DEVELOPING A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

MURUGANANTHI K	953419104034
DIVYA P	953419104019
PAVITHRA R	953419104039
VISHWA ABIRAMI	953419104056

TEAM ID:PNT2022TMID50880

In partial fulfillment for the award of degree of
Bachelor of Engineering (B.E) in
COMPUTER SCIENCE AND ENGINEERING

VV COLLEGE OF ENGINEERING,THISAYANVILAI

THIRUNELVELI-627117

CHAPTER-1

INTRODUCTION

1.1 Project Overview

- Fight delay prediction is fundamental to establish the more efficient airline business.Flight delay is a significant Fight delay prediction is fundamental to establish the more efficient airline business.Flight delay is a significant problem that negatively impacts the aviation industry and costs billions of dollars each year.
- Most existing studies investigated this issue using various methods based on applying machine learning methods to predict the flight delay.
- However, due to the highly dynamic environments of the aviation industry, relying only on single route of airport may not be sufficient and applicable to forecast the future of flights.

1.2 Purpose

- The purpose of this project is to analyze a broader scope of factors which may potentially influence the flight delay it compares several machine learning-based models in designed generalized flight delay prediction tasks.
- In this project we have used flight delay dataset from US Department of Transportation (DOT) to predict flight delays and also supervised learning algorithms to predict flight departure delay and then model evaluation is done to get best model and our model can identify which features were more important when predicting flight delays.

CHAPTER-2

LITERATURE SURVEY

2.1 Existing problem

- The main issues associated with flight delay prediction are known and arranged in taxonomy. It includes the problem that causes the flight delay, the range of institution it affects, and ways that of handling flight delay prediction downside.
- It considers flight domain options, like problem and scope. Major problem which causes delay in flights can be delay propagation, delay caused on the departure point or the root of the flight, and cancellation of flights.
- These problems cannot be eliminated forever, but a delay prediction tool will allow the operator and the administrators to take the concerned actions for smooth operation.
- This problem that is causes delay affects Airline, Airport and the enroute airspace which are independent entities which works in synchronization and hence delay in flight causes issues in all the sectors.
- Various methods that can be used to develop a system which predicts the delay in flights can be Machine Learning, Probabilistic models, Statistical analysis or Network Representations.

2.2 References

[1] We chose the “Airlines Delay” data fromwww.kaggle.com/datasets.

[2] 2018 U.S. Airlines Delay

Analysishttp://www.milantomin.com/2018-usairlinesdelay-analysis/.

[3] Taxi in/out time means the time when a flight wheel was on/off to the time the flight gate in/on time. See Aviation System Performance Metrics:

https://aspmhelp.faa.gov/index.php/ASPM_Taxi_Times:_Definitions_of_Variables.

[4] statisticsbyjim.com/regression/choosingregression-analysis/.

[5] towardsdatascience.com/the-poissondistribution-and-poisson-process-explained4e2cb17d459.

[6] Fox news reporter Rick Seaney article “Do flights ever leave early? And 4 other common travel questions”, https://www.foxnews.com/travel/doflights-everleaveearly-and-4-other-commontravelquestions.

[7] https://developers.amadeus.com/flight-delaypredictionmachine.

2.3 Problem Statement Definition

Flight delays are quite frequent (19% of the US domestic flights arrive more than 15 minutes late), and are a major source of frustration and cost for the passengers.

As we will see, some flights are more frequently delayed than others, and there is an interest in providing this information to travellers.

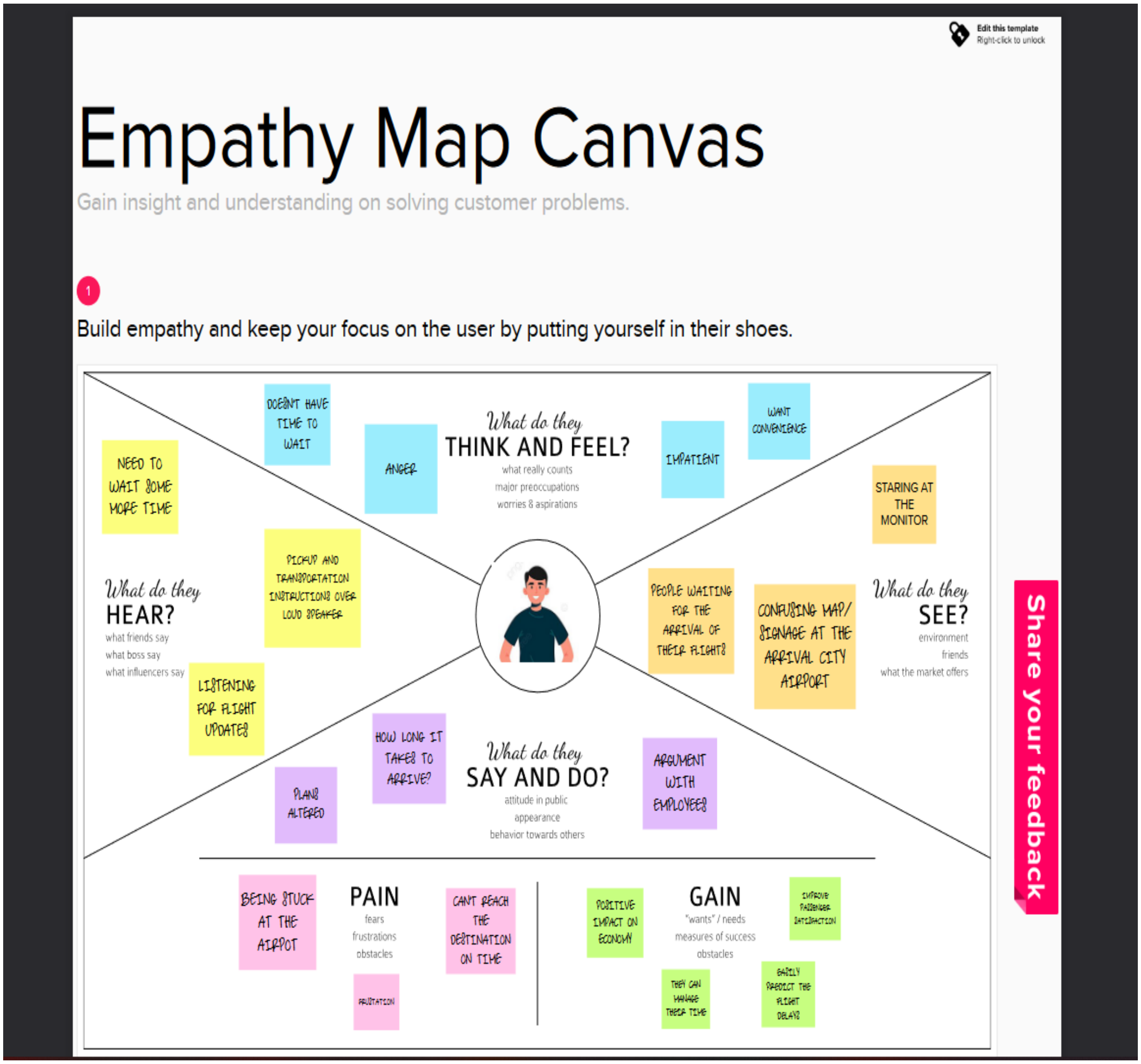
Flight prediction is crucial during the decision-making process for all players of commercial aviation.

Moreover, the development of accurate prediction models for flight delays became cumbersome due to the complexity of air transportation system, the number of methods for prediction, and the deluge of flight data. Based on data, we would like to analyse what are the major cause for flight delays and assign a probability on whether a particular flight will be delayed

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3. Empathy Map Canvas :



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL? what really counts major preoccupations worries & aspirations

What do they HEAR? what friends say what boss say what influencers say

What do they SEE? environment friends what the market offers

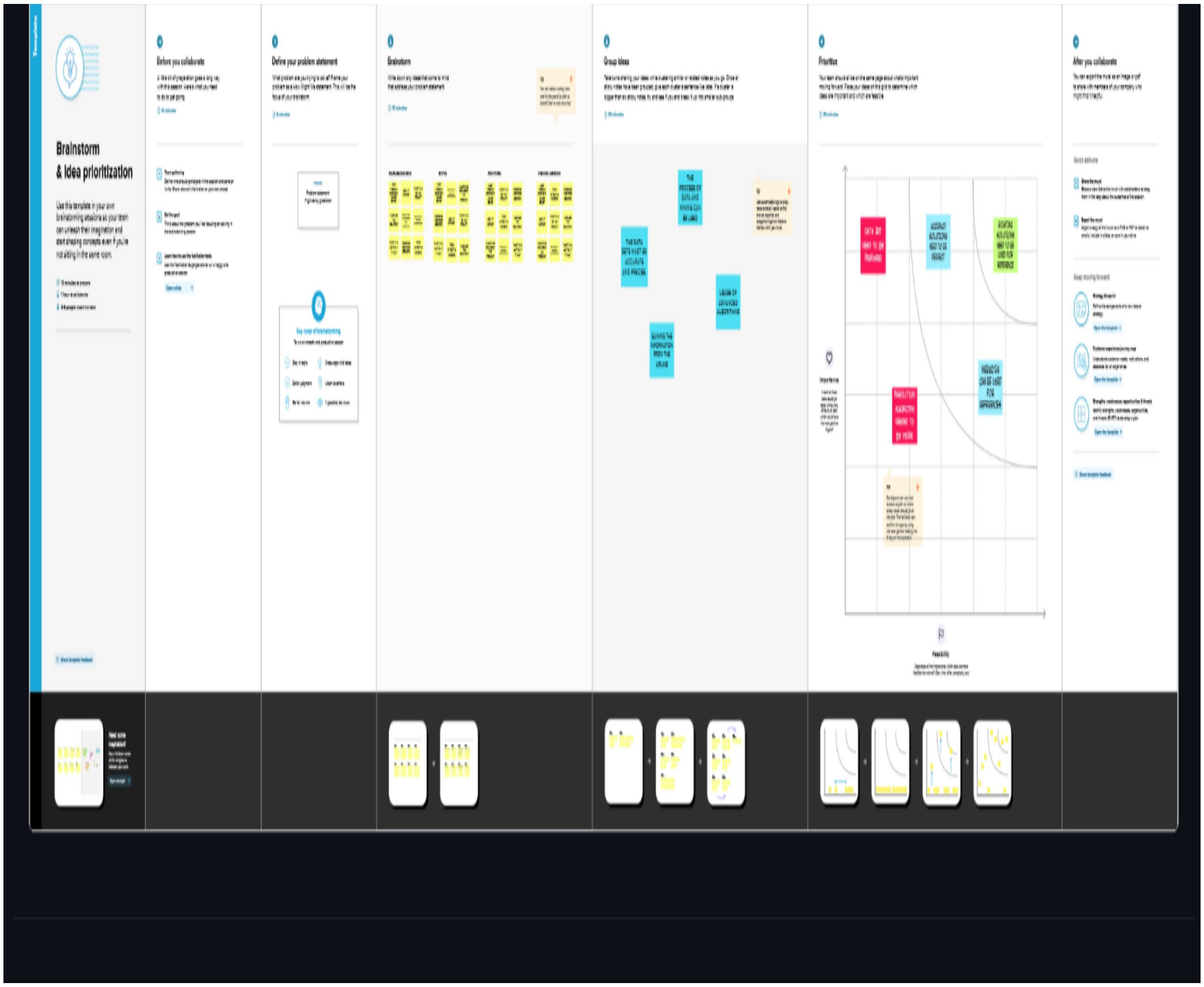
What do they SAY AND DO? attitude in public appearance behavior towards others

PAIN fears frustrations obstacles

GAIN "wants" / needs measures of success obstacles

DOESN'T HAVE TIME TO WAIT
ANGER
IMPATIENT
WANT CONVENIENCE
NEED TO WAIT SOME MORE TIME
PICKUP AND TRANSPORTATION INSTRUCTIONS OVER LOUD SPEAKER
LISTENING FOR FLIGHT UPDATES
STARING AT THE MONITOR
PEOPLE WAITING FOR THE ARRIVAL OF THEIR FLIGHTS
CONFUSING MAP/ SIGNAGE AT THE ARRIVAL CITY AIRPORT
PLANS ALTERED
HOW LONG IT TAKES TO ARRIVE?
ARGUMENT WITH EMPLOYEES
BEING STUCK AT THE AIRPOT
FRUSTATION
CAN'T REACH THE DESTINATION ON TIME
POSITIVE IMPACT ON ECONOMY
IMPROVE PASSENGER SATISFACTION
THEY CAN MANAGE THEIR TIME
EASILY PREDICT THE FLIGHT DELAYS
Share your feedback

3.2 Ideation & Brainstorming



Brainstorm
& Idea prioritization

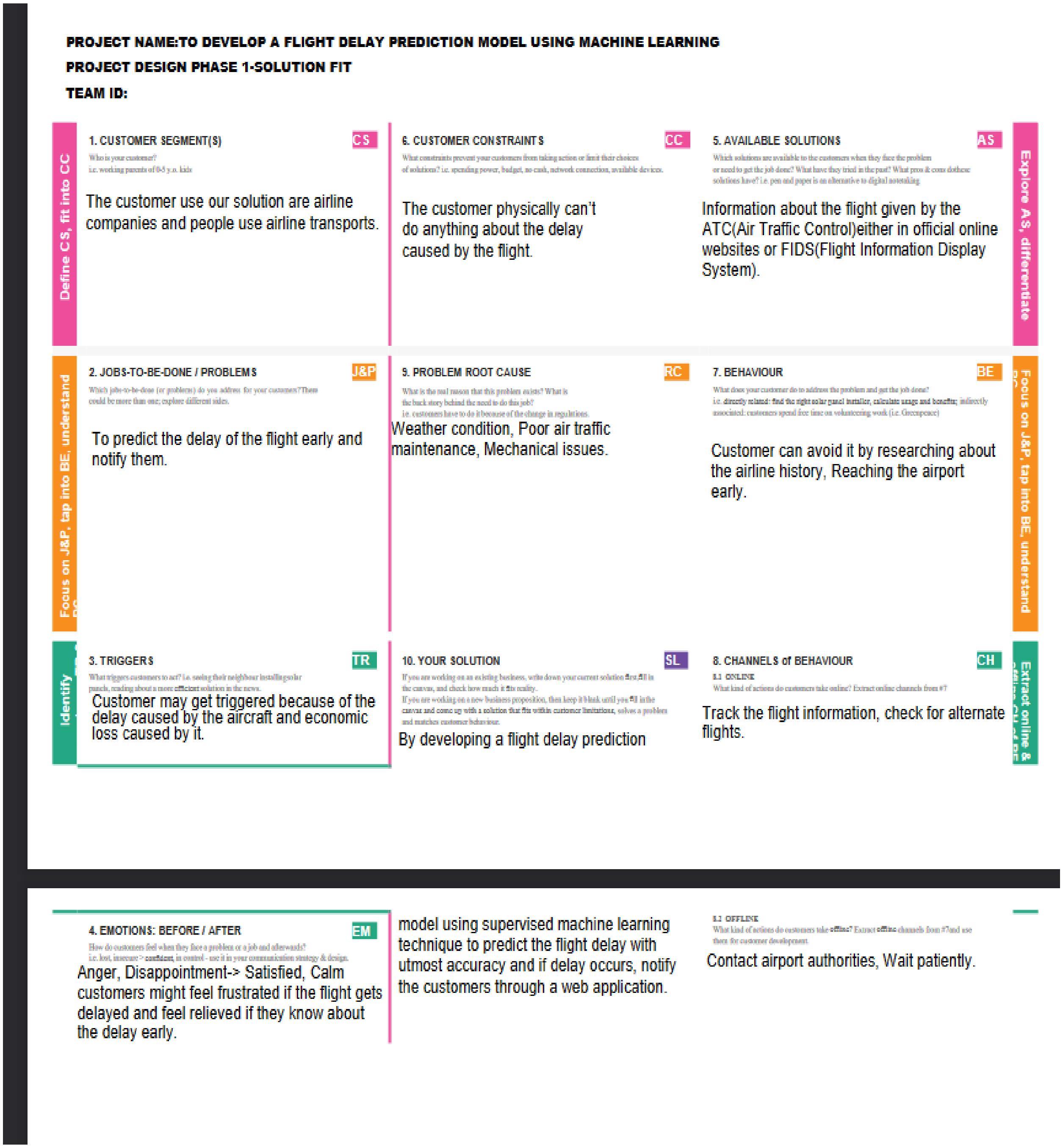
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• To predict flight delays using ML algorithm.
2.	Idea / Solution description	• Predicting flight delays using algorithms such as Random Forest, Logistic Regression, Decision Tree Support Vector Machine. • A user will be notified of the booked flight's location frequently. • Giving an accurate delay prediction will help in better customer service. • Cancellations will also be notified. • Multiple metrics like arrival/ departure delays, delays based on geographic areas are considered, making this solution more precise.
3.	Novelty / Uniqueness	• Frequent updates on the flight's location and accurate prediction of the delays. • Gives the status of different airports too.
4.	Social Impact / Customer Satisfaction	• Proper planning of trips. • Reduction of mental pressure and stress. • Prior information helps in avoiding loggerheads with other people.

		• Reduction of business losses.
5.	Business Model (Revenue Model)	• This model can be used by all the people who travel via flights and the app can be accessed through any device. • The existing solutions do not give frequent updates to the customer directly. • The ML algorithms to be used have accuracy between 87% - 91%.
6.	Scalability of the Solution	• The scalability of the solution is expanded for travellers all over the world, irrespective of their purpose for travelling. • This app can help customers to get updates of the flight of any part of the flight. • This is also beneficial for all the airline authorities by reducing complaints and increasing customer satisfaction

3.4 Problem Solution fit



PROJECT NAME:TO DEVELOP A FLIGHT DELAY PREDICTION MODEL USING MACHINE LEARNING

PROJECT DESIGN PHASE 1-SOLUTION FIT

TEAM ID:

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The customer use our solution are airline companies and people use airline transports.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

The customer physically can't do anything about the delay caused by the flight.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons dothese solutions have? i.e. pen and paper is an alternative to digital notetaking

Information about the flight given by the ATC(Air Traffic Control)either in official online websites or FIDS(Flight Information Display System).

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

To predict the delay of the flight early and notify them.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Weather condition, Poor air traffic maintenance, Mechanical issues.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Customer can avoid it by researching about the airline history, Reaching the airport early.

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Customer may get triggered because of the delay caused by the aircraft and economic loss caused by it.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Anger, Disappointment-> Satisfied, Calm customers might feel frustrated if the flight gets delayed and feel relieved if they know about the delay early.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

By developing a flight delay prediction model using supervised machine learning technique to predict the flight delay with utmost accuracy and if delay occurs, notify the customers through a web application.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Track the flight information, check for alternate flights.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7and use them for customer development.

Contact airport authorities, Wait patiently.

Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify
Extract online & offline CH of BE

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional requirement :

Data-Driven Probabilistic Flight Delay Predictions
In this section, we obtain probabilistic flflight delay predictions using two machine learning algorithms, Mixture Density Networks and Random Forests Regression.

Data Description
For this analysis, flight schedules available at Rotterdam The Hague Airport (RTM) between 1 January 2017 and 29 February 2020 are considered. In total, 17,365 departing and 17,336 arriving flights are considered. These flights arrive from and depart to 42 airports across Europe and North Africa. The shortest route included is to London City Airport(LCY), and the longest to Tenerife South Airport (TFS), with an average of 1300 km. shows a map indicating all airports to or from which flights depart or arrive. The delay distribution of these flight. The departing flights have an average absolute delay of 17.8 min with a standard deviation of 25.1 min, and the arriving flights have an average absolute delay of 15.4 min with a standard deviation of 26.4 min. Here, the delay is considered to be the positive or negative time difference from the schedule.

Weather Dataset:-
We also consider the weather conditions, such as the temperature, pressure and wind speed, measured at the origin/destination airport of all flights arriving/departing at RTM in the period 2017–2020. Measurements are available every 30 min.

Feature Selection :-
Feature selection is performed using the Pearson Correlation Coeffificient .The correlation between any two features and the correlation between the features and the target (the flight delay) are calculated for a given training set. The features are selected as follows: for any two features are correlated by more than the threshold value of 0.7, the feature that has the smallest correlation with the target variable is removed. features that have been selected for flight delay prediction a description is provided for each of the selected features. The features Airport, Airline, Season, Time of day, Day of week, Day of month, Day of year, Airport latitude and longitude, Distance, Month, Year and Scheduled flflights 2h and day are obtained or calculated from the flight schedule dataset. The feature Seats is derived from the aircraft type assigned to perform a flight. The features Temperature, Dewpoint, Visibility, Pressure, and Wind speed are

obtained from the weather dataset

Prediction Features:-

Departure delay Airport a , Airline a , Season a , Time of day b , Day of week b , Day of month b , Day of year b , Airport latitude c , Airport longitude c , Day of month c , Seats c , Year c , Scheduled flights 2 h c , Scheduled flights day c , Dewpoint c , Visibility c , Pressure c , Wind speed c .Arrival delay ,Airport a , Airline a , Aircraft type a , Season a , Time of day b , Day of week b , Day of month b , Month b, Airport longitude c , Day of month c , Distance c , Seats c , Year c , Scheduled flights 2h c , Scheduled flights day c , Temperature c , Visibility c , Pressure c , Wind speed c. This feature is target encoded; b This feature is trigonometrically encoded; c This feature is numerically encoded.

Feature Description:-

Airport - the airport of destination (departures) or origin (arrivals)
Airline - the airline operating the flight
Aircraft - type the aircraft type used for the flight
Season - the flight season (summer or winter schedule)
Time of day - scheduled time of day of the flight
Day of week - scheduled day of the week of the flight
Day of month - scheduled day of the month of the flight
Day of year - scheduled day of the year of the flight
Month - scheduled month number of the flight
Airport - latitude and longitude the latitude and longitude of the destination/origin airport
Distance - the distance between the origin and destination
Seats - the seat capacity of the used aircraft
Year - the year in which the flight was operated
Temperature - the air temperature at the destination/origin airport
Dewpoint - the dewpoint temperature at the destination/origin airport
Visibility - the prevailing visibility at the destination/origin airport
Pressure - pressure altimeter at the destination/origin airport
Wind speed - wind speed at the destination/origin airport
Scheduled flights day - the number of flights scheduled to depart/arrive during the day of flight
Scheduled flights 2h - the number of flights scheduled to depart/arrive during the period between one hour before and one hour after the scheduled time of the flight.

The features are either categorical, time-related, or numerical. The categorical features are target encoded based on a binary delay threshold of 15 min. The encoded value of the sample feature is the delay rate of the category to which the sample belongs. For example: if 8 out of 20 samples flying on Tuesdays are more than 15 min delayed, all Tuesday flights are encoded with value 0.4 for the feature Day of the week. The time features are encoded using trigonometric functions to preserve the periodicity. Two features (sine and cosine) are extracted from every time feature. For example, the features Month sine and cosine are calculated using sin(2πm12 and (cos 2πm 12) for a given month m.

The remaining features are numerically encoded, i.e., the encoded value is the same as the original feature value. Note that the time features are both trigonometrically and numerically encoded. For example, the data field Day of the week yields the features Day of the week sine, Day of the week cosine, and Day of the week. The encoding method of every selected feature is denoted . After encoding, all feature values are scaled to the interval [0, 1] to eliminate undesired feature domination in neural network classifiers most features are selected for at least one of the departure/arrival pair, and that the trigonometrically encoded time features are selected more oftenthan the non-encoded time features.

Machine-Learning Algorithms to Estimate the Probability Distribution of Flight Delays:-

Two algorithms are proposed to estimate the distribution of flflight delays: Mixture Density Networks (MDN) and Random Forests regression (RFR). These algorithms belong to different classes of machine learning algorithms, neural networks, and decision trees, respectively.
Mixture Density Networks (MDNs).
A Mixture Density Network is a combination of a neural network and Gaussian mixture model. Given feature values xi of flflight i, an MDN outputs the parameters for each Gaussian in the mixture: the weight α, the mean μ, and the standard deviation σ. With these parameters, the probability density function p(yi|xi) of the target variable yi, the flight delay, is determined. In general, the MDN is particularly suitable to estimate multimodal probability distributions. It is therefore able to predict a distribution with peaks at, for example, two separate likely delay values.

The flight delay probability distribution is constructed as the weighted sum of Gaussian distributions as follows:

p(yi|xi) = m∑j=1 αj(xi)φj(yi|xi), (1) φj(yi|xi) = 1 q 2πσj(xi)2 exp −(yi − μj(i))2σj(x)2 ! (2)

where p(yi|xi) is the probability distribution of delay value yi given feature values xi from flight sample i, while αj(xi), μj(xi) and σj(xi) are the weight, mean, and standard deviation of the jth Gaussian component, 1 components considered for the mixture. ≤ j ≤ m with m the total number of Gaussian while the parameters αj, μj , and σj are the output of the MDN. Thus, there are 3m outputs of the MDN. The weights use a softmax activation function, and the standard deviations use an exponential activation function, while the means are unrestricted.
The neural network is trained using backpropagation, i.e, the network parameters, the weights and biases of each node are updated using an error function E, which is the negative logarithm of the likelihood that the model derived from the output of the current network gives rise to the training data . This likelihood is the product of the likelihood of every data point, given the

current network parameters.

$$E = Nf \sum i=1- \ln m\sum =1\ \alpha j(xi)\varphi j(yi|xi)! = Nf\ i=1 - \ln p(yi|xi), (3)$$

where Nf is the total number of samples in the training set. For every data point fed to the neural network, the derivatives of the error with respect to all network parameters are used to update the weights and biases of the network. Following training, the MDN is applied to a test set and multimodal probability distributions for the delay of each flflight in the test set are estimated. The MDN method illustrated. Schematic representation of a Mixture Density Network: parameters for a multimodalGaussian distribution are obtained using a Neural Network. Random Forests Regression and Kernel Density Estimation Random Forests regression (RFR) is a class of decision tree-based machine learning algorithms .

The regular RFR algorithm is an ensemble method that combines the results of a number of decision trees. When building each tree, a random subset of the feature values of each training data point is used to make branches. The algorithm outputs a point estimate for the target variable (flflight delay) of every test sample by averaging the output values of all considered decision trees. However, for our analysis, we are interested in estimating the probability distribution for the delay of the given flflight, rather than a point estimate. In order to obtain the flight delay distribution of a flflight in the test phase, the output values of the decision trees are not averaged, but collected, and a kernel density estimation (KDE) is performed . A KDE results in a normalized probability density function.

Two settings of the KDE are the kernel type and the bandwidth. In our analysis, a bandwidth of 1.5 is used to render the estimated distribution smooth. Gaussian kernels have been selected for their generality. Random Forests regression is a well -established technique that has been applied in many research areas. However, there are very few examples of studies utilizing the algorithm to obtain probability distributions. För steretal use quantile values, obtained from Quantile Random Forests, to construct a right-continuous cumulative distribution function of aircraft's time-to-flfly from the turn onto the fifinal approach course to the runway threshold. Schlosseretal and Rahmanetal use Random Forests algorithms to obtain probability distributions for precipitation forecasts and drug sensitivity, respectively.

Both studies make use of feature probability distributions estimated via maximum likelihood to make splitting decisions when constructing the decision trees. In contrast,
in this study, the feature values and splitting decisions are kept deterministic throughout the Random Forests algorithm. In this way, the probability density function is estimated from deterministic feature values without the need for stochastic variables. Furthermore, the working of the original Random Forests regression algorithm need not be changed.

Hyperparameter Tuning:-

The hyperparameters of the MDN and the RFR prediction algorithms have been optimized using a grid search. The hyperparameters leading to the lowest mean CRPS scores have been selected shows the selected hyperparameters and their search range. For MDN, a network with three hidden layers of 50 nodes is selected. The output layer of the network consists of 24 nodes, with which an 8-modal Gaussian distribution function is constructed. For RFR, 200 decision trees with a maximum depth of 10 layers are constructed. For every branch split, three out of four features are considered of at least seven training samples.

MIXTURE DENSITY NETWORK :-

Number of modes m = 8 = [3, 5, 8, 10, 15]
Number of hidden layers = 3 = [1, 2, 3]
Number of nodes per hidden layer = 50 = [25, 50, 75, 100]
Number of epochs = 1000 = [500, 750, 1000, 1250, 1500]

RANDOM FOREST REGRESSION:-

Number of estimators = 200 = [100, 150, 200, 300]
Split criterion = Mean-squared error = [MSE, MAE]
Maximum tree depth = 20 = [4, 6, 8, 10, 12, 15, 20, 30]
Minimum samples per leaf node = 7 = [0, 3, 5, 7, 9]
Fraction of features considered for split = 0.75 = [0.25, 0.50, 0.75, 1.00]
KDE Bandwidth h = 1.5 = [0.5, 1, 1.5, 2]

4.2 Non-Functional requirements :

This section presents multiple different avionics NFRs, related to dependability, performance, development, and operation.

Security

Security is a NFR that gains importance in integrated avionics as described by Jacob[21], Johnson [22] and Royalty [33] for the commercial airplanes sector. For example,recent approaches of connecting aircraft management networks, passenger entertainment, and avionic system in combination with the deployment of well-known COTS technology such as Ethernet and variants of internet protocols have led to increased security considerations as indicated by a recent FAA inquiry for Boeing's 787 [54].

Maintenance

Traditionally commercial aircraft maintenance has followed very cyclic and scheduled maintenance approaches with multiple levels of maintenance actions depending on service. This is largely driven by the requirement for operational schedules of aircraft and crews as well as passenger satisfaction demanding high on-time departure and arrival. Clear quantitative metrics like schedule interruption indexes are designed to and tracked for an aircraft and its subsystems like avionics**.**

Safety, Availability, and Integrity

Dependability as NFR – especially related to safety – can be divided in two major classes. One class addresses the requirements of operation in the system environmentin case of component failures. Requirements of this class often directly manifests in the architecture of avionics. An example of this class would be the success of a mission despite faulty components. The second class concerns the development process of the system itself and its implication of performance in the environment. Such NFRs may or may not show up in the avionics architecture.

Temporal Performance Aspects

IMA uses powerful processors to optimize size, weight and power. Yet, increasing use of pipeline approaches and caches for performance reasons make worst-case execution time analyses harder. Dual-processor approaches with shared resources (like caches and memory busses) may exacerbate this even further.

Testing and Diagnosis

In addition to built-in test and latent error scrubbing tests described above, integration and flight test and potentially diagnosis can require special architecture and NFR. Especially in IMAs, flight test requires observation of a significant amount of data flowing over the network. In addition, end systems send data on the network especially for testing purposes. All test data may need to be selectively evaluated. Given the amount of data, this requires special network bandwidth considerations and other performance and scheduling considerations.

Obsolescence

COTS electronics refresh cycles get shorter and shorter resulting in potential earlier redesign of avionics platforms. As this is often not financially viable, obsolescence aspects become an important non-functional design aspect of avionics.

Schedulability

Since the IMA approach allows multiple applications of different criticality levels to share common computing resources, it is important to keep individual applications away from potential interference.

The main way for protecting integrated applications

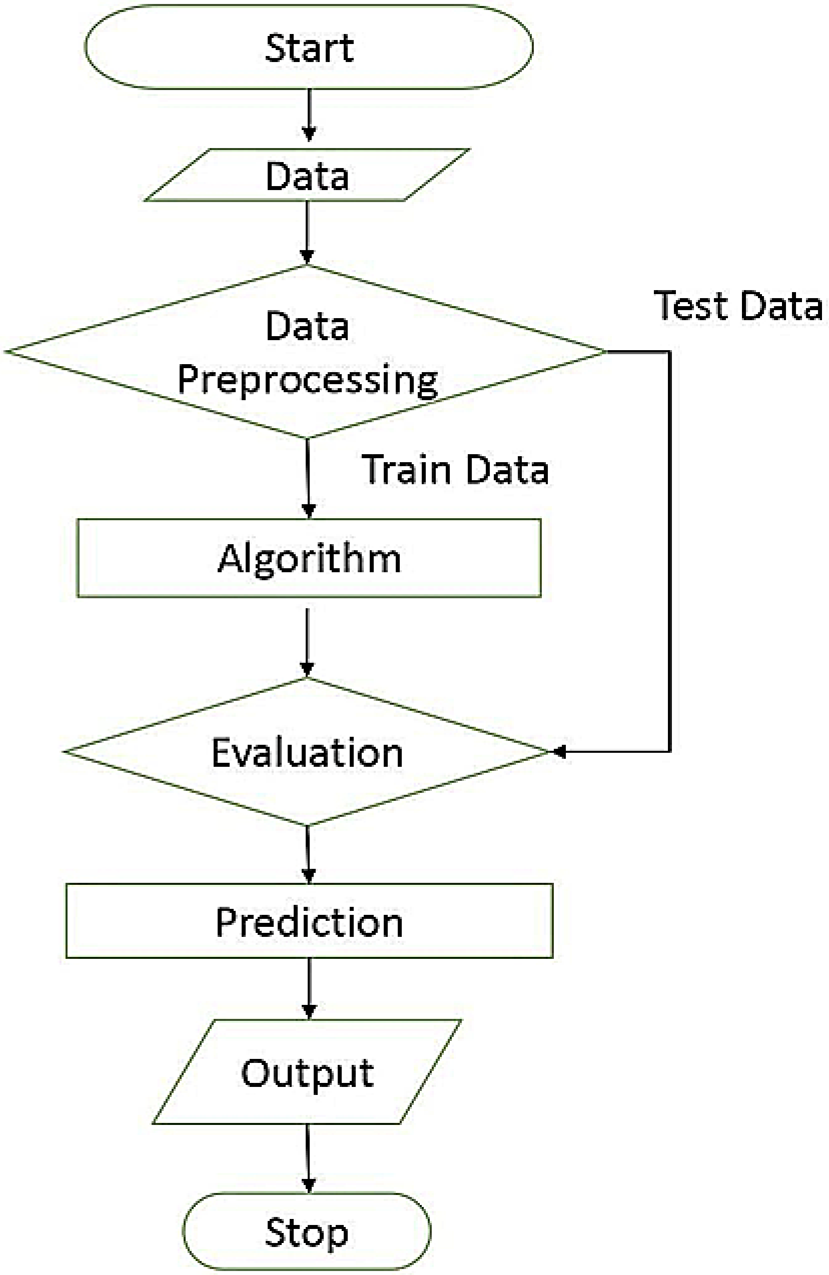
and system resources is via temporal and spatial partitioning. Spatial partitioning guarantees that an application has exclusive control over its own data and state information. With spatial partitioning, an application can be protected from any erroneous behaviors of other applications while sharing same physical resources.

Temporal partitioning guarantees that an application or communication server has temporal exclusive access to its pre-allocated resources. With guaranteed pre-scheduled temporal partitioning, an application can meet their timing requirements.

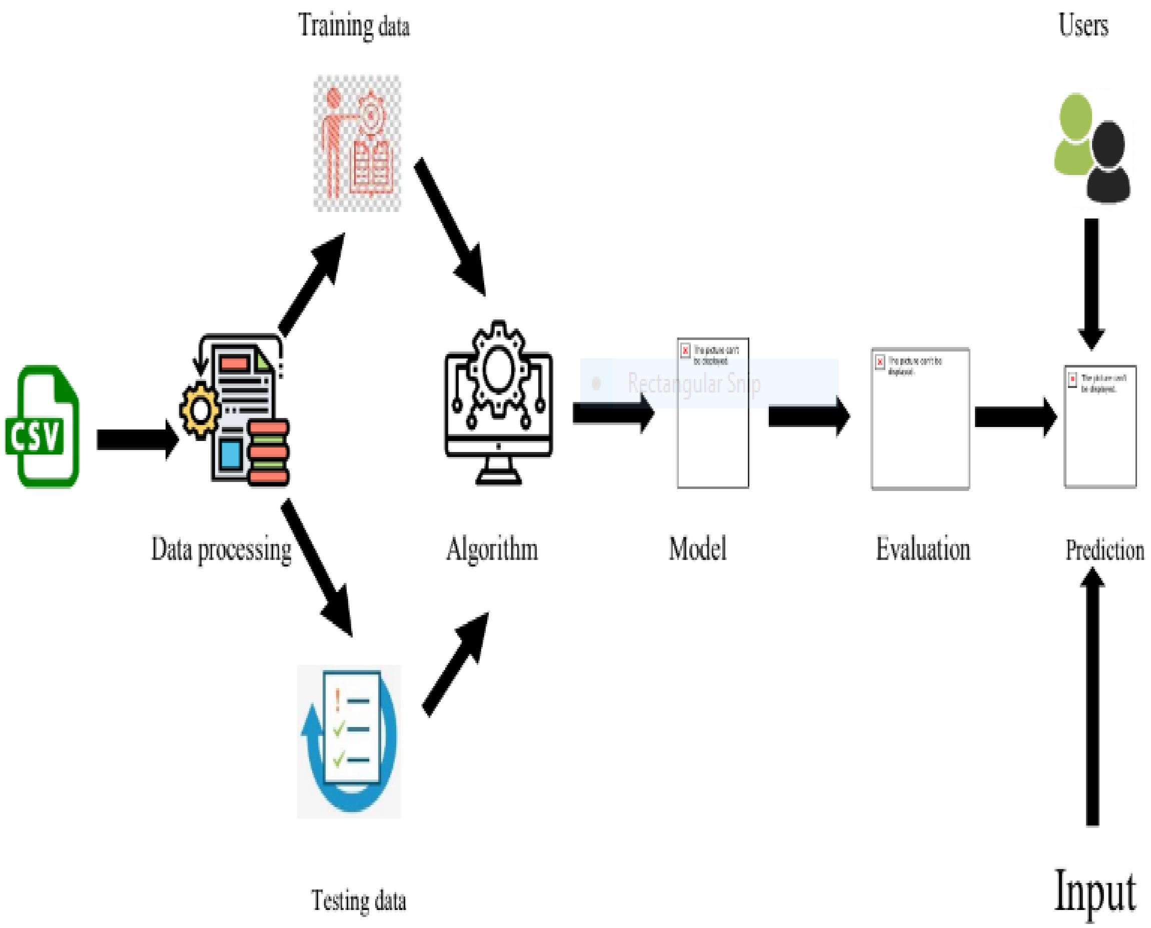
CHAPTER-5

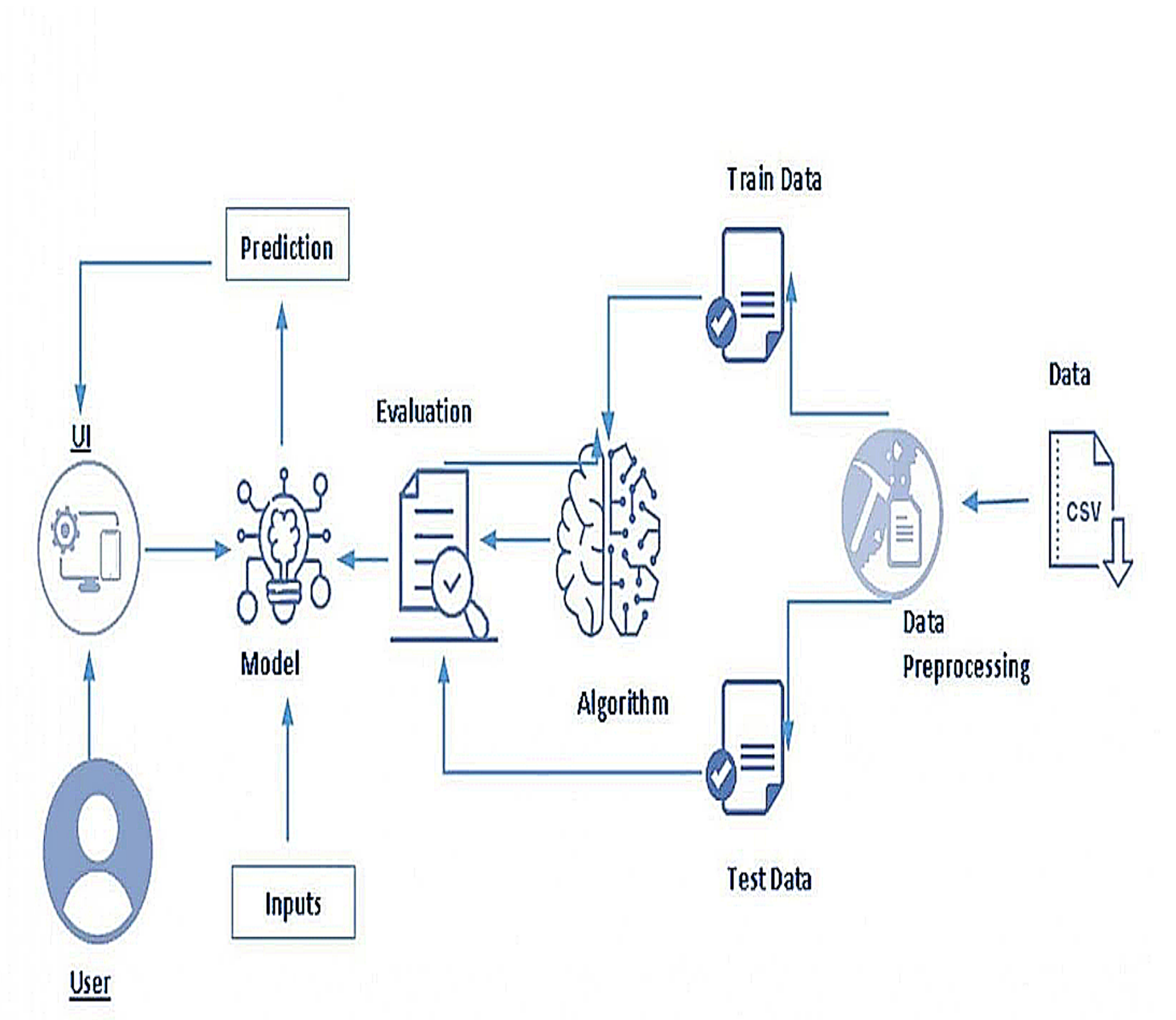
PROJECT DESIGN

5.1 Data Flow Diagrams :



5.2 Solution & Technical Architecture :





5.3 User Stories:

User Type	Functional Requirement	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-2
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-3
		USN-4	As a user, I can register for the application through Gmail		medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	User input	USN-1	As a user I can input the particular URL in the required field and waiting for validation	I can go access the website without any problem	High	Sprint-1
Customer Care Executive	Feature extraction	USN-1	After I compare in case if none found on comparison then we can extract feature using heuristic and visual similarity approach	As a User I can have comparison between websites for security	High	Sprint-1
Administrator	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithms such as Logistic Regression , KNN	In this I can have correct prediction on the particular algorithms	High	Sprint
	Classifier	USN-2	Here I will send all the model output to classifier in order to produce final result	I this I will find the correct classifier for producing the result	medium	Sprint-2

CHAPTER-6
Project Planning & Scheduling

6.1 Sprint Planning & Estimation :

Sprint	Functional Requirement (Epic)	Functional Requirement (Epic)	User Story / Task	StoryPoints	Priority	Team Members
Sprint-1	Data Collection and Preprocessing	USN-1	As a user, I am unable to engage with anything.	2	high	Muruganandhi K Divya P Pavithra R Vishwa Abirami P
Sprint-1	Build HTML Pages	USN-2	As a user, I can view the web pages to enter flight details	1	medium	Muruganandhi K Divya P Pavithra R Vishwa Abirami P
Sprint-2	Build Python Pages	USN-3	As a user, I am unable to engage with anything.	2	high	Muruganandhi K Divya P Pavithra R Vishwa Abirami P
Sprint-2	Execute And Test Your Model	USN-4	As a user, I can predict flight delays using the best created ML models.	2	high	Muruganandhi K Divya P Pavithra R Vishwa Abirami P
Sprint-3	Train The ML Mode	USN-6	As a user, I can predict flight delays using the best created ML models.	2	high	Muruganandhi K Divya P Pavithra R Vishwa Abirami P

Sprint-3	Integrate Flask with Model	USN-5	As a user, I can predict flight delays using the user interface.	2	high	Murugananthi K Divya P Pavithra R Vishwa Abirami P
Sprint-3	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	Murugananthi K Divya P Pavithra R Vishwa Abirami P

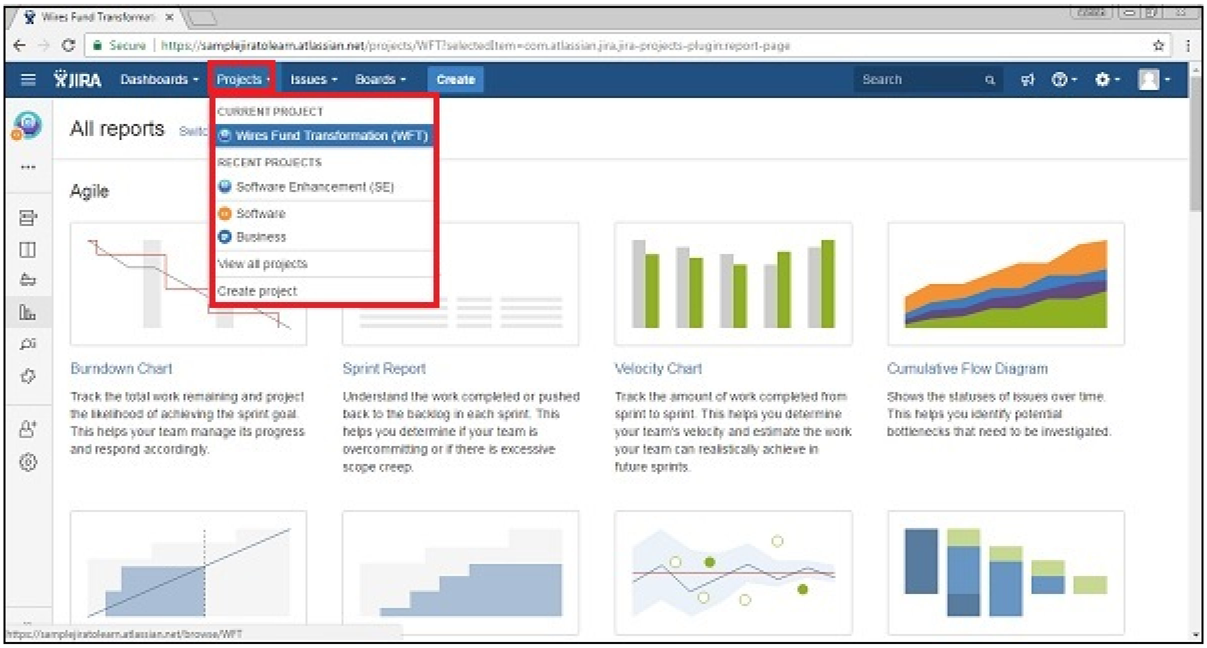
6.2 Sprint Delivery Schedule :

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End D**	**Sprint Release Date (Actual)**
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

6.3 Reports from JIRA :

Work management made easier with Jira reports Identify trends and work smarter, with out-of-the-issue analysis and forecasting in Jira Software.

To access reports in JIRA, the user should go to Project → choose Specific project. The following screenshot shows how to navigate to a specific project.


Wires Fund Transformati
Secure
https://samplejiratolearn.atlassian.net/projects/WFT?selectedItem=com.atlassian.jira.jira-projects-plugin:report-page
JIRA
Dashboards
Projects
Issues
Boards
Create
Search
CURRENT PROJECT
Wires Fund Transformation (WFT)
RECENT PROJECTS
Software Enhancement (SE)
Software
Business
View all projects
Create project
All reports
Agile
https://samplejiratolearn.atlassian.net/browse/WFT

Burndown Chart

Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

Sprint Report

Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.

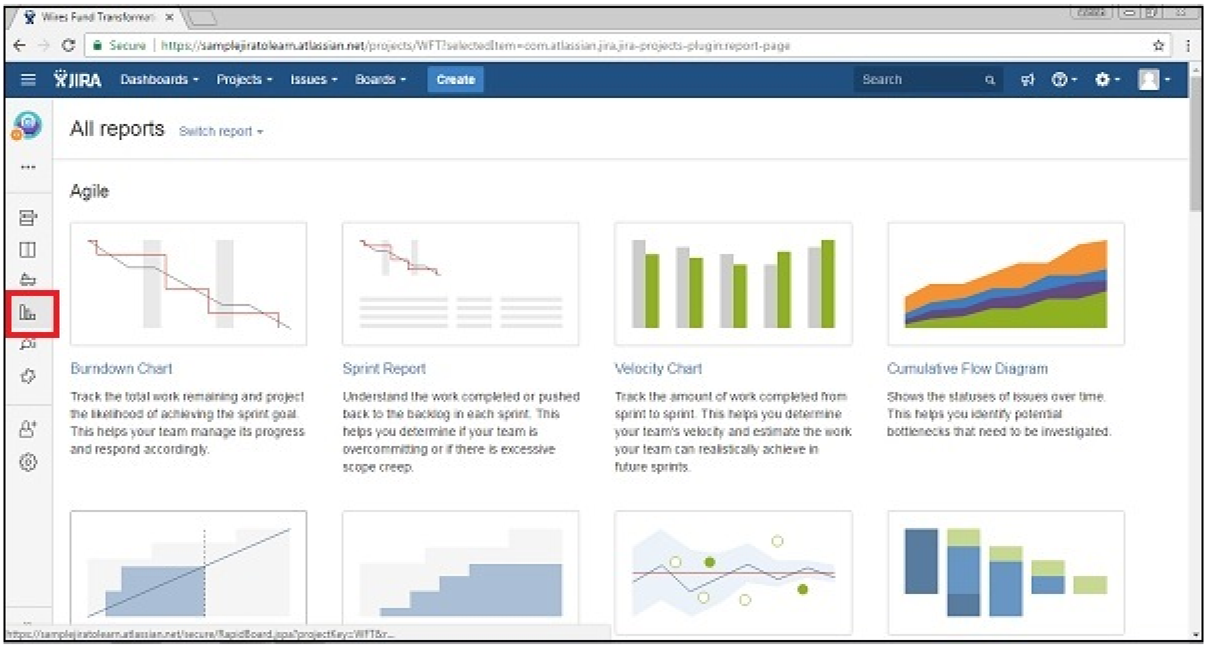
Velocity Chart

Track the amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

Cumulative Flow Diagram

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

Click on the Reports icon on the left side of the page. It will display all the reports supported by JIRA. The following screenshot shows how to access the Report section.


Wires Fund Transformat
Secure | https://samplejiratolearn.atlassian.net/projects/WFT?selectedItem=com.atlassian.jira.jira-projects-plugin:report-page
JIRA
Dashboards
Projects
Issues
Boards
Create
Search
All reports
Switch report
Agile
https://samplejiratolearn.atlassian.net/secure/RapidBoard.jspa?projectKey=WFT&r...

Burndown Chart

Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

Sprint Report

Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.

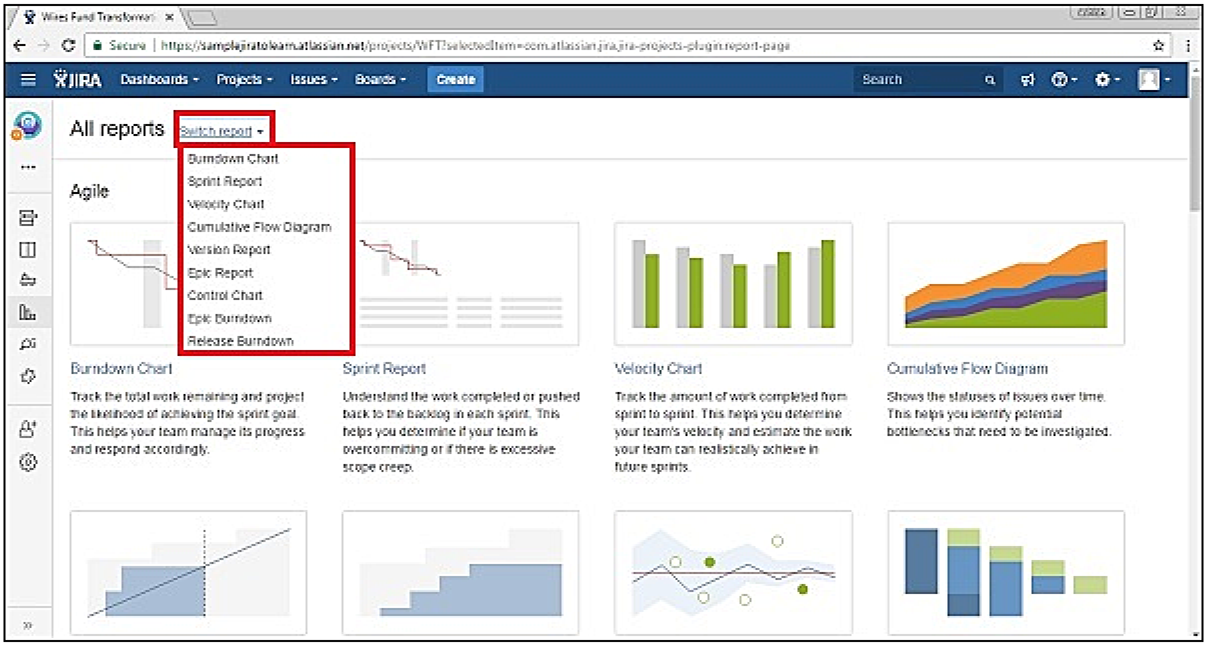
Velocity Chart

Track the amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

Cumulative Flow Diagram

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

When the user clicks on Switch report, it will display the list of reports. The following screenshot shows list of reports available for quick switch.



Burndown Chart

Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

Sprint Report

Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.

Velocity Chart

Track the amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

Cumulative Flow Diagram

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

CHAPTER-7
CODING & SOLUTIONING

7.1 Feature 1

APPLICATION BUILDING USING HTML:

```
<html>
<head>
<!-- CSS only -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9B
v1WTRi" crossorigin="anonymous">
<link rel="stylesheet"
href="{{url_for('static',filename='css/main.css')}}">
</head>
<body>
<center>
<h1>FLIGHT DELAY PREDICTION</h1>

<div id="form">
<form class="form-group" action="/prediction" method="post" >
<div class="row">
<div class="col-md-12">
<div class="row ">
<p class="fw-bolder col-6">ENTER THE FLIGHT

NUMBER : </p>

<input class="form-control-lg form-control-inline form-
control " type="text" name="name" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">MONTH : </p>

<input class="form-control-lg form-control-inline form-
```

```
control" type="text" name="month" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">DAY OF MONTH : </p>

<input class="form-control-lg form-control-inline form-
control" type="text" name="dayofmonth" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">DAY OF WEEK : </p>

<input class="form-control-lg form-control-inline form-
control" class="form-control-inline form-control" type="text"

name="dayofweek" />
</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">ORIGIN : </p>
<select class="form-select form-select-lg form-control-inline "

name="origin">

<option value="alt">ATL - Hartsfield-Jackson Atlanta

International</option>

<option value="dtw">DTW - Detroit Metropolitan Wayne

County </option>

<option value="sea">SEA - Seattle-Tacoma

International</option>

<option value="msp">MSP - Minneapolis–Saint Paul
```

```
International</option>

<option value="jfk">JFK - John F. Kennedy

International</option>
</select>
</div>
<div class="row input-feilds">
<p class="fw-bolder col-6">DESTINATION : </p>
<select class="form-select form-select-lg form-control-inline"

name="destination">

<option value="alt">ATL - Hartsfield-Jackson Atlanta

International</option>

<option value="dtw">DTW - Detroit Metropolitan Wayne

County </option>

<option value="sea">SEA - Seattle-Tacoma

International</option>

<option value="msp">MSP - Minneapolis−Saint Paul

International</option>

<option value="jfk">JFK - John F. Kennedy

International</option>
</select>
</div>
<h2 class="flight-time">ENTER THE FLIGHT TIMINGS</h2>
<div class="row time input-feilds " >
<p class="fw-bolder sdt">SCHEDULED DEPARTURE TIME :

</p>

<div class="col-sm-6 time2">
```

```
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="depthr" />

</div>
<div class="col-sm-6">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="deptmin" />

</div>
</div>
<div class="row time">

<p class="fw-bolder">ACTUAL DEPARTURE TIME : </p>
<div class="col-sm-6 time2">
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="actdepthr" />

</div>
<div class="col-sm-6 ">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="actdeptmin" />

</div>
</div>

<div class="row time">
<p class="fw-bolder">SCHEDULED ARRIVAL TIME : </p>
<div class="col-sm-6 time2">
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="arrtimehr" />
```

```
</div>
<div class="col-sm-6">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-
inline form-control" type="text" name="arrtimemin" />

</div>
</div>
<button type="submit" class="input-feilds btn btn-outline-light

btn-lg">PREDICT</button>
</div>
</div>
</form>
<p>{{y}}</p>
</div>
</center>

</body>
</html>
```

RESULT PAGE HTML:

```
<html>
<head>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9B
v1WTRi" crossorigin="anonymous">
<link rel="stylesheet"
href="{{url_for('static',filename='css/main.css')}}">
</head>
<body>
<center>
<div id="form" class="result">
<h1 class="result-text">{{y}}</h1>
```

```
</div>

</center>
</body>
</html>
```

STYLE.CSS:

```
body{
background-image:url("cloud.jpg");
background-repeat:no-repeat ;
background-size: cover;
background-position: center;
}

h1{
color:white;
font-size: 70px;
margin-top: 30px;
margin-bottom: 30px;
}

p{
color:white;
font-size: 25px;

}

#form{

border:5px outset white ;

border-bottom: 3px solid white;
width:fit-content;
background-image:url("flights.jpg");
background-repeat:no-repeat ;
background-size: cover;
background-position: center;
margin-bottom: 30px;
padding: 20px;
```

```
}

label{
color:white
}

.time{
margin: 10px;
}

.time2{
margin-bottom: 4px;
}

.form-control-inline{
min-width: 0;
width: auto;
display: inline;
}

.input-feilds{

margin-top: 30px;
padding:0px
}

label{
font-size: 20px;

}

.result{
margin-top: 250px;
}

.result-text{
font-size: 100px;
color:rgb(255, 255, 255)
}

.flight-time{
```

```
color:white;
font-size: 50px;
margin-top: 30px;
margin-bottom: 40px;
}
```

7.2 Feature 2 :

APPLICATION BUILDING USING FLASK:

```
from flask import Flask,render_template,request
import requests
# NOTE: you must manually set API_KEY below using information
retrieved from your IBM Cloud account.

# NOTE: you must manually set API_KEY below using information
retrieved from your IBM Cloud account.

API_KEY = "2SNGxCC84_SnT4w-
CK18BSgHa22dH7hgM673se9fq57B"

token_response =
requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

app=Flask( name )

@app.route('/')
def index():
return render_template('index.html')

@app.route('/prediction',methods=["POST"])
def predict():
if request.method=="POST":
name=request.form["name"]
```

```
month=request.form["month"]
if(int(month)>12):
ans="Please Enter the correct Month"
return render_template("index.html" ,y=ans)

dayofmonth=request.form["dayofmonth"]
if(int(dayofmonth)>31):
ans="Please Enter the correct Day of Month"
return render_template("index.html" ,y=ans)

dayofweek=request.form["dayofweek"]
if(int(dayofweek)>7):
ans="Please Enter the correct Day of Week"
return render_template("index.html" ,y=ans)

origin=request.form["origin"]
destination=request.form['destination']

if(origin==destination):
ans="Origin airport and destination airport can't be same"
return render_template("index.html" ,y=ans)

if(origin=="msp"):
origin1,origin2,origin3,origin4,origin5=0,0,0,1,0

if(origin=="dtw"):
origin1,origin2,origin3,origin4,origin5=0,1,0,0,0
if(origin=="jfk"):
origin1,origin2,origin3,origin4,origin5=0,0,1,0,0
if(origin=="sea"):
origin1,origin2,origin3,origin4,origin5=0,0,0,0,1
if(origin=="alt"):
origin1,origin2,origin3,origin4,origin5=1,0,0,0,0

if(destination=="msp"):
destination1,destination2,destination3,destination4,destination5=0,0,0,1,
0
if(destination=="dtw"):
destination1,destination2,destination3,destination4,destination5=0,1,0,0,
0
if(destination=="jfk"):
```

```
destination1,destination2,destination3,destination4,destination5=0,0,1,0,
0
if(destination=="sea"):
destination1,destination2,destination3,destination4,destination5=0,0,0,0,
1
if(destination=="alt"):

destination1,destination2,destination3,destination4,destination5=1,0,0,0,0


depthr=request.form['depthr']
deptmin=request.form['deptmin']
if(int(depthr)>23 or int(deptmin)>59):
ans="Please enter the correct Departure time"
return render_template("index.html" ,y=ans)
else:
dept=depthr+deptmin

actdepthr=request.form['actdepthr']
actdeptmin=request.form['actdeptmin']
if(int(actdepthr)>23 or int(actdeptmin)>59):
ans="Please enter the correct Actual Departure time"
return render_template("index.html" ,y=ans)
else:
actdept=actdepthr+actdeptmin

arrtimehr=request.form['arrtimehr']
arrtimemin=request.form['arrtimemin']
if(int(arrtimehr)>23 or int(arrtimemin)>59):
ans="Please enter the correct Arrival time"
return render_template("index.html" ,y=ans)
else:
arrtime=arrtimehr+arrtimemin

if((int(actdept)-int(dept))<15):
dept15=0

else:
dept15=1
```

```
print(dept15)
total=[[month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin
5,destination1,destination2,destination3,destination4,destination5,dept,a
ctdept,dept15,arrtime]]

# NOTE: manually define and pass the array(s) of values to be
scored in the next line
payload_scoring = {"input_data": [{"fields":
["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14","f15
","f16"], "values": total}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/74fb0eec-a7f5-4bb6-ab8b-
d423e91a872c/predictions?version=2022-11-16', json=payload_scoring,

headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())

pred = response_scoring.json()
value = pred['predictions'][0]['values'][0][0]

print(value)
if(value==[0.]):

ans="THE FLIGHT WILL BE ON TIME"
else:
ans="THE FLIGHT WILL BE DELAYED"

return render_template("results.html" ,y=ans)

if name ==" main ":
app.run(debug=False)
```

CHAPTER-8
TESTING

8.1 Test Cases

1.USER LOGIN

Incorrect input : Incorrect user name, which is the email-id in the case of the Airline Reservation System.

Pass Criteria : An appropriate message should be generated to indicate that an invalid user name has been typed.

Correct Input : The correct input would be a valid email id of the user and a correct password associated with the email is which is used to login.

Pass criteria : The user should be directed to the webpage that the customer is intended to go to after logged into the system.

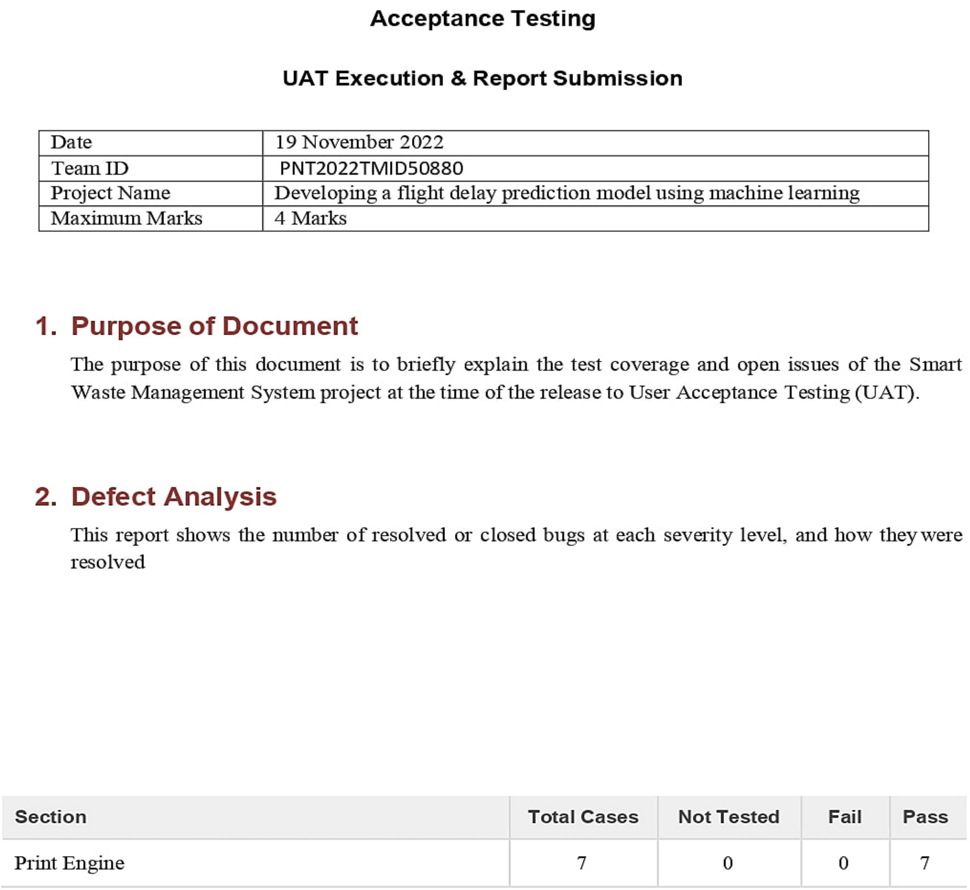
2.USER REGISTRATION

Incorrect Input: Wrong format entered in the input fields for the registration page.

Page Criteria: An appropriate message should be generated to the user saying that he was entered the wrong format in the specific input field.

Correct Input: The correct input would a correct format entered by the customer into the input fields of the registration page.

8.2 User Acceptance Testing



Acceptance Testing

UAT Execution & Report Submission

Date	19 November 2022
Team ID	PNT2022TMID50880
Project Name	Developing a flight delay prediction model using machine learning
Maximum Marks	4 Marks

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Waste Management System project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7

Client Application	51	0	0	51

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	78

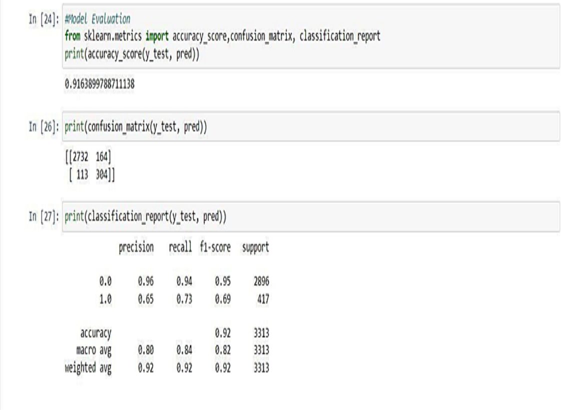
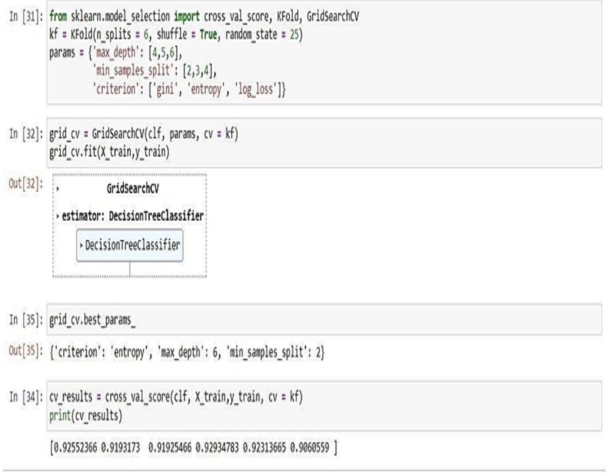
3. Test Case Analysis

This report shows the number of test cases that have passed ,failed ,and untested.

Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER-9
RESULTS

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	Classification Model: Confusion Matrix , Accuracy Score & Classification Report	
2.	Tune the Model	Hyperparameter Tuning , Validation Method	

```
In [24]: #Model Evaluation
         from sklearn.metrics import accuracy_score,confusion_matrix, classification_report
         print(accuracy_score(y_test, pred))

         0.9163899788711138

In [26]: print(confusion_matrix(y_test, pred))

         [[2732  164]
          [ 113  304]]

In [27]: print(classification_report(y_test, pred))

                       precision    recall  f1-score   support

                  0.0       0.96      0.94      0.95      2896
                  1.0       0.65      0.73      0.69       417

             accuracy                           0.92      3313
            macro avg       0.80      0.84      0.82      3313
         weighted avg       0.92      0.92      0.92      3313
```

```
In [31]: from sklearn.model_selection import cross_val_score, KFold, GridSearchCV
         kf = KFold(n_splits = 6, shuffle = True, random_state = 25)
         params = {'max_depth': [4,5,6],
                   'min_samples_split': [2,3,4],
                   'criterion': ['gini', 'entropy', 'log_loss']}

In [32]: grid_cv = GridSearchCV(clf, params, cv = kf)
         grid_cv.fit(X_train,y_train)

Out[32]:          GridSearchCV
          estimator: DecisionTreeClassifier
             DecisionTreeClassifier

In [35]: grid_cv.best_params_

Out[35]: {'criterion': 'entropy', 'max_depth': 6, 'min_samples_split': 2}

In [34]: cv_results = cross_val_score(clf, X_train,y_train, cv = kf)
         print(cv_results)

         [0.92552366 0.9193173  0.91925466 0.92934783 0.92313665 0.9060559 ]
```

CHAPTER-10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

-

- Predicting flight delays can **improve airline operations and passenger satisfaction**, which will result in a positive impact on the economy.
- In this study, the main goal is to compare the performance of machine learning classification algorithms when predicting flight delays.

DISADVANTAGES:

- Flight delays not only irritate air passengers and disrupt their schedules.
- It cause a decrease in efficiency
- Increase in capital costs
- Reallocation of flight crews and aircraft
- Additional crew expenses

CHAPTER-11
CONCLUSION

- In this project, we use flight data, weather, and demand data to predict flight departure delay.
- Our result shows that the Random Forest method yields the best performance compared to the SVM model.
- Somehow the SVM model is very time consuming and does not necessarily produce

better results. In the end, our model correctly predicts 91% of the non-delayed flights.

- However, the delayed flights are only correctly predicted 41% of time. As a result, there can be additional features related to the causes of flight delay that are not yet discovered using our existing data sources.

CHAPTER-12
FUTURE SCOPE

- The scope of this project is very much confined to flight and weather data of United States, but we can include more countries like China, India, and Russia.
- Expanding the scope of this project, we can also add the flight data from international flights and not just restrict our self to the domestic flights.

CHAPTER-13
SOURCE CODE

APPLICATION BUILDING USING HTML:

```
<html>
<head>
<!– CSS only -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9B
v1WTRi" crossorigin="anonymous">
<link rel="stylesheet"
href="{{url_for('static',filename='css/main.css')}}">
</head>
<body>
<center>
<h1>FLIGHT DELAY PREDICTION</h1>
```

```
<div id="form">
<form class="form-group" action="/prediction" method="post" >
<div class="row">
<div class="col-md-12">
<div class="row ">
<p class="fw-bolder col-6">ENTER THE FLIGHT

NUMBER : </p>

<input class="form-control-lg form-control-inline form-
control " type="text" name="name" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">MONTH : </p>

<input class="form-control-lg form-control-inline form-
control" type="text" name="month" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">DAY OF MONTH : </p>

<input class="form-control-lg form-control-inline form-
control" type="text" name="dayofmonth" />

</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">DAY OF WEEK : </p>

<input class="form-control-lg form-control-inline form-
control" class="form-control-inline form-control" type="text"

name="dayofweek" />
```

```
</div>

<div class="row input-feilds">
<p class="fw-bolder col-6">ORIGIN : </p>
<select class="form-select form-select-lg form-control-inline "

name="origin">

<option value="alt">ATL - Hartsfield-Jackson Atlanta

International</option>

<option value="dtw">DTW - Detroit Metropolitan Wayne

County </option>

<option value="sea">SEA - Seattle-Tacoma

International</option>

<option value="msp">MSP - Minneapolis–Saint Paul

International</option>

<option value="jfk">JFK - John F. Kennedy

International</option>
</select>
</div>
<div class="row input-feilds">
<p class="fw-bolder col-6">DESTINATION : </p>
<select class="form-select form-select-lg form-control-inline"

name="destination">

<option value="alt">ATL - Hartsfield-Jackson Atlanta
```

```
International</option>

<option value="dtw">DTW - Detroit Metropolitan Wayne

County </option>

<option value="sea">SEA - Seattle-Tacoma

International</option>

<option value="msp">MSP - Minneapolis–Saint Paul

International</option>

<option value="jfk">JFK - John F. Kennedy

International</option>
</select>
</div>
<h2 class="flight-time">ENTER THE FLIGHT TIMINGS</h2>
<div class="row time input-feilds " >
<p class="fw-bolder sdt">SCHEDULED DEPARTURE TIME :

</p>

<div class="col-sm-6 time2">
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-
inline form-control" type="text" name="depthr" />

</div>
<div class="col-sm-6">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-
inline form-control" type="text" name="deptmin" />
```

```html
</div>
</div>
<div class="row time">

<p class="fw-bolder">ACTUAL DEPARTURE TIME : </p>
<div class="col-sm-6 time2">
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="actdepthr" />

</div>
<div class="col-sm-6 ">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="actdeptmin" />

</div>
</div>

<div class="row time">
<p class="fw-bolder">SCHEDULED ARRIVAL TIME : </p>
<div class="col-sm-6 time2">
<label class="fw-bolder">Hour : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="arrtimehr" />

</div>
<div class="col-sm-6">
<label class="fw-bolder">Minutes : </label>

<input placeholder="00" class="form-control-lg form-control-inline form-control" type="text" name="arrtimemin" />
```

```
</div>
</div>
<button type="submit" class="input-feilds btn btn-outline-light

btn-lg">PREDICT</button>
</div>
</div>
</form>
<p>{{y}}</p>
</div>
</center>

</body>
</html>
```

RESULT PAGE HTML :

```
<html>
<head>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9B
v1WTRi" crossorigin="anonymous">
<link rel="stylesheet"
href="{{url_for('static',filename='css/main.css')}}">
</head>
<body>
<center>
<div id="form" class="result">
<h1 class="result-text">{{y}}</h1>
</div>

</center>
</body>
</html>
```

STYLE.CSS:

```
body{
background-image:url("cloud.jpg");
background-repeat:no-repeat ;
background-size: cover;
background-position: center;
}

h1{
color:white;
font-size: 70px;
margin-top: 30px;
margin-bottom: 30px;
}

p{
color:white;
font-size: 25px;

}

#form{

border:5px outset white ;

border-bottom: 3px solid white;
width:fit-content;
background-image:url("flights.jpg");
background-repeat:no-repeat ;
background-size: cover;
background-position: center;
margin-bottom: 30px;
padding: 20px;
}

label{
```

```
color:white
}

.time{
margin: 10px;
}

.time2{
margin-bottom: 4px;
}

.form-control-inline{
min-width: 0;
width: auto;
display: inline;
}

.input-feilds{

margin-top: 30px;
padding:0px
}

label{
font-size: 20px;

}

.result{
margin-top: 250px;
}

.result-text{
font-size: 100px;
color:rgb(255, 255, 255)
}
```

```
.flight-time{

color:white;
font-size: 50px;
margin-top: 30px;
margin-bottom: 40px;
}
```

APPLICATION BUILDING USING FLASK:

```
from flask import Flask,render_template,request
import requests
# NOTE: you must manually set API_KEY below using information
retrieved from your IBM Cloud account.

# NOTE: you must manually set API_KEY below using information
retrieved from your IBM Cloud account.

API_KEY = "2SNGxCC84_SnT4w-
CK18BSgHa22dH7hgM673se9fq57B"

token_response =
requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

app=Flask( name )

@app.route('/')
def index():
return render_template('index.html')

@app.route('/prediction',methods=["POST"])
def predict():
```

```
if request.method=="POST":
name=request.form["name"]
month=request.form["month"]
if(int(month)>12):
ans="Please Enter the correct Month"
return render_template("index.html" ,y=ans)

dayofmonth=request.form["dayofmonth"]
if(int(dayofmonth)>31):
ans="Please Enter the correct Day of Month"
return render_template("index.html" ,y=ans)

dayofweek=request.form["dayofweek"]
if(int(dayofweek)>7):
ans="Please Enter the correct Day of Week"
return render_template("index.html" ,y=ans)

origin=request.form["origin"]
destination=request.form['destination']

if(origin==destination):
ans="Origin airport and destination airport can't be same"
return render_template("index.html" ,y=ans)

if(origin=="msp"):
origin1,origin2,origin3,origin4,origin5=0,0,0,1,0

if(origin=="dtw"):
origin1,origin2,origin3,origin4,origin5=0,1,0,0,0
if(origin=="jfk"):
origin1,origin2,origin3,origin4,origin5=0,0,1,0,0
if(origin=="sea"):
origin1,origin2,origin3,origin4,origin5=0,0,0,0,1
if(origin=="alt"):
origin1,origin2,origin3,origin4,origin5=1,0,0,0,0

if(destination=="msp"):
```

```
destination1,destination2,destination3,destination4,destination5=0,0,0,1,
0
if(destination=="dtw"):
destination1,destination2,destination3,destination4,destination5=0,1,0,0,
0
if(destination=="jfk"):

destination1,destination2,destination3,destination4,destination5=0,0,1,0,
0
if(destination=="sea"):
destination1,destination2,destination3,destination4,destination5=0,0,0,0,
1
if(destination=="alt"):

destination1,destination2,destination3,destination4,destination5=1,0,0,0,
0

depthr=request.form['depthr']
deptmin=request.form['deptmin']
if(int(depthr)>23 or int(deptmin)>59):
ans="Please enter the correct Departure time"
return render_template("index.html" ,y=ans)
else:
dept=depthr+deptmin

actdepthr=request.form['actdepthr']
actdeptmin=request.form['actdeptmin']
if(int(actdepthr)>23 or int(actdeptmin)>59):
ans="Please enter the correct Actual Departure time"
return render_template("index.html" ,y=ans)
else:
actdept=actdepthr+actdeptmin

arrtimehr=request.form['arrtimehr']
arrtimemin=request.form['arrtimemin']
if(int(arrtimehr)>23 or int(arrtimemin)>59):
ans="Please enter the correct Arrival time"
```

```
return render_template("index.html" ,y=ans)
else:
arrtime=arrtimehr+arrtimemin

if((int(actdept)-int(dept))<15):
dept15=0

else:
dept15=1

print(dept15)
total=[[month,dayofmonth,dayofweek,origin1,origin2,origin3,origin4,origin
5,destination1,destination2,destination3,destination4,destination5,dept,a
ctdept,dept15,arrtime]]

# NOTE: manually define and pass the array(s) of values to be
scored in the next line
payload_scoring = {"input_data": [{"fields":
["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14","f15
","f16"], "values": total}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/74fb0eec-a7f5-4bb6-ab8b-
d423e91a872c/predictions?version=2022-11-16', json=payload_scoring,

headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())

pred = response_scoring.json()
value = pred['predictions'][0]['values'][0][0]

print(value)
if(value==[0.]):

ans="THE FLIGHT WILL BE ON TIME"
else:
```

```
ans="THE FLIGHT WILL BE DELAYED"

return render_template("results.html" ,y=ans)

if name ==" main ":
app.run(debug=False)
```

GITHUB & PROJECT DEMO LINK

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-48693-1660811508

PROJECT DEMO LINK :

https://drive.google.com/file/d/11ScgBWDwvwQuilVoY45b56vHpfh6c6ky/view?usp=drivesdk