RMK ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai, Gummidipoondi Taluk,
Thiruvallur District 601 206.

PROJECT

DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

DONE BY

TEAM ID: PNT2022TMID15779

NOTAM KEDARI (111719104107)

PEDDAMALLU SAI DIVYA (111719104115)

MONEEKHA.G (111719104096)

NITHYA SREE P.B (111719104106)

TABLE OF CONTENTS

INTRODUCTION

In the present world, the major components of any transportation system include passenger airline, cargo airline, and air traffic control system. With the passage of time, nations around the world have tried to evolve numerous techniques of improving the airline transportation system. This has brought drastic change in the airline operations. Flight delays occasionally cause inconvenience to the modern passengers. Every year approximately 20% of airline flights are cancelled or delayed, costing passengers more than 20 billion dollars in money and their time.

1.1 Project Overview

Average aircraft delay is regularly referred to as an indication of airport capacity. Flight delay is a prevailing problem in this world. It's very tough to explain the reason for a delay. A few factors responsible for the flight delays like runway construction to excessive traffic are rare, but bad weather seems to be a common cause. Some flights are delayed because of the reactionary delays, due to the late arrival of the previous flight. It hurts airports, airlines, and affects a company's marketing strategies as companies rely on customer loyalty to support their frequent flying programs.

1.2 Purpose

Using techniques such as Natural Language Processing, Naïve Bayes, and Support Vector Machine, researchers-built algorithms for analysis that helped them in extracting features in the model. Most of them focused on predicting overall flight delays. Our research concentrated mainly on predicting flight delays for a particular airport over a specific period of time. First, we used a regression model to examine the significance of each feature and then, a feature selection approach to examine the impact of feature combination. These two techniques determined the features to retain in the model. Instead of using the whole set, we sampled 5,000 records at a time to run through different machine learning models. The machine learning models implemented here were Random Forest classifier and Support Vector Machine (SVM) classifier. Further, we applied an approach called One-Hot-Encoder to create a variant of the model for evaluating potential prediction performance.

2. LITERATURE SURVEY

General:

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources 8 and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations, or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

2.1 Existing Problem

In the present world, the major components of any transportation system include passenger airline, cargo airline, and air traffic control system. With the passage of time, nations around the world have tried to evolve numerous techniques of improving the airline transportation system. This has brought drastic change in the airline operations. Flight delays occasionally cause inconvenience to the modern passengers. Every year approximately 20% of airline flights are cancelled or delayed, costing passengers more than 20 billion dollars in money and their time.

2.2 References

The prediction of short term delays (for the next hours or so) is already a largely explored field. Indeed, using information about weather conditions, airports congestion and current flight delays allows quite accurate predictions of future delays, as some parameters influencing them are known, even if thThe prediction of short term delays (for the next hours or so) is already a largely explored field. Indeed, using information about weather conditions, airports congestion and current flight delays allows quite accurate predictions of future delays, as some parameters influencing them are known, even if they still have a random

component). For example, the website FlightCaster exploit several sources of information (airports, airlines, weather and possibly historical data) to provide probabilities of being on-time, less than one hour late or more than one hour late, to travelers. However, this website is using the same estimations for all the flights when no short-term information is available.

A lot of researches have also been conducted on the management and propagation of flight delays, focused on traffic management systems. Mueller and Chatterji [1] tried to model the departure, en-route and arrival delays with Normal or Poisson distributions, that could possibly be taken into account in traffic management systems. Those rough models do not take into account any characteristic of the flights, but only give global trends. In another article focused on Ground Delay Programs improvement by Allan et al [2], are studied in details the meteorological conditions and their impact on on-ground and flight delays.

On a short-term perspective, an article by Zonglei et al [3] presents predictions of the overall traffic status on an airport (percentage of delayed flights), using decision trees and neural networks.

Finally, Tu & Ball tried to estimate in [4] the departure delay distribution by modeling the underlying mechanisms, with three components: a seasonal trend, a daily trend and a random residual, fitted using genetic algorithm. Their method seems quite expensive to compute and is not extensively tested. However it is close to the goal of my project, so we will try to compare our future results with the one presented in this article.

The main concern of the researchers and analysts is to predict the reasons for flight delays and for that they have put in their efforts on collecting data about flight and the weather.

[1] M. Abdel-Aty, C. Lee, Y. Bai, X. Li, and M. Michalak. Detecting periodic patterns of arrival delay. Journal of Air Transport Management, 13(6):355–361, Nov. 2007. ISSN 0969-6997.

[2] K. F. Abdelghany, S. S. Shah, S. Raina, and A. F. Abdelghany. A model for projecting flight delays during irregular operation conditions. Journal of Air Transport Management, 10(6):385–394, Nov. 2004. ISSN 0969-6997.

[3] S. AhmadBeygi, A. Cohn, Y. Guan, and P. Belobaba. Analysis of the potential for delay propagation in passen- ger airline networks. Journal of Air Transport Management, 14(5):221–236, Sept. 2008. ISSN 0969-6997.

[4] S. Ahmadbeygi, A. Cohn, and M. Lapp. Decreasing airline delay propagation by re-allocating scheduled slack. IIE Transactions (Institute of Industrial Engineers), 42(7):478–489, 2010.

[5] ANAC. Agˆencia Nacional de Aviac¸˜ao Civil. Technical report, http://www.anac.gov.br/, 2017

[6] C. Ariyawansa and A. Aponso. Review on state of art data mining and machine learning techniques for intelligent Airport systems. In Proceedings of 2016 International Conference on Information Management, ICIM 2016, pages 134–138, 2016.

[7] F. Azadian, A. E. Murat, and R. B. Chinnam. Dynamic routing of time-sensitive air cargo using real-time information. Transportation Research Part E: Logistics and Transportation Review, 48(1):355–372, Jan. 2012. ISSN 1366-5545.

[8] E. Balaban, I. Roychoudhury, L. Spirkovska, S. Sankararaman, C. Kulkarni, and T. Arnon. Dynamic routing of aircraft in the presence of adverse weather using a POMDP framework. In 17th AIAA Aviation Technology, Integration, and Operations Conference, 2017, 2017.

[9] P. Balakrishna, R. Ganesan, and L. Sherry. Airport taxi-out prediction using approximate dynamic programming: Intelligence-based paradigm. Transportation Research Record, (2052):54–61, 2008.

[10] P. Balakrishna, R. Ganesan, L. Sherry, and B. S. Levy. Estimating Taxi-out times with a reinforcement learning algorithm. In 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, pages 3.D.3–1–3.D.3–12, Oct. 2008.

[11] P. Balakrishna, R. Ganesan, and L. Sherry. Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa

Bay departures. Transportation Research Part C: Emerging Technologies, 18(6):950–962, Dec. 2010. ISSN 0968-090X.

2.3 Problem Statement Definition

As explained, the goal of this project is to estimate the probability of any flight to be more than x minutes late, for any x being the difference between 5 the total connection time and the time to go to the departure gate. Moreover, as we would like to give this information to the customer during the search and reservation process, the model will have to give long-term predictions, up to several months forward, and will not take into account short-term effects, like current weather or traffic situation. This model will be based on the unique public large dataset of flight delays, provided by the Bureau of Transportation Statistics of the United States Department of Transportation.

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 BRAINSTROM AND IDEA PRIORITIZATION

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

(9 10 minutes to prepare
S: 1 hour to collaborate
.:, 2•8 people recommended

Share template feedback

C)

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

<!) 10mlnutes

[!] Team gathering
Define who should participate In the session and send an

Invite. Share relevant information or pre-work &haad.

m Setthegoal
Think about the problem you'll be focusing on solving in the brainstorming session.

c **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article

0

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

(!) Sminutes

r w accurat.ty we can rmlne the flight delay from our model?

1 • %

Key rules of brainstorming
To run en smooth and productive session

@ Stay in topic. - - Encourage wild ideas.
@ Defer Judgment listen to others.
® Go for volume. ® If possible, be vtsual.

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example →

Step-2: Brainstorm, Idea Listing and Grouping

0

Brainstorm

Wrtte down any Ideas that come to mind that address your problem statement

(!)10minutu

" '
'IJ' J[,... ltcl,st<i<Vnoto>
¥1dhll1heptftdI1 lo
1"vtd,ln,i10MM1n..............r,v1

P Sai Divya

Collecting

data regarding v.eather

u..
0-!clson
Tree Tor
Prection

Categorize Data

\Vetm;rapng

Competitive analys-s

Notam Kedari

Manage

c>plorP.reo:1l

Minimum connection Time

11meupda1ei.
from ilr1ous
web les

S1a11st1cal Mode,s

Onsrte

to handle technlcc1l glitches

Man11ge Minimum connection Time

Nithya Sree P B

collecting data about Flight take off

allowtoflter flights by layover Duratio'l

Model Evaluation

Training and Testing the Data

Radial Basis Neural Network

Moneekha G

Freo en1
rem,nder1to
i:,ns""9e
,eg,1,0-'9
De rture

Data Vii;;ualisation

Preprocessing t!'lec'o111

Speed up the process of Security Checkup

Ad:>rOo'hl!O'
n,w1,1al:on
•.uMunf!
1.. ,,us,lto
•nc!hf'ft«mrr•

e

Group ideas

Take turns sharing your Ideas white clustering slmilar or related notes as you go. Once all sticky notes have been grouped, g!\le each cluster a sentence41ke label. If a cluster Is bigger than six sticky notes. try and see ff you and break It up Into smaller sub-groups.

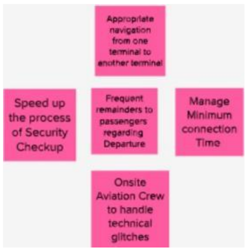
<!)20nWnutH

Data Analysis

Data Vlsuallsatlon

Model Evaluation

Training and Testing the Data



Comp@tlttve analySls

Statistical ModelS

Radial Basis Neural Network

Use Decision Tree for PredlctJon

Categorize o…

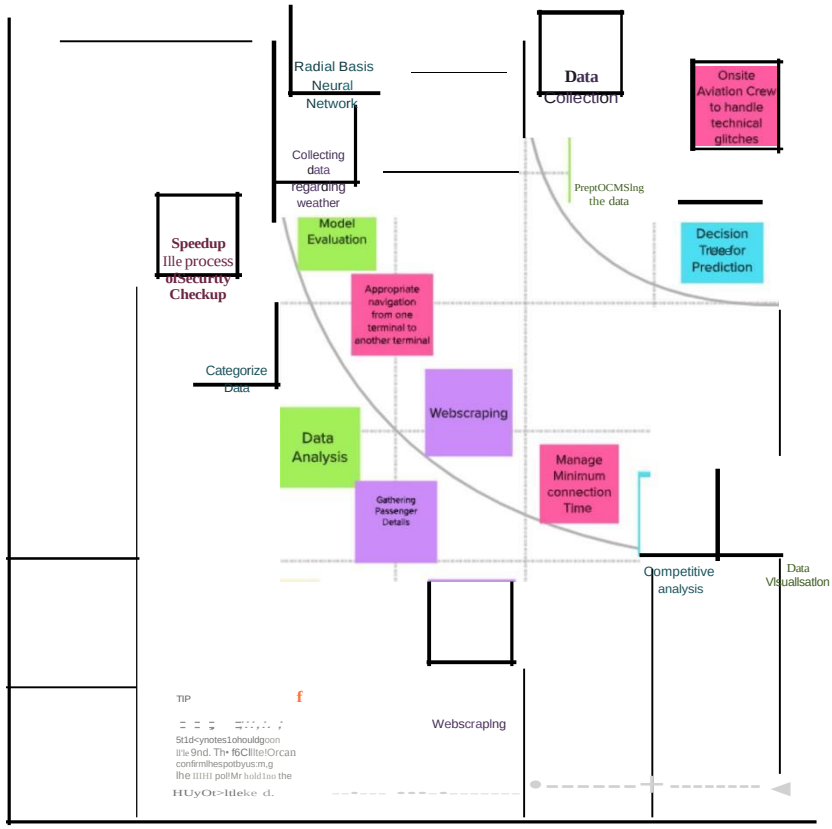
IDEA PRIORITIZATION

0

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which idea5 are important and which are feasible.

(:) 20 minutes



0

Importance

C)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find It helpful.

Quick add-ons

rn :: : n;:::,Ik tothemural with stakeholders to keep them in lhe loop about the outcomes of 1he session.

rn Export... """"
Export a copy of lhe mural as II PNG or PDF to atlach to emails, include in slides, or save in your drive.

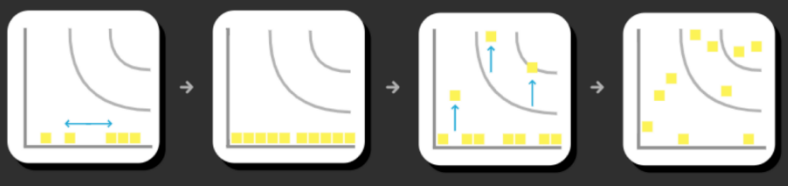
Keep moving f01Ward

Strategy blueprint
Define tha compooents of• new Idea or strategy.
Open the template

Customer experience journey map
UndefStand customer needs. motivations. and obstacles for an experience.
Open the template

\2:V Strength weaknesse opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template

Share template fee



3.3 Proposed Solution:

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a Flight delay prediction model using machine Learning.
2.	Idea / Solution description	We use deep neural network and compare decision tree classifier with logistic regression.
3.	Novelty / Uniqueness	Notify users about the delay of more than 10 minutes via a notification through the application developed.
4.	Social Impact / Customer Satisfaction	optimize flight operations and minimize delays.
5.	Business Model (Revenue Model)	**With the proposed model**: Identify delay and schedule our plans accordingly and know about air traffic. **without the proposed model**: flight delays lead to large economic and environmental losses.
6.	Scalability of the Solution	The proposed solution can be used to identify delays based on the air traffic.

4. REQUIREMENT ANALYSIS

General:

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
    A. Hardware requirements
    B. Software requirements

4.1 Functional Requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists 18 requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

4.2 Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

Environmental Requirements:

1. Software Requirements:

    Operating System : Windows

    Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor	:Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

Software Description:

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system "Conda". The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open-source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the conda build command and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab

- Jupyter Notebook
- QtConsole
- Spyder
- Glueviz
- Orange
- Rstudio
- Visual Studio Code

Conda:

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-languages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

The Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Notebook document:

Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both humanreadable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

Jupyter Notebook App:

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App

can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

Kernel:

A notebook kernel is a “computational engine” that executes the code contained in a Notebook document. The IPython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels). When you open a Notebook document, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

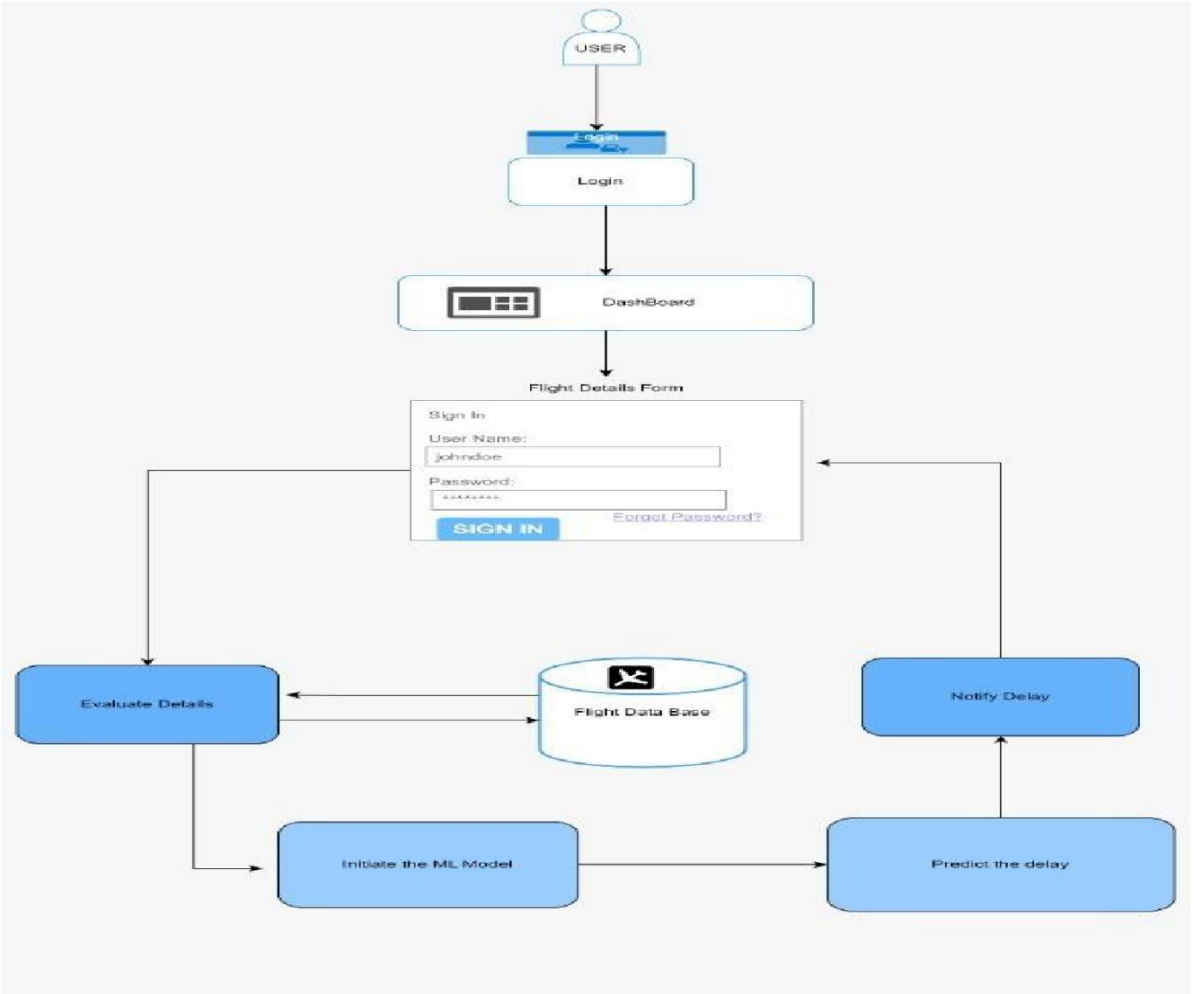
Notebook Dashboard:

The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can login at any time	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can Fill the flight details	High	Sprint-1
	Dashboard	USN-5	As a user,i can fill the flight details for which i want to get the prediction	I can get the delay prediction	High	Sprint-2
Customer (Web user)	Login	USN-6	As a user, I can login to my account and enter the web page through Email& password.	I can access the Web page	High	Sprint-1
Administrator	Access	USN-7	As a administrator, I can access and change the database and model	I can update the database and model	High	Sprint-2
	Modify	USN-8	As a administrator, I can edit the webpage based on the feedbacks	I can customize the web page	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection andPre-processing	USN-1	As a user, I can't interact anything. Waiting is user's task. User can listen the relationship exist between the various attributes of data by presentation of developer	2	high	Notam Kedari Peddamallu Sai Divya
Sprint-1	Model Building	USN-2	As a user, I can predict flight delay by various developed ML models by console	1	high	Notam Kedari Peddamallu Sai Divya
Sprint-2	Model Evaluation	USN-3	As a user, I can predict flight delay by best Model in various developed ML model by console	2	high	Notam Kedari Peddamallu Sai Divya
Sprint-2	Model Deployment onIBM Cloud using IBM Watson	USN-4	As a user, I can use the model by requesting the deployed model on Cloud	1	Medium	Notam Kedari Peddamallu Sai Divya
Sprint-2	Basic user interaction Dashboard	USN-5	As a user, I can use the model or prediction from model by interacting with dashboard	2	high	Notam Kedari Nithya Sree PB
Sprint-3	Improved Dashboardand GUI	USN-6	As a user, I can use the model or prediction from model by interacting with improved dashboard	1	Medium	Peddamallu Sai Divya Moneekha G
Sprint-3	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Moneekha G Nithya Sree PB
Sprint-3	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Moneekha G Nithya Sree PB
Sprint-3	Login	USN-8	As a user, I can log into the application by entering email & password and I can register .login to the application through Gmail	2	Medium	Notam Kedari Peddamallu Sai Divya
Sprint-4	Raise query/complaintand give feedback	USN-9	As a user, I can raise complaint or query and give feedback	1	Medium	Notam Kedari Peddamallu Sai Divya
Sprint-4	Improve overall webapp	USN-10	As a user, I can user revised and improved version of web application	1	High	Notam Kedari Peddamallu Sai Divya Moneekha G Nithya Sree PB

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

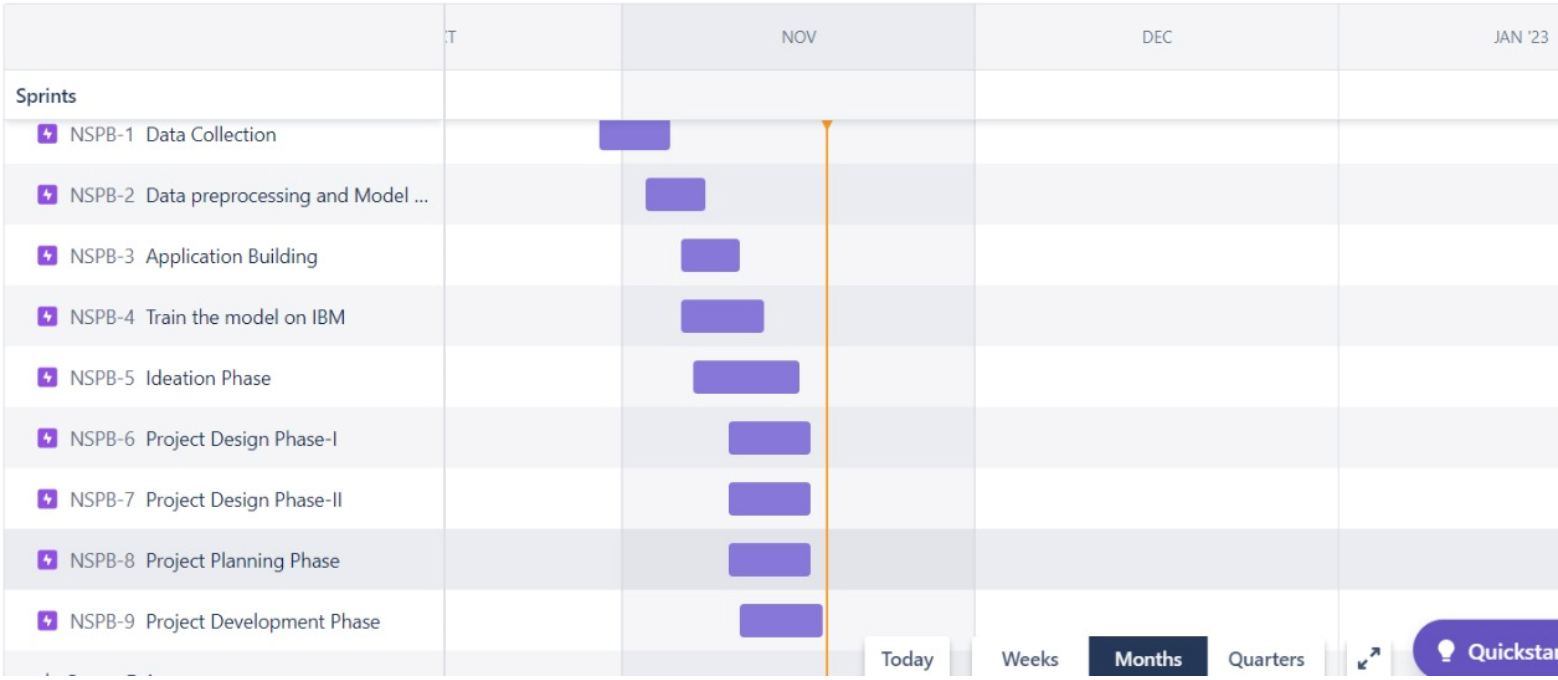
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

7. CODING & SOLUTIONING



8. TESTING

8.1 Test Cases Testing Levels: -

All major activities of various testing level are described below.

1.Unit Testing

2.Integration Testing

3.Functional Testing

4.System Testing

5.White box Testing

6. Black Box Testing

1. Unit Testing: -

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

2. Integration Testing: -

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3.Functional Testing: -

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

4.System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An

example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.2 User Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

8.2 User Acceptance Testing

S. No	Test Cases	Yes/ No
1.	Responds in manually drafted rules	yes
2.	Manages multiple users	yes
3.	Users can enter details	yes
4.	Run the Test cases	yes
5.	Learns from real interactions	no
6.	Training via historical data	no
7.	Has decision-making skills	no

9. RESULTS

9.1 Performance Metrics Comparing Algorithm with prediction in the form of best accuracy result:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 2 different algorithms are compared:

Logistic Regression

Random Forest

- dimensions of new features in a numpy array called ‘n’ and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.

- So, the predicted target value comes out to be 0. Finally, to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

Sensitivity:

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1. Let's try and understand this with the model used for predicting whether a person is suffering from 36 the disease. Sensitivity is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease. In other words, the person who is unhealthy actually got predicted as unhealthy.

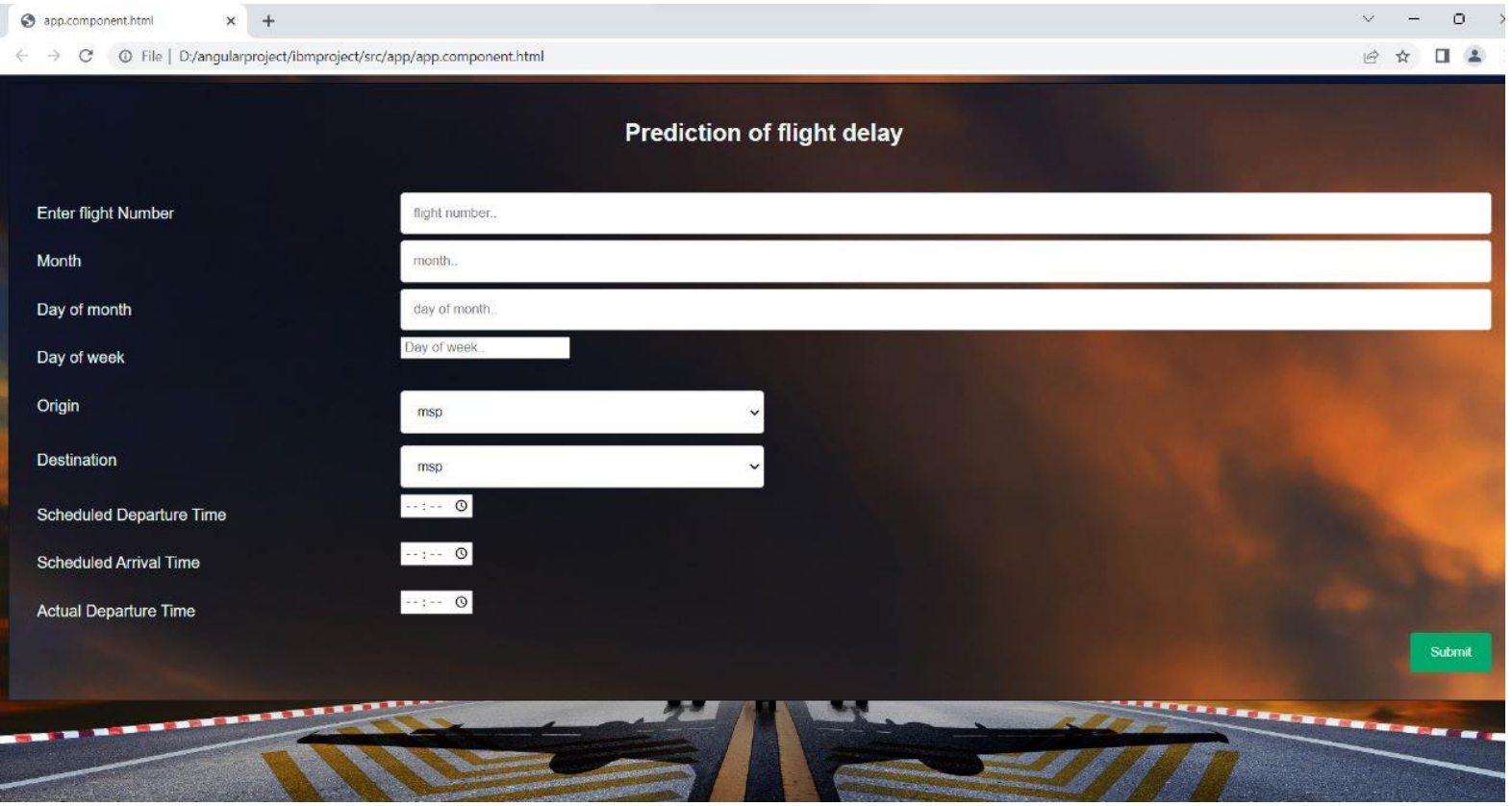
Mathematically, sensitivity can be calculated as the following:

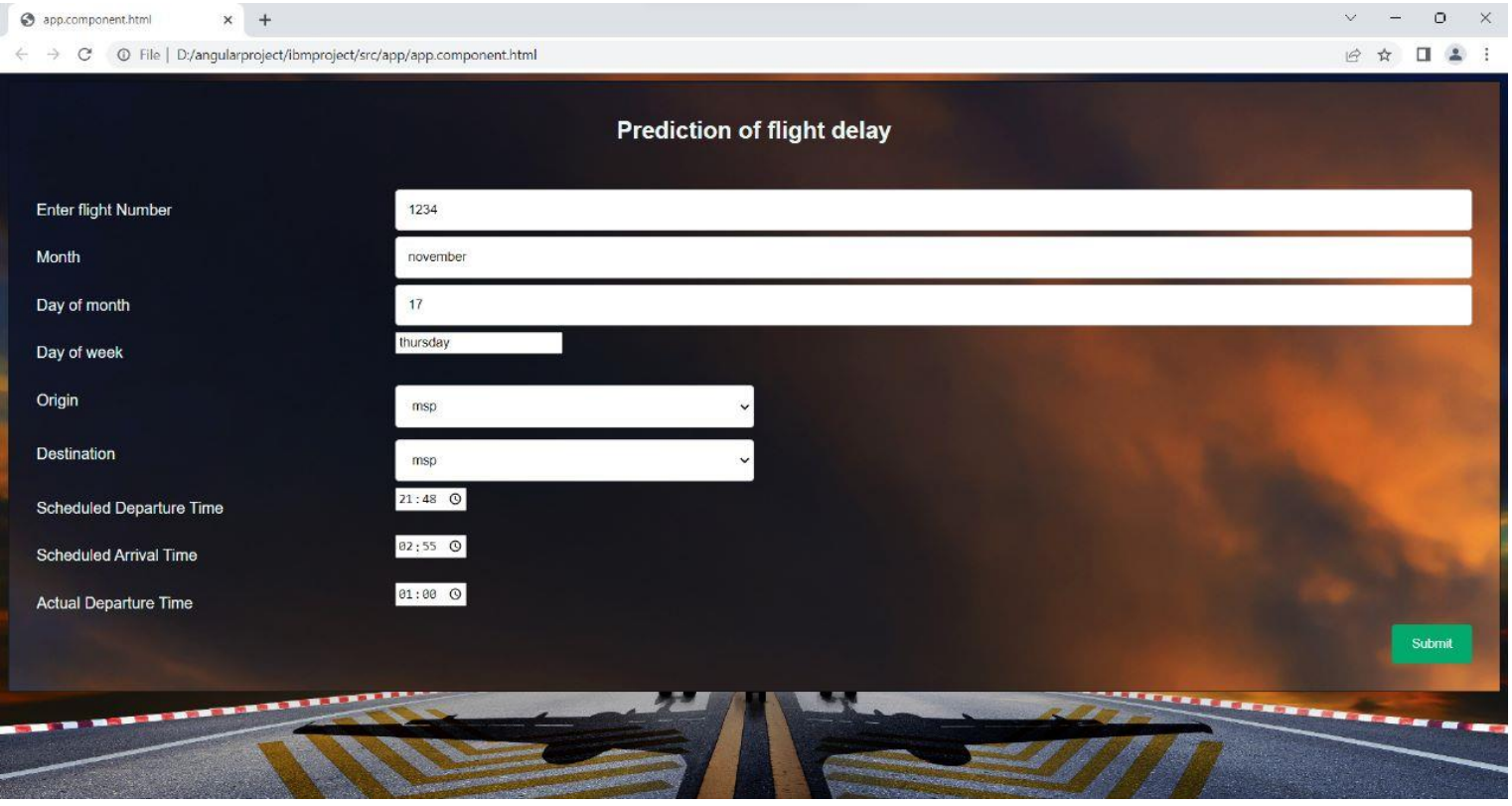
Sensitivity = (True Positive) / (True Positive + False Negative)

The following is the details in relation to True Positive and False Negative used in the above equation.

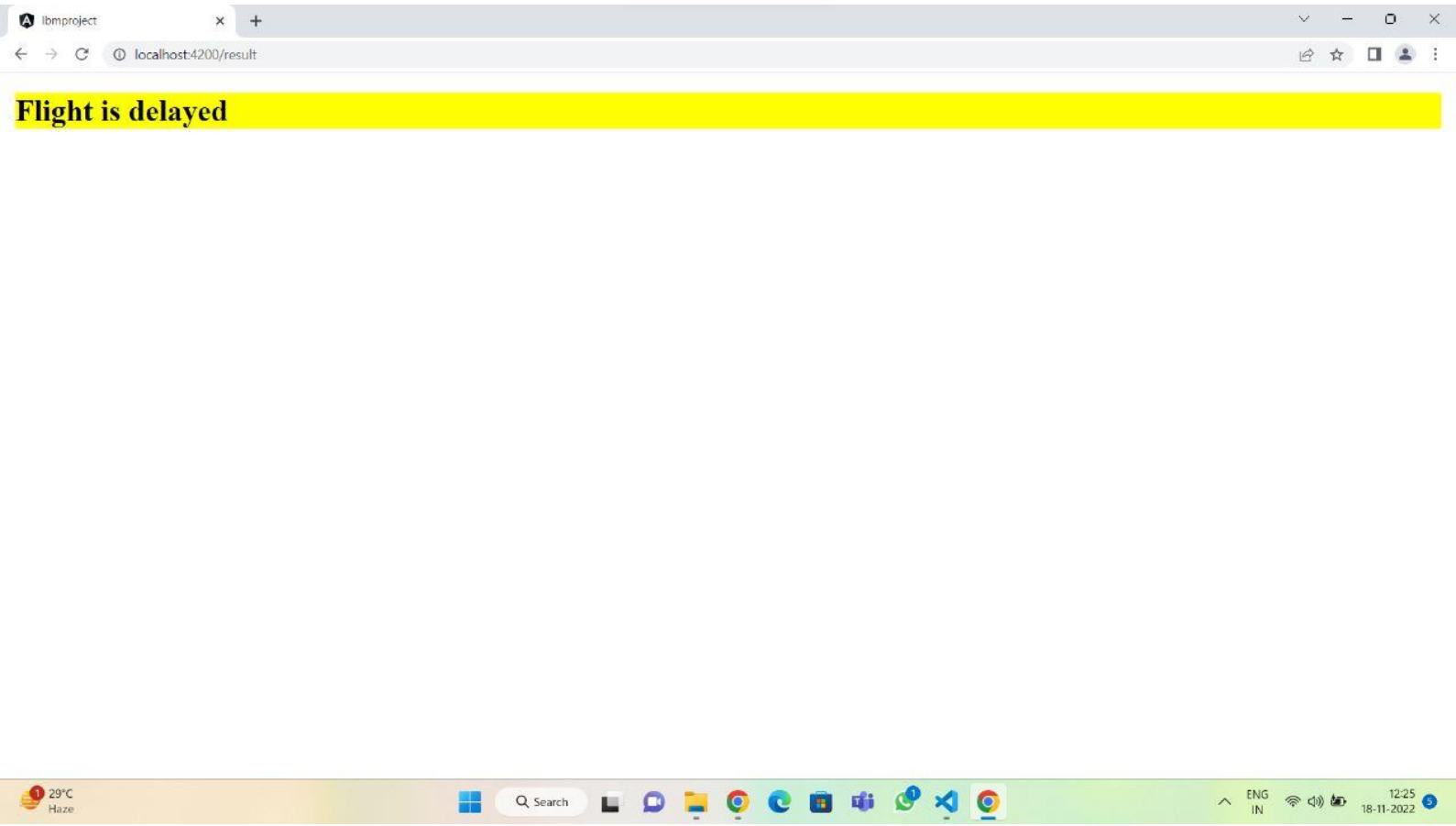
- True Positive = Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.

- False Negative = Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening


app.component.html
File | D:/angularproject/ibmproject/src/app/app.component.html
Prediction of flight delay
Enter flight Number
flight number..
Month
month..
Day of month
day of month..
Day of week
Day of week..
Origin
msp
Destination
msp
Scheduled Departure Time
Scheduled Arrival Time
Actual Departure Time
Submit


app.component.html
File | D:/angularproject/ibmproject/src/app/app.component.html
Prediction of flight delay
Enter flight Number
1234
Month
november
Day of month
17
Day of week
thursday
Origin
msp
Destination
msp
Scheduled Departure Time
21:48
Scheduled Arrival Time
02:55
Actual Departure Time
01:00
Submit

RESULT


Ibmproject
localhost:4200/result
Flight is delayed
29°C
Haze
Search
ENG
IN
12:25
18-11-2022

10. ADVANTAGES

No queueing in responses

Latest data and requirements are updated

Updated to the latest details

Easy to enter data and get results

DISADVANTAGES

Data cleaning is more challenging

Frequent Updating

Miscalculation of data

Limited entry resources

Technology and Hacking

11. CONCLUSION

We have explored different prediction models and various evaluation method. By measuring the performance of the models using real data, we have seen interesting results on the predictability of the delays. The best delay prediction method appeared to be the most specific one, which takes into account all the combination of categorical parameters and a condition on the arrival hour.

The performances of the models were challenging to evaluate, due to the variety of measures used, and the different parameterizations adapted to them. However the predictions obtained appeared to be better than the one seen in the literature or used by Flight Caster (their long-term prediction being only based on the empirical cumulative distribution).

The kernel density estimation method was a very interesting method to learn and manipulate. Being a data-centered method, it can be used each time we want to reconstruct a probabilistic model from some observations.

12. FUTURE WORK

The models presented could be improved in several ways. First of all, more complex density estimation methods could be used, using for example variable bandwidths, dynamically adapting the smoothing degree to the local density.

The models could also be refined, in order to give specific predictions for some period of time, like holidays and days off, during which a lot of people are travelling. We can also imagine building an expert system, which will use a combination of the different models we have seen, to select the best one for different use cases.

We could also model the phenomenon more precisely instead of looking only at the distribution of past data. We can for instance build separate models per time period, per type of aircraft, per airline, per region, and then grouping them into a general model. This way, we may be able to predict the delays of a new flight, without needing several months of data to build a prediction model.

Another step forward would be to generalize the model to flights of the entire world, or at least to exploit more data sources, to build more complete predictions.

Finally, the most interesting step would be to integrate such a model into a flight booking tool, to provide the delay prediction to future passengers, even if this would require a strong confidence in the information provided, considering the possible impact in terms of reservations.

13. ABSTRACT

Source Code

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
 h2 {text-align: center;}
* {
 box-sizing: border-box;
}
body{
 font-family: Arial, Helvetica, sans-serif;
 background-image:
url(https://www.washingtonpost.com/wpapps/imrs.php?src=https://arc-
anglerfish-washpost-
prodwashpost.s3.amazonaws.com/public/PBKJ5C6KJJC75BO46RZEWUGL6
A.jpg&w
=860);
 background-size: cover;
 background-attachment: fixed;
}
input[type=text], select, textarea {
 width: 100%;
 padding: 12px;
 border: 1px solid #ccc;
 border-radius: 4px;
 resize: vertical;
```

```
}
label {
 padding: 12px 12px 12px 0;
 display: inline-block;
}
input[type=submit] {
 background-color: #04AA6D;
 color:white;
 padding: 12px 20px;
 border: none;
 border-radius: 4px;
 cursor: pointer;
 float: right;
}
input[type=submit]:hover {
 background-color: #45a049;
}
.container {
 border-radius: 5px;
 background-color:transparent;
 padding: 20px;
}
.col-25 {
 float: left;
 width: 25%;
 margin-top: 6px;
}
```

```
.col-75 {
 float: left;
 width: 75%;
 margin-top: 6px;
}
/* Clear floats after the columns */
.row:after {
 content: "";
 display: table;
 clear: both;
}
</style>
</head>
<body>
<h2 style="color: #ebf7f3">Prediction of flight delay</h2>
<div class="container">
 <form action="/action_page.php">
 <div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Enter flight Number</label>
 </div>
 <div class="col-75">
 <input type="text" id="fname" numbers="flight number" placeholder="flight
number..">
 </div>
 </div>
 <div class="row">
```

```
 <div class="col-25">
 <label style="color: #ebf7f3">Month</label>
 </div>
 <div class="col-75">
 <input type="text" id="" name="month" placeholder="month..">
 </div>
 </div>
 <div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Day of month</label>
 </div>
 <div class="col-75">
 <input type="text" id="" name="month" placeholder="day of month..">
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Day of week</label>
 </div>
 <div class="col-75">
 <input type="calender" id="fname" numbers="Day of week"
placeholder="Day of week..">
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Origin</label>
```

```
 </div>
 <div class="col-75">
 </div>
<div class="col-25">
 <select id="country" name="origin">
 <option value="region">msp</option>
 <option value="region">sea</option>
 <option value="region">dtw</option>
 <option value="region">jfk</option>
 <option value="region">alt</option>
 </select>
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Destination</label>
 </div>
 <div class="col-75">
 </div>
<div class="col-25">
 <select id="region" name="origin">
 <option value="region">msp</option>
 <option value="region">sea</option>
 <option value="region">dtw</option>
 <option value="region">jfk</option>
 <option value="region">alt</option>
 </select>
```

```
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Scheduled Departure Time</label>
 </div>
 <div class="col-75">
 <input type="time" id="fname" numbers="predict" placeholder="scheduled
Depatureb Time..">
 </div>
 </div>
 <div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3"> Scheduled Arrival Time</label>
 </div>
 <div class="col-75">
 <input type="time" id="fname" numbers="predict" placeholder="Arrival
Depatureb Time..">
 </div>
 </div>
<div class="row">
 <div class="col-25">
 <label style="color: #ebf7f3">Actual Departure Time</label>
 </div>
 <div class="col-75">
 <input type="time" id="fname" numbers="predict" placeholder="..">
 </div>
```

```
 </div>
<div class="row">
 <input type="submit" value="Submit">
 </div>

 </form>
</div>
</body>
</html>
```

GITHUB AND PROJECT DEMO LINK :

GITHUB:

https://github.com/IBM-EPBL/IBM-Project-25296-1659957840

DEMO LINK:

https://drive.google.com/file/d/1grCvlOe-Ugwm-uFi38NetbB5exgCOooh/view