

FLIGHT DELAY PREDICTION MODEL

1. Introduction

1.1 Project Overview

In this study, multiple ML forecasting algorithms are applied to predict flight delays at an American airport. Techniques such as Decision Trees, Random Forest, and KNearest Neighbors are used for predicting individual flight delays. Binary classification performed by the model helps predict the scheduled flight delay. These algorithms estimate well the distribution of arrival and departure flight delays with a Mean Absolute Error of less than 15 min. To illustrate the utility of the estimated delay distributions, integration of these probabilistic predictions into a probabilistic flight-to-gate assignment problem is done. The objective of this problem is to increase the robustness of flight-to-gate assignments. To summarize, the results illustrate the utility of considering probabilistic forecasting for robust airport operations' optimization.

1.2 Purpose

A flight delay is when an airline flight takes off and/or lands later than its scheduled time. The Federal Aviation Administration (FAA) considers a flight to be delayed when it is 15 minutes later than its scheduled time. A cancellation occurs when the airline does not operate the flight at all for a certain reason. This has led to phenomenal growth in air traffic and on the ground. An increase in air traffic growth has also resulted in massive levels of aircraft delays on the ground and in the air. These delays are responsible for large economic and environmental losses. The main objective of the model is to predict flight delays accurately in order to optimize flight operations and minimize delays. The problem of flight delay prediction is approached most often by predicting a delay class or value. However, the aviation industry can benefit greatly from probabilistic delay predictions on an individual flight basis, as these give insight into the uncertainty of the delay predictions.

2. LITERATURE SURVEY

2.1 Existing problem

Commercial aviation is a complex distributed transportation system. It deals with valuable resources, demand fluctuations, and a sophisticated origin-destination matrix that needs orchestration to provide smooth and safety operations. Furthermore, individual passengers follow her itineraries while airlines plan various schedules for aircrafts, pilots and flight attendants. Stages can take place at terminal boundaries, airports, runways, and airspace, being susceptible to different kinds of delays. Some examples include mechanical problems, weather conditions, ground delays, air traffic control, runway queues and capacity constraints.

2.2 References

http://scientiainica.sharif.edu/article_20020_0.html

<https://ieeexplore.ieee.org/document/8903554>

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z>

<https://ieeexplore.ieee.org/document/9633571>

<https://ieeexplore.ieee.org/document/9512525>

<https://ieeexplore.ieee.org/document/8373742>

2.3 Problem Statement Definition

Analysis of flight delay and causal factors is crucial in maintaining airspace efficiency and safety. However, delay samples are not independent since they always show a certain aggregation pattern. Therefore, this study develops a novel spatial analysis approach to explore the delay and causal factors which is able to take dependence and the possible problem involved including error correlation and variable lag effect of causal factors on delay into account using a Machine Learning Model.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Manishankar

The passenger should be able to get the intimation of a flight delay a few hours before.

Should state the reason behind the delay

The application should make arrangements for lounge or waiting area

In case the delay time is more, in that case the application should help the user get a compensation.

The app should be user friendly.

The passenger should get to know the updated time of arrival and departure.

Monitor the flights arrival and departure.

Connecting Social media to get weather prediction and other problems.

Passenger flight details should match with the upcoming flights.

Passenger should get updated timings.

The departure timing and reason for delay must be known.

The passenger should get the intimation of flight in prior.

Weather details of the destination should be available for the user.

The app should be user friendly.

Ashwin

Anishvikram

Application should suggest ways to adjust passenger's leisure time.

Weather updates should be done frequently.

People should follow instructions to ensure smooth flight travels

If the flight gets delayed more often for a certain destination on the same day, in that case all flights must be cancelled.

Flight delay duration should be informed to the passengers.

Flight data should be updated frequently.

Weather predictions of a week are to be found earlier for smoother air routes.

Checking the airport websites for possible delays.

Research alternate flights from the same airlines or partnered airlines.

Understand the reason behind the delay.

Keeping an eye on the weather conditions.

Book a back up flight.

If external or unpredictable factors causes unexpected delays, it should be informed to passengers.

Check-in delays should be avoided.

Kiran

The passenger should be able to get the intimation of flight delay a hour or few hours before

The passenger should get the intimation of flight in prior

Passengers should get notified by the flight delay details

The passenger should get to know the updated time of arrival and departure

Flight delay duration should be informed to the passengers

Should get the updated timings

The application environment should be user friendly

The application must be user friendly

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Should state the reason for flight delay

Understand why flights get delayed

Monitor the flights arrival and departure

Flight data should be updated frequently

The departure timing and reason for the delay must be known

Weather updation should be done frequently

Weather prediction should be done before flight timings

Weather details of the destination should be available for the user

Weather predictions of one week should be found earlier for smooth air routes

Keeping an eye on the weather conditions

In case if the delay time is more in that case the application should help the user get a compensation

Book a backup flight in the last case

Re-search alternate flights with the same airline, partner airlines, even competitors

Application should suggest ways to adjust passenger's leisure time due to flight delay

People should follow certain instructions to ensure the smooth flight travels

Passengers flight details should match with the upcoming flights

Application should make arrangements for lounge or waiting area

If external or unpredictable factor causes the unexpected delay, it should be informed to passengers promptly

If flight gets delayed more often for a certain destination on the same day in that case all the flight must be cancelled for that destination on that day

Checking the airport website for possible related delays

Check-in delays should be avoided

Connecting social media to get weather prediction and other problems



3.3 Proposed Solution

<u>S.No</u>	<u>Parameter</u>	<u>Description</u>
1.	Problem Statement	Analysis of flight delay and causal factors is crucial in maintaining airspace efficiency and safety. However, delay samples are not independent since they always show a certain aggregation pattern. Therefore, this study develops a novel spatial analysis approach to explore the delay and causal factors which is able to take dependence and the possible problem involved including error correlation and variable lag effect of causal factors on delay into account using a Machine Learning Model.
2.	Idea/Solution Description	We can predict flight arrival delays using prediction models. The input to our algorithm is rows of feature vectors like departure date, departure delay, distance between the two airports, scheduled arrival time etc. We then use an ensemble of decision tree classifiers to predict if the flight arrival will be delayed or not. A flight is considered to be delayed if the difference between scheduled and actual arrival times is greater than 15 minutes. Furthermore, we compare decision tree classifiers with logistic regression, KNN, SVM and a simple neural network for various figures of merit.
3.	Novelty/Uniqueness	We can create an app which shows all sorts of delays with precision and accuracy. We can connect or interact through visually intelligent systems and make the app more simplified. Integration with the airline booking system can be done to increase efficiency. We can notify the user about delays through SMS or mail.
4.	Social Impact/Customer Satisfaction	Passenger groups include business people, tourists, civilians etc. Customers who are not satisfied with the delays can lessen their travel and this results in losses. By predicting flight

		<p>delay customer experience is improved and customers will have a better and peaceful experience. It can help customer to * decrease their waiting time. * provide complimentary snacks for using our app in case of delay. * suggest customers with the best nearby hotel with reviews in case they have to stay. * entertain customers with movies and songs through our app or provide wifi.</p>
5.	Business Model (Revenue Model)	<p>Through our application the revenue for the company will be in the form of advertisements. It also has the paid subscriptions which asks the user to pay for additional exclusive features.</p>
6.	Scalability of Solution	<p>The system can handle a large number of users. The scalability of this project includes incorporating a larger dataset for multiparty communication. The above methodology can be performed on the data collected for the recent years, owing to the population rise in recent years leading to increase in the number of flights. To obtain a detailed analysis, a more complete localized search and research can be conducted to accurately determine the arrival or departure delay. Integration with airline booking systems can yield good efficiency.</p>

3.4 Problem Solution fit

<p><u>1.Customer Segment</u></p> <p>Flight delays are gradually increasing and bring more financial difficulties and customer dissatisfaction to airline companies. To resolve this situation, supervised machine learning models were implemented to predict flight delays.</p>	<p><u>6.Customer Limitation</u></p> <p>The results show that adverse weather conditions, low ceilings, and low visibility conditions strongly influence flight delays. Similarly, Asfe et al. Investigated the major causal factors of flight delays by ranking different factors using the analytical hierarchical process.</p>	<p><u>5. Available Solutions</u></p> <p>The available models would just state the reason for the delay along with the updated time of arrival and departure.</p>
<p><u>2. Jobs-to-be-done/Problems</u></p> <p>Flight delays making it difficult for the passengers and causing financial losses, the dissatisfaction of passengers, time losses, loss of reputation and bad business relations.</p>	<p><u>9.Problem root/Cause</u></p> <p>Flight delay prediction problems can be treated by Different point of view: (i) delay propagation, (ii) root delay and cancellation. In delay propagation, one studies how delay propagates through the network of the transportation system .On the other hand, considering that new problems may happen eventually, it is also important to predict further delays and understand their causes.</p>	<p><u>7. Behavior</u></p> <p>Match the flight details with the scheduled flights or enter the flight details then check the time of scheduled arrival and departure and if the flight is delayed then find the updated time of arrival and departure and reason for the delay. The passenger can also check the availability of backup flights in case of long delays.</p>
<p><u>3.Triggers to Act</u></p> <p>The main public datasets and the papers analyzed, we have organized them main commonly attributes used into seven classes depicted in the data model . They abstract the main input attributes for delay prediction models.</p>	<p><u>10.Your Solution</u></p> <p>This context, researchers created Fight delay models for delay prediction over the last years, and this work contributes with an analysis of these models from a Data Science perspective. We developed a taxonomy scheme and it can be classified models with respect to detailed components.</p>	<p><u>8.Channels of Behavior</u></p> <p>A typical operation of a commercial Fight. Stages can take place at terminal boundaries, airports, runways, and airspace, being susceptible to different kinds of delays. Some examples include mechanical problems, weather conditions, ground delays, air traffic control, runway queues and capacity constraints.</p>
<p><u>4. Emotions</u></p> <p>Due to delays in flights the entire plan of the passengers would be collapsed but with the predictions of flight delay the passengers can manage their time in an efficient and effective way.</p>		

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User payment	User pay via paypal , cards
FR-4	User requirements	Name ,address ,age
FR-5	User friendliness	This system is easy to learn and understand.

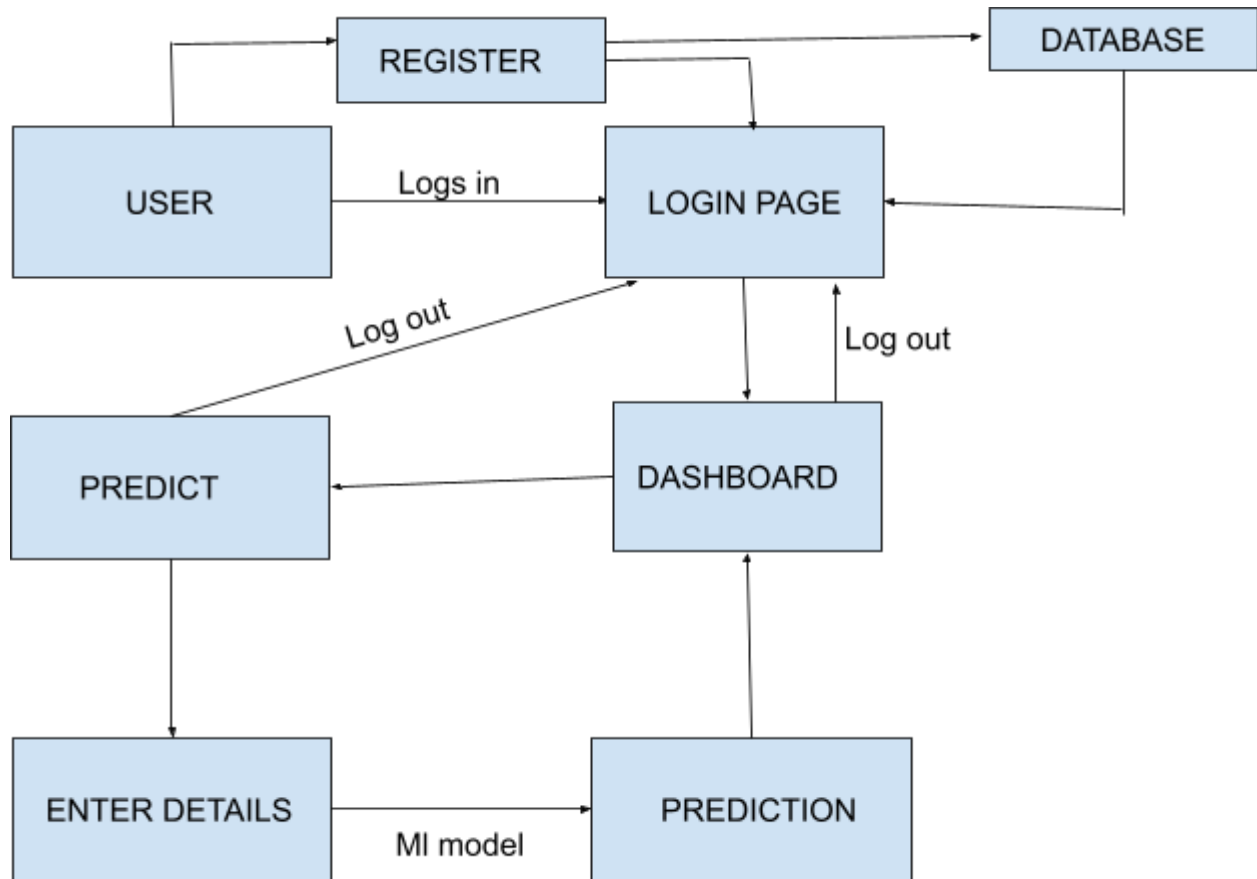
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	How easy is it for a customer to use the system?
NFR-2	Security	Security Its part will be protected against malware attacks or unauthorized access. But there's a catch. The lion's share of security non-functional requirements can be translated into concrete functional counterparts. If you want to protect the admin panel from unauthorized access, you would define the login flow and different user roles as system behavior or user actions.
NFR-3	Reliability	Reliability Specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions. Traditionally, this probability is expressed in percentages. For instance, if the system has 85 percent reliability for a month, this means that during this month, under normal usage conditions, there's an 85 percent chance that the system won't experience critical failure.
NFR-4	Performance	Performance defines how fast a software system or a particular piece of it responds to certain users' actions under a certain workload. In most cases, this metric explains how long a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it's not always like that. Performance requirements may describe background processes invisible to users, e.g. backup. But let's focus on user-centric performance.

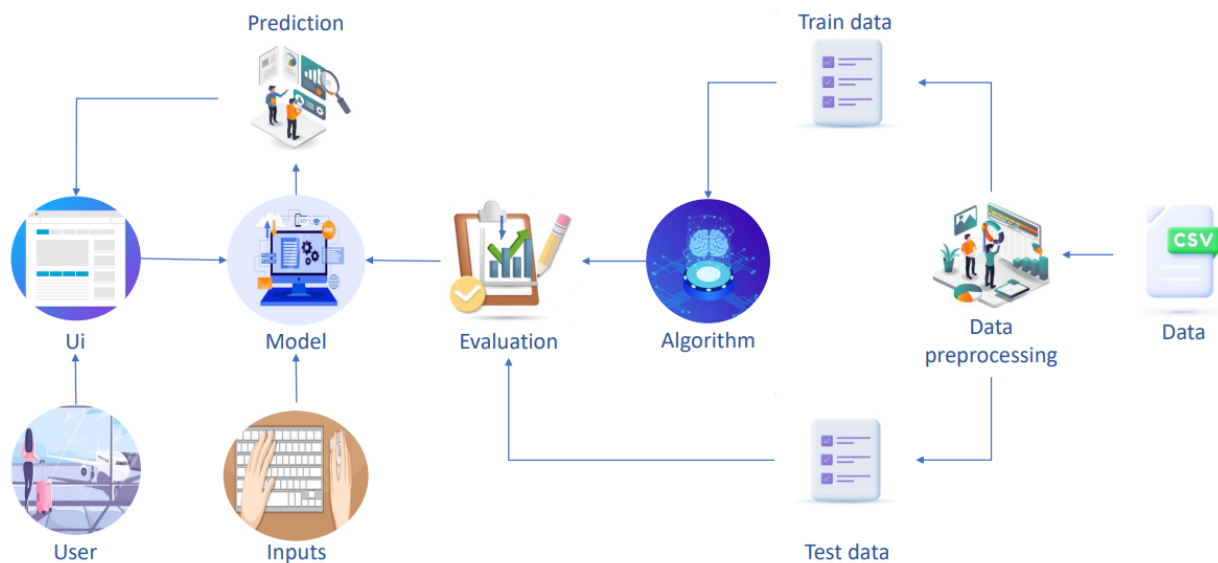
NFR-5	Availability	<p>Availability describes how likely the system is accessible to a user at a given point in time. While it can be expressed as an expected percentage of successful requests, you may also define it as a percentage of time the system is accessible for operation during some time period. For instance, the system may be available 98 percent of the time during a month. Availability is perhaps the most</p>
		<p>business-critical requirement, but to define it, you also must have estimations for reliability and maintainability.</p>
NFR-6	Scalability	<p>Scalability assesses the highest workloads under which the system will still meet the performance requirements. There are two ways to enable your system scale as the workloads get higher: horizontal and vertical scaling.</p>

5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture



S.No	Component	Description	Technology
1.	User Interface	User interacts using web application- Web UI	HTML, CSS, JavaScript / Angular js.
2.	Application Logic-1	Logic for preprocessing.	Python (numpy libraries)
3.	Application Logic-2	Logic for a training and testing	Sklearn library (train_test_split function)
4.	Application Logic-3	Logic for model building	Machine learning model Decision tree classifier .
5.	Database	Database contains the user information and flight details.	MySQL.
6.	File Storage	File storage requirements	IBM Block Storage , Local Filesystem.
7.	External API-1	Time Door is a REST API for statistical insights into time series data.	Http, Timedoor.
8.	Machine Learning Model	Ensemble of multiple decision trees provide better classification accuracy. Random forest obtains a class vote for decision tree.	Random forest , Decision tree classifier.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning, Estimation & Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection and Preprocessing	USN-1	As a user, I am unable to engage with anything.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-1	Build HTML Pages	USN-2	As a user, I can view the web pages to enter flight details.	1	Medium	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-1	User registration and Login	USN-3	As a user, I am able to login successfully and register my login info in the database.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-2	Build Python Pages	USN-4	As a user, I am unable to engage with anything.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-2	Build a user dashboard	USN-5	As a user, I can view the flight delay dashboard or insights on flight delays in the visual form.	2	Medium	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-3	Train the ML Model	USN-6	As a user, I can predict flight delays using the best created ML models.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-3	Save the trained model and use it to obtain predictions.	USN-7	As a user, I can give flight details as input to the ML model.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin

Sprint-4	Model Deployment on IBM Cloud using IBM Watson	USN-8	As a user, I can use the model by requesting the deployed model on Cloud.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin
Sprint-4	Integrate Flask with Model for the user to obtain predictions	USN-9	As a user, I can predict flight delays using the user interface.	2	High	Manishankar Kiran Kannan Anishvikram Ashwin

Sprint	Total Story Points	Duration	Start Date	End Date	Story Points Completed	Final Release Date
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1:

Users can register their account and only secured users can login.

Code snippet:

```
@app.route('/', methods=('GET','POST'))
def home():
    if request.method=='POST':
        email=request.form['em']
        uname=request.form['unme']
        password=request.form['pswd']
        login.insert_one({"email":email,"username":uname,"password":password})
        return render_template('login.html')
    return render_template('login.html')

@app.route('/sign')
```

```

def sign():
    return render_template('signup.html')

@app.route('/dash', methods=('GET','POST'))
def dashh():
    if request.method=='POST':
        coll=login.find()
        uname=request.form['uname']
        password=request.form['psw']
        for i in coll:
            if(uname==i['username'] and password==i['password']):
                return render_template('dashboard.html')
    if request.method=='GET':
        return render_template('dashboard.html')

    return render_template('login.html')

```

7.2 Feature 2:

Users can predict the flight delay using the deployed ML model by giving inputs of their flight details.

Code snippet:

```

@app.route('/predict',methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """

    b=int(request.form["month"])
    c=request.form["daym"]
    d=request.form["dayw"]
    e=request.form["fnum"]
    f=int(request.form["airport"])
    g=int(request.form["airportd"])
    h=request.form["dtime"]
    i=request.form["atime"]
    j=request.form["ttime"]
    if b==1 or b==2 or b==3:
        a=1
        l=2
    elif b==4 or b==5:

```



```
a=2
l=3
elif b==6:
    a=2
    l=0
elif b==7 or b==8:
    a=3
    l=0
elif b==9:
    a=3
    l=1
elif b==10 or b==11:
    l=1
    a=4
elif b==12:
    a=4
    l=2
ff=f
gg=g
if ff==gg:
    return render_template('summa.html', prediction_text='No delay(same airport!))')
if ff<gg:
    ff,gg=gg,ff
if gg==1 and ff==2:
    k=594
elif gg==1 and ff==3:
    k=760

elif gg==1 and ff==4:
    k=907

elif gg==1 and ff==5:
    k=2182

elif gg==2 and ff==3:
    k=509

elif gg==2 and ff==4:
    k=528
```

```
elif gg==2 and ff==5:  
    k=1927
```

```
elif gg==3 and ff==4:  
    k=1029
```

```
elif gg==3 and ff==5:  
    k=2422
```

```
elif gg==4 and ff==5:  
    k=1399
```

```
#print (a,b,c,d,e,f,g,h,i,j,k,l)
```

```
payload_scoring = {"input_data": [{"field":  
[["QUARTER","MONTH","DAY_OF_MONTH","DAY_OF_WEEK","FL_NUM","ORIGIN","  
DEST","CRS_DEP_TIME","CRS_ARR_TIME","CRS_ELAPSED_TIME","DISTANCE","SEA  
SON"]], "values": [[a,b,c,d,e,f,g,h,i,j,k,l]]}]}
```

```
response_scoring =  
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7d6c3b49-ec70-4cfe-ab88-  
4f6dbe6a7997/predictions?version=2022-11-16', json=payload_scoring,  
    headers={'Authorization': 'Bearer ' + mltoken})  
#print("Scoring response")  
predictions=response_scoring.json()  
m=predictions['predictions'][0]['values'][0][0]
```

```
if m==0:  
    return render_template('pred.html', prediction_text='No delay is predicted to happen. HAVE  
A NICE FLIGHT!!')  
elif m==1:  
    return render_template('pred.html', prediction_text='Delay in flight departure is predicted to  
happen')  
elif m==2:  
    return render_template('pred.html', prediction_text='Delay in both flight departure and  
arrival is predicted to happen')  
elif m==3:  
    return render_template('pred.html', prediction_text='Flight is predicted to get Diverted')  
elif m==4:
```

```
        return render_template('pred.html', prediction_text='Flight is predicted to get Cancelled!')
    else:
        return render_template('pred.html', prediction_text='output {}'.format(m))
```

```
if __name__ == "__main__":
    os.environ.setdefault('FLASK_ENV', 'development')
    app.run(debug=False)
```

7.3 Database Schema (if Applicable):

Database name: login

Collection name: users

Fields: Email, password and username.

The screenshot shows the MongoDB Compass interface for the **login.users** collection. The **Schema** tab is selected, displaying a filter bar with the query `{ field: 'value' }`. Below the filter, a message states: "This report is based on a sample of 3 documents." The schema details are as follows:

- _id**: Type `objectid`. A calendar icon indicates a time range from Sunday to Sunday, with the first document timestamped as `first: 2022-11-17 20:47:48`.
- email**: Type `string`. Examples shown are `m@m`, `em@s`, and `j@J`.
- password**: Type `string`. Examples shown are `pdwd`, `1`, and `1234`.
- username**: Type `string`. Examples shown are `d`, `anish`, and `asd`.

8. TESTING

8.1 Test Cases:

8.2 User Acceptance Testing:

9. RESULTS

9.1 Performance Metrics

Evaluation of Random Forest

```
In [53]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
pred=f.predict(x_test)
cm=confusion_matrix(y_test, pred)
plt.figure(figsize=(10,6))
sns.heatmap(cm, annot=True, cmap='winter', linewidths=0.3, linecolor='black', annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn', (TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Random Forest', (TP/(TP+FN)))
print('Testing Specificity for Random Forest', (TN/(TN+FP)))
print('Testing Precision for Random Forest', (TP/(TP+FP)))
print('Testing accuracy for Random Forest', accuracy_score(y_test, pred))

Testing Sensitivity for Random Forest 0.9360230547550432
Testing Specificity for Random Forest 0.8716577540106952
Testing Precision for Random Forest 0.9854368932038835
Testing accuracy for Random Forest 0.8368506493506493
```

```
In [54]: print(classification_report(y_test,pred))#RandomForest
```

	precision	recall	f1-score	support
0.0	0.86	0.96	0.91	1683
1.0	0.73	0.53	0.61	308
2.0	0.67	0.49	0.57	288
3.0	0.88	0.65	0.75	55
4.0	0.92	0.75	0.82	130
accuracy			0.84	2464
macro avg	0.81	0.68	0.73	2464
weighted avg	0.83	0.84	0.83	2464

10. ADVANTAGES & DISADVANTAGES:

Advantage:

- 1) With this model, we can easily simplify the extensive traffic at the airport and can prevent the major confusions over flight delays.
- 2) This can enable customer satisfaction and incomes of major airlines.
- 3) Accuracy is measured with the previous models and we have analyzed that this model is much more effective in every way.
- 4) The delay prediction model can make the concerned authorities be well prepared for any possible problem.
- 5) The model can easily be understood by a layman: the model is simple and effective.

Disadvantages:

1. This model needs to be more compact and flexible. The interoperability feature should be more enhanced.
2. The model can be automated instead of manually entering data from the user. Manually entering data is hectic work for the user.

11. CONCLUSION:

In the present world, the major components of any transportation system include passenger airline, cargo airline and air traffic control system. They all face difficulties due to some sort of miscommunication. Our model has been made with the motive of simplifying complex situations due to flight delays and increasing customer satisfaction. With delays being predicted before, the passengers can easily schedule their plans well before. Our model works with an accuracy of 84% and is considered as an efficient model.

12. FUTURE SCOPE:

The project can be extended to a wider range of airports. Current model only supports the data from 5 airports. If the dataset is extended by a vast quantity that has data from airports worldwide then the model can predict any flight delay across the globe. But to do so the complexity of power required will be much greater and the model needs to be trained better to have a higher speed and accuracy of computing results.

13. APPENDIX:

Source Code GitHub & Project Demo Link:

GitHub link: <https://github.com/IBM-EPBL/IBM-Project-34512-1660236682>

Demo video link: