint: 3 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [14]: sns.catplot(x="ARR_DEL15",y="ARR_DELAY",kind='bar',data=dataset)
plt.show()
ARR_DELAY
40
20
0
0.0
1.0
ARR_DEL15
In [15]: sns.heatmap(dataset.corr())
plt.show()
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_1540\1914151216.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.


Desktop/python projet/
flight delay prediction maincod
localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb
jupyter flight delay prediction maincode Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [15]: sns.heatmap(dataset.corr())
plt.show()
DAY_OF_WEEK
FL_NUM
ORIGIN_AIRPORT_ID
DEST_AIRPORT_ID
CRS_DEP_TIME
DEP_TIME
DEP_DELAY
DEP_DEL15
CRS_ARR_TIME
ARR_TIME
ARR_DELAY
ARR_DEL15
CANCELLED
DIVERTED
CRS_ELAPSED_TIME
ACTUAL_ELAPSED_TIME
DISTANCE
0.6
0.4
0.2
0.0
−0.2
In [16]: #filter the dataset to eliminate columns that are not relevant to a predictive model.
dataset = dataset[["FL_NUM", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "ORIGIN", "DEST", "CRS_ARR_TIME","DEP_DEL15", "ARR_DEL1
dataset.isnull().sum()
Out[16]: FL_NUM 0
MONTH 0



Desktop/python projet/ | flight delay prediction maincode

localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb

Jupyter flight delay prediction maincode Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

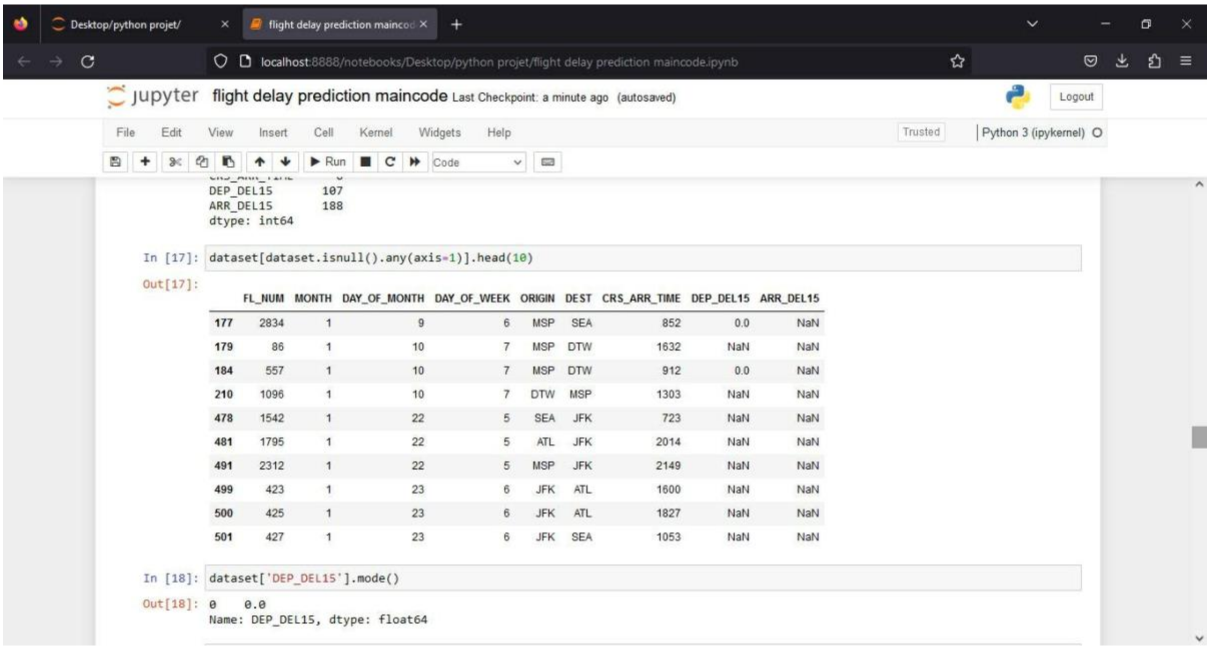
In [16]:
```
#filter the dataset to eliminate columns that are not relevant to a predictive model.
dataset = dataset[["FL_NUM", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "ORIGIN", "DEST", "CRS_ARR_TIME","DEP_DEL15", "ARR_DEL1
dataset.isnull().sum()
```

Out[16]:
```
FL_NUM            0
MONTH             0
DAY_OF_MONTH      0
DAY_OF_WEEK       0
ORIGIN            0
DEST              0
CRS_ARR_TIME      0
DEP_DEL15       107
ARR_DEL15       188
dtype: int64
```

In [17]:
```
dataset[dataset.isnull().any(axis=1)].head(10)
```

Out[17]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
177	2834	1	9	6	MSP	SEA	852	0.0	NaN
179	86	1	10	7	MSP	DTW	1632	NaN	NaN
184	557	1	10	7	MSP	DTW	912	0.0	NaN
210	1096	1	10	7	DTW	MSP	1303	NaN	NaN
478	1542	1	22	5	SEA	JFK	723	NaN	NaN



Desktop/python projet/ | flight delay prediction maincode

localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb

Jupyter flight delay prediction maincode Last Checkpoint: a minute ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
DEP_DEL15       107
ARR_DEL15       188
dtype: int64
```

In [17]:
```
dataset[dataset.isnull().any(axis=1)].head(10)
```

Out[17]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
177	2834	1	9	6	MSP	SEA	852	0.0	NaN
179	86	1	10	7	MSP	DTW	1632	NaN	NaN
184	557	1	10	7	MSP	DTW	912	0.0	NaN
210	1096	1	10	7	DTW	MSP	1303	NaN	NaN
478	1542	1	22	5	SEA	JFK	723	NaN	NaN
481	1795	1	22	5	ATL	JFK	2014	NaN	NaN
491	2312	1	22	5	MSP	JFK	2149	NaN	NaN
499	423	1	23	6	JFK	ATL	1600	NaN	NaN
500	425	1	23	6	JFK	ATL	1827	NaN	NaN
501	427	1	23	6	JFK	SEA	1053	NaN	NaN

In [18]:
```
dataset['DEP_DEL15'].mode()
```

Out[18]:
```
0    0.0
Name: DEP_DEL15, dtype: float64
```

Desktop/python projet/ × | flight delay prediction maincod × | +

localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb

jupyter flight delay prediction maincode Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)

In [19]:
```
#replace the missing values with 1s.
dataset = dataset.fillna({'ARR_DEL15': 1})
dataset = dataset.fillna({'DEP_DEL15': 0})
dataset.iloc[177:185]
```

Out[19]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
177	2834	1	9	6	MSP	SEA	852	0.0	1.0
178	2839	1	9	6	DTW	JFK	1724	0.0	0.0
179	86	1	10	7	MSP	DTW	1632	0.0	1.0
180	87	1	10	7	DTW	MSP	1649	1.0	0.0
181	423	1	10	7	JFK	ATL	1600	0.0	0.0
182	440	1	10	7	JFK	ATL	849	0.0	0.0
183	485	1	10	7	JFK	SEA	1945	1.0	0.0
184	557	1	10	7	MSP	DTW	912	0.0	1.0

In [20]:
```
import math

for index, row in dataset.iterrows():
    dataset.loc[index, 'CRS_ARR_TIME'] = math.floor(row['CRS_ARR_TIME'] / 100)
dataset.head()
```

Out[20]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15

Desktop/python projet/ × | flight delay prediction maincod × | +

localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb

jupyter flight delay prediction maincode Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)

In [20]:
```
import math

for index, row in dataset.iterrows():
    dataset.loc[index, 'CRS_ARR_TIME'] = math.floor(row['CRS_ARR_TIME'] / 100)
dataset.head()
```

Out[20]:

	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
0	1399	1	1	5	ATL	SEA	21	0.0	0.0
1	1476	1	1	5	DTW	MSP	14	0.0	0.0
2	1597	1	1	5	ATL	SEA	12	0.0	0.0
3	1768	1	1	5	SEA	MSP	13	0.0	0.0
4	1823	1	1	5	SEA	DTW	6	0.0	0.0

In [21]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['DEST'] = le.fit_transform(dataset['DEST'])
dataset['ORIGIN'] = le.fit_transform(dataset['ORIGIN'])
```

In [22]:
```
dataset.head(5)
```

Out[22]:

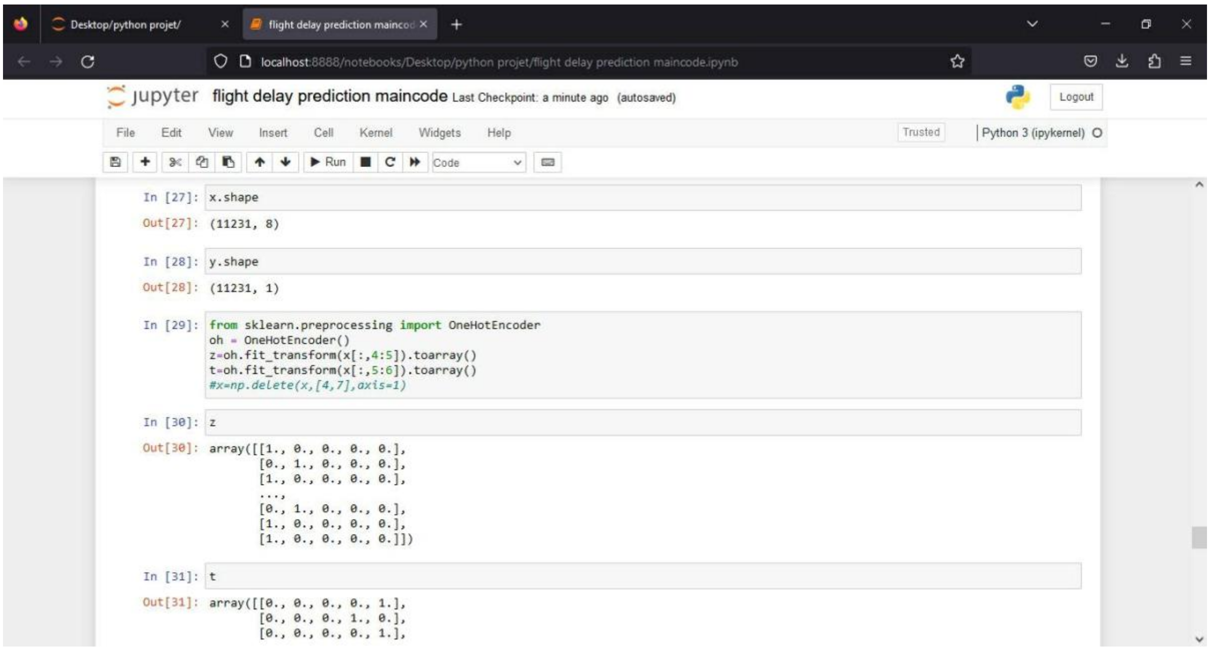
	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
0	1399	1	1	5	0	4	21	0.0	0.0
1	1476	1	1	5	1	3	14	0.0	0.0

```
In [22]: dataset.head(5)
Out[22]:
```

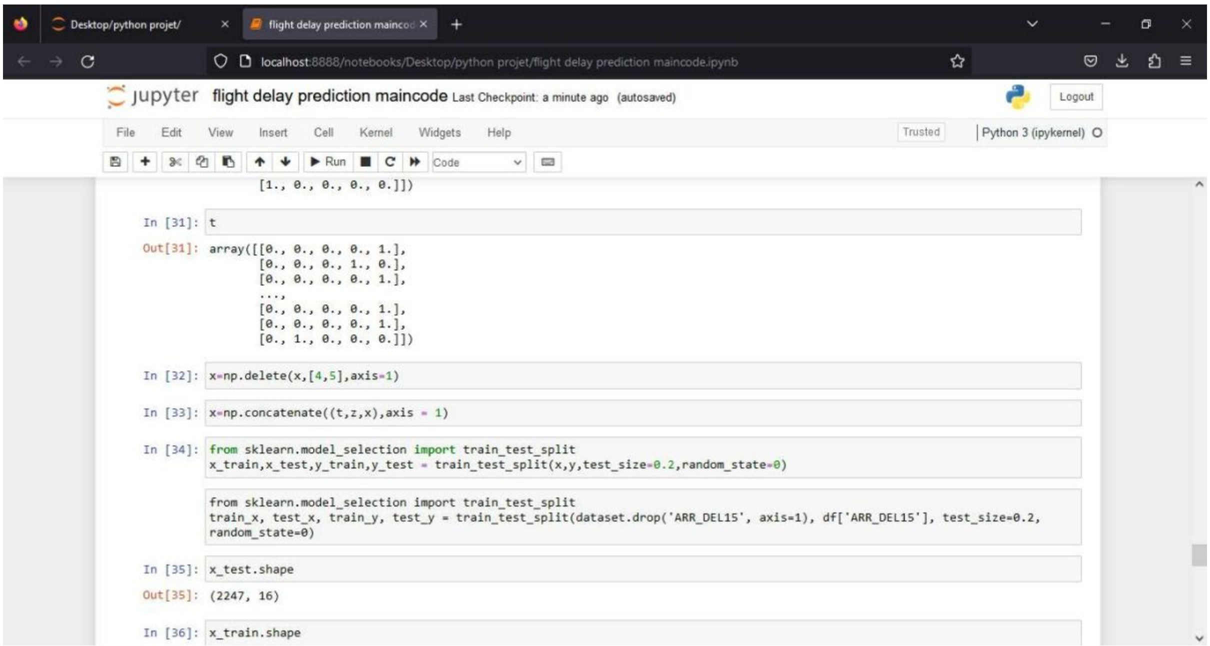
	FL_NUM	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	ORIGIN	DEST	CRS_ARR_TIME	DEP_DEL15	ARR_DEL15
0	1399	1	1	5	0	4	21	0.0	0.0
1	1476	1	1	5	1	3	14	0.0	0.0
2	1597	1	1	5	0	4	12	0.0	0.0
3	1768	1	1	5	4	3	13	0.0	0.0
4	1823	1	1	5	4	1	6	0.0	0.0

```
In [23]: dataset['ORIGIN'].unique()
Out[23]: array([0, 1, 4, 3, 2])

         dataset = pd.get_dummies(dataset, columns=['ORIGIN', 'DEST'])
         dataset.head()

In [24]: x = dataset.iloc[:, 0:8].values
         y = dataset.iloc[:, 8:9].values

In [25]: x

Out[25]: array([[1.399e+03, 1.000e+00, 1.000e+00, ..., 4.000e+00, 2.100e+01,
                 0.000e+00],
                [1.476e+03, 1.000e+00, 1.000e+00, ..., 3.000e+00, 1.400e+01,
```

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

In [25]: x

Out[25]: array([[1.399e+03, 1.000e+00, 1.000e+00, ..., 4.000e+00, 2.100e+01,
                 0.000e+00],
                [1.476e+03, 1.000e+00, 1.000e+00, ..., 3.000e+00, 1.400e+01,
                 0.000e+00],
                [1.597e+03, 1.000e+00, 1.000e+00, ..., 4.000e+00, 1.200e+01,
                 0.000e+00],
                ...,
                [1.823e+03, 1.200e+01, 3.000e+01, ..., 4.000e+00, 2.200e+01,
                 0.000e+00],
                [1.901e+03, 1.200e+01, 3.000e+01, ..., 4.000e+00, 1.800e+01,
                 0.000e+00],
                [2.005e+03, 1.200e+01, 3.000e+01, ..., 1.000e+00, 9.000e+00,
                 0.000e+00]])

In [26]: y

Out[26]: array([[0.],
                [0.],
                [0.],
                ...,
                [0.],
                [0.],
                [0.]])

In [27]: x.shape
```



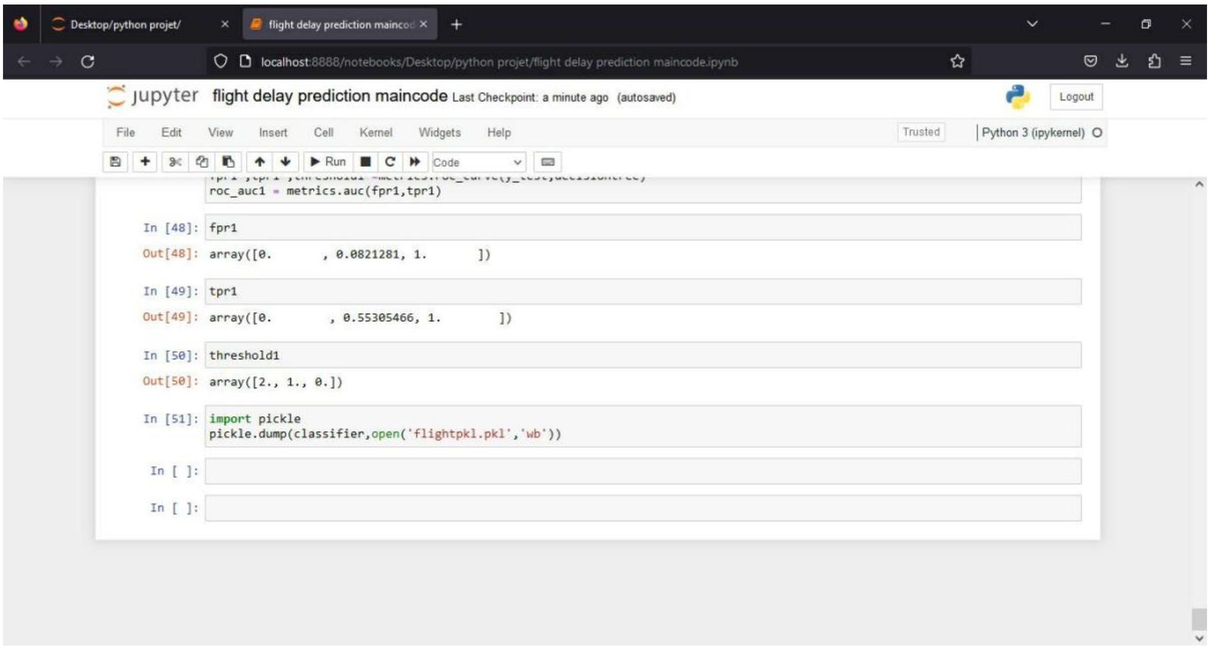

```
In [27]: x.shape
Out[27]: (11231, 8)

In [28]: y.shape
Out[28]: (11231, 1)

In [29]: from sklearn.preprocessing import OneHotEncoder
         oh = OneHotEncoder()
         z=oh.fit_transform(x[:,4:5]).toarray()
         t=oh.fit_transform(x[:,5:6]).toarray()
         #x=np.delete(x,[4,7],axis=1)

In [30]: z
Out[30]: array([[1., 0., 0., 0., 0.],
                [0., 1., 0., 0., 0.],
                [1., 0., 0., 0., 0.],
                ...,
                [0., 1., 0., 0., 0.],
                [1., 0., 0., 0., 0.],
                [1., 0., 0., 0., 0.]])

In [31]: t
Out[31]: array([[0., 0., 0., 0., 1.],
                [0., 0., 0., 1., 0.],
                [0., 0., 0., 0., 1.],
```




```
                [1., 0., 0., 0., 0.]])

In [31]: t
Out[31]: array([[0., 0., 0., 0., 1.],
                [0., 0., 0., 1., 0.],
                [0., 0., 0., 0., 1.],
                ...,
                [0., 0., 0., 0., 1.],
                [0., 0., 0., 0., 1.],
                [0., 1., 0., 0., 0.]])

In [32]: x=np.delete(x,[4,5],axis=1)

In [33]: x=np.concatenate((t,z,x),axis = 1)

In [34]: 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['ARR_DEL15'], test_size=0.2,
         random_state=0)

In [35]: x_test.shape
Out[35]: (2247, 16)

In [36]: x_train.shape
```

```
In [37]: y_test.shape
Out[37]: (2247, 1)

In [38]: y_train.shape
Out[38]: (8984, 1)

In [39]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         x_train = sc.fit_transform(x_train)
         x_test = sc.transform(x_test)

In [40]: from sklearn.tree import DecisionTreeClassifier
         classifier = DecisionTreeClassifier(random_state = 0)
         classifier.fit(x_train,y_train)

Out[40]: DecisionTreeClassifier
         DecisionTreeClassifier(random_state=0)

In [41]: decisiontree = classifier.predict(x_test)

In [42]: decisiontree
Out[42]: array([1., 0., 0., ..., 0., 0., 1.])
```

```
In [41]: decisiontree = classifier.predict(x_test)

In [42]: decisiontree
Out[42]: array([1., 0., 0., ..., 0., 0., 1.])

In [43]: from sklearn.metrics import accuracy_score
         desacc = accuracy_score(y_test,decisiontree)

In [44]: desacc
Out[44]: 0.8673787271918113

In [45]: from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test,decisiontree)

In [46]: cm
Out[46]: array([[1777,  159],
                [ 139,  172]], dtype=int64)

In [47]: import sklearn.metrics as metrics
         fpr1 ,tpr1 ,threshold1 =metrics.roc_curve(y_test,decisiontree)
         roc_auc1 = metrics.auc(fpr1,tpr1)

In [48]: fpr1
```


Desktop/python projet/
flight delay prediction maincod
localhost:8888/notebooks/Desktop/python projet/flight delay prediction maincode.ipynb
jupyter flight delay prediction maincode Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
roc_auc1 = metrics.auc(fpr1,tpr1)
In [48]: fpr1
Out[48]: array([0.   , 0.0821281, 1.   ])
In [49]: tpr1
Out[49]: array([0.   , 0.55305466, 1.   ])
In [50]: threshold1
Out[50]: array([2., 1., 0.])
In [51]: import pickle
pickle.dump(classifier,open('flightpkl.pkl','wb'))
In [ ]:
In [ ]: