

PREDICTING FLIGHT DELAYS WITH ERROR CALCULATION USING MACHINE LEARNED CLASSIFIERS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in Information Technology

by

BALAMURUGAN.R (Reg No : 38120019)
BARANIDARAN.GT(Reg No : 38120021)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

MARCH - 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of BALAMURUGAN.R(REG NO: 38120019) BARANIDARAN.GT (REG NO: 38120021) who have done the Project work as a team who carried out the project entitled "**PREDICTING FLIGHT DELAYS WITH ERROR CALCULATION USING MACHINE LEARNED CLASSIFIERS**" under my supervision from November 2021 to April 2022.

Internal Guide
V. MARIA ANU M.E., Ph.D.,

Head of the Department
DR.R.SUBHASHINI M.E., PH.D.,

Submitted for Viva Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, BALAMURUGAN.R (REG NO: 38120019) and BARANIDARAN.GT (REG NO: 38120021) hereby declare that the Project Report entitled “**PREDICTING FLIGHT DELAYS WITH ERROR CALCULATION USING MACHINE LEARNED CLASSIFIERS**” done by us under the guidance of **V.MARIA ANU M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.R.Subhashini M.E., Ph.D. , Head** of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **V.Maria Anu M.E., Ph.D.**, his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Flight delay is a major problem in the aviation sector. During the last two decades, the growth of the aviation sector has caused air traffic congestion, which has caused flight delays. Flight delays result not only in the loss of fortune also negatively impact the environment. Flight delays also cause significant losses for airlines operating commercial flights. Therefore, they do everything possible in the prevention or avoidance of delays and cancellations of flights by taking some measures. In Tree Regression this paper, using machine learning models such as Logistic Regression, Decision Bayesian, Ridge, Random Forest Regression and Gradient Boosting Regression we predict whether the arrival of a particular flight will be delayed or not.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	1
	1.1 Machine Learning	1
	1.2 Logistic Regression Algorithm	3
	1.3 Decision Tree Algorithm	5
	1.4 Random Forest Algorithm	7
	1.5 Literature Review	8
2	PROBLEM STATEMENT	11
	2.1 Existing System	11
	2.1.1 Disadvantages	11
3	DEVELOPMENT PROCESS	12
	3.1 Requirement Analysis	12
	3.2 Resource Requirements	13
	3.3 System Design	14
	3.4 System Architecture	15
	3.5 Module Description	15
4	SYSTEM STUDY	21
	4.1 Feasibility Study	21
	4.1.1 Economic Feasibility	22
	4.1.2 Technology Feasibility	22
	4.1.3 Social Feasibility	22
5	TESTING	22
	5.1 Type OF Tests	23
	5.1.1 Unit Testing	23

	5.1.2 Integration Testing	23
	5.1.3 Function Test	23
	5.1.4 System Test	23
6	CONCLUSION AND FUTURE WORK	28
	6.1 Summary	28
	A. Sample code	28
	B. PublicationReport	36
	IEEE copy rightform	38
	IEEEAcceptanceform	42
	Reference	43

C. Sample code	28
C.Publication Report	36
IEEE copy right form	38
IEEE Acceptance form	42
Reference	43

LIST OF FIGURES		
S.NO	NAME	PAGE NO
1.1	Block Diagram	2
1.2	Logistic Function	4
1.3	Decision Tree Structure	6
3.3	System Architecture	14
3.5	Dataset Collected	16
3.5	Flight Delay Prediction	19
3.5	Flight Delay Error	19
5.1.7	Variable distribution's distance	25
5.1.7	The week's day	26
5.1.7	Graph of accuracy algorithms	27

LIST OF ABBREVIATION

ML	Machine Learning
FAA	Federal Aviation Administration
GUI	Graphical User Interface

CHAPTER 1

INTRODUCTION

Flight delay is studied vigorously in various research in recent years. The growing demand for air travel has led to an increase in flight delays. According to the Federal Aviation Administration (FAA), the aviation industry loses more than \$3 billion in a year due to flight delays and, as per BTS, in 2016 there were 860,646 arrival delays. The reasons for the delay of commercial scheduled flights are air traffic congestion, passengers increasing per year, maintenance and safety problems, adverse weather conditions, the late arrival of plane to be used for next flight. In the United States, the FAA believes that a flight is delayed when the scheduled and actual arrival times differs by more than 15 minutes. Since it becomes a serious problem in the United States, analysis and prediction of flight delays are being studied to reduce large costs.

1.1. MACHINE LEARNING

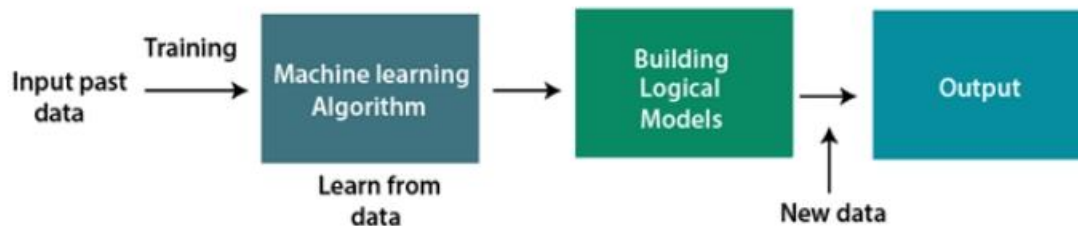
Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as: “Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed”.

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of

these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



1.1.1. Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

1.1.2. Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

It can be further classified into two categories of algorithms:

- Clustering
- Association

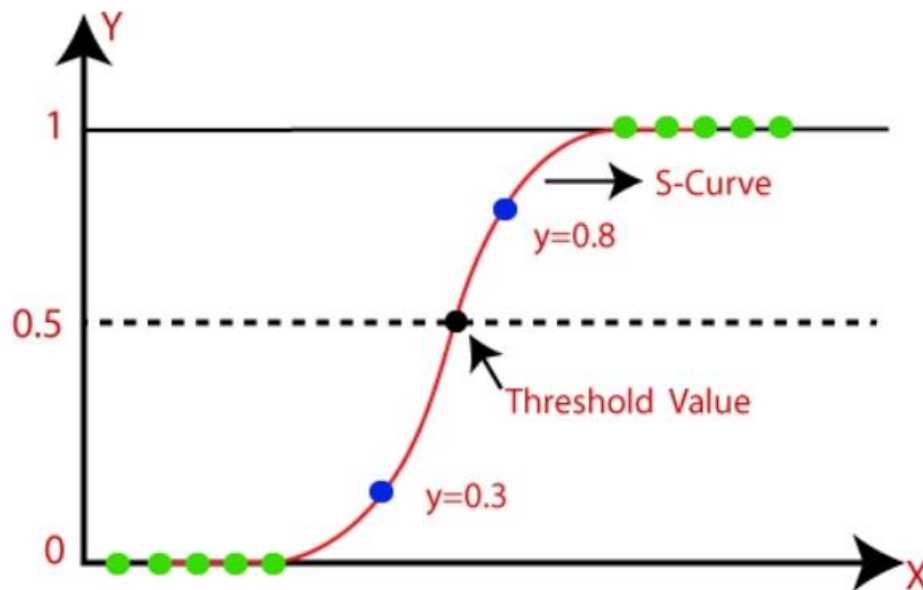
1.2. LOGISTIC REGRESSION ALGORITHM

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line,

we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function



1.2.1. Logistic Function (Sigmoid Function):

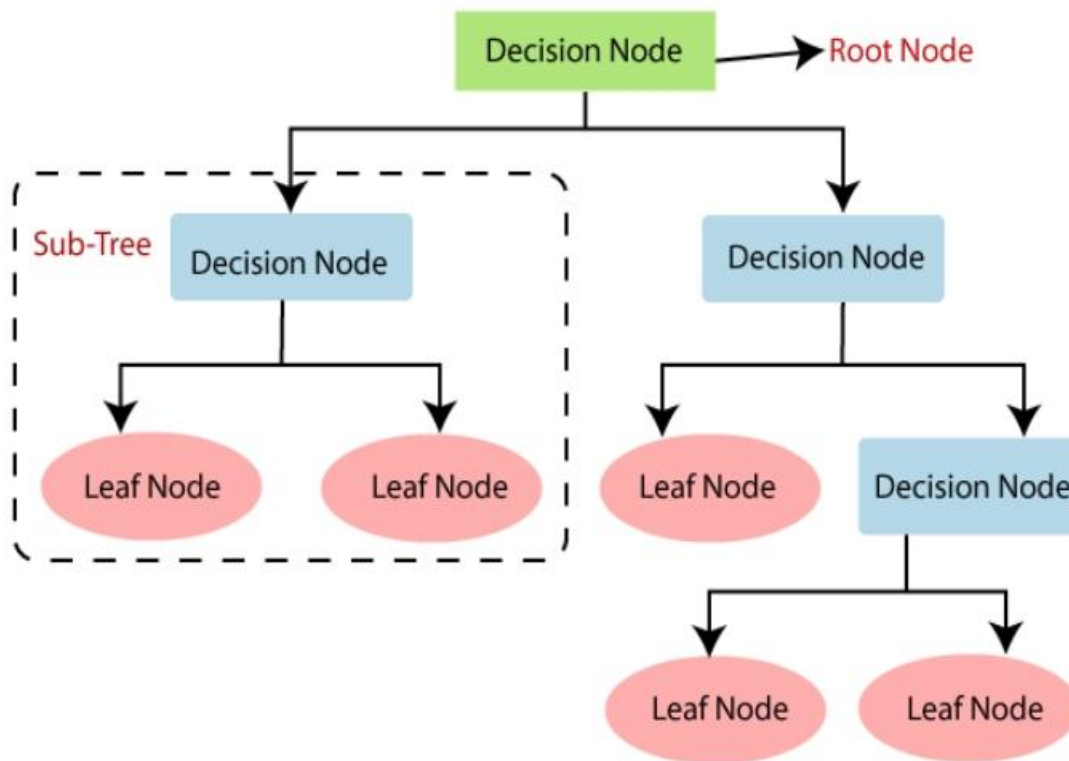
- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

1.3. DECISION TREE ALGORITHM

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:



There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

1.4. RANDOM FOREST ALGORITHM

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.

- It can also maintain accuracy when a large proportion of data is missing.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

1.5. LITERATURE REVIEW

[1] Title: Development of a predictive model for on-time arrival flight of airliner by discovering correlation between flight and weather data

Authors: Noriko Etani - 2019

Description:

An important business of airlines is to get customer satisfaction. Due to bad weather, a mechanical reason, and the late arrival of the aircraft to the point of departure, flights delay and lead to customer dissatisfaction. A predictive model of on-time arrival flight is proposed with using flight data and weather data. The key research in this paper is to discover the correlation between flight data and weather data. The relation between pressure pattern and flight data of Peach Aviation, which is LCC (low-cost carrier) in Japan, are clarified, and it is found that the sea-level pressures of 3 weather observation spots, which are Wakkanai as the most northern spot, Minami-Torishima as the most eastern spot, and Yonagunijima as the most western spot, can classify the pressure patterns. As a result, on-time arrival flight is predicted at 77% of the accuracy with using Random Forest Classifier of machine learning. Furthermore, feasibility of the predictive model is evaluated by developing a tool of on-time arrival flight prediction.

[2] Title: Flight delay prediction for commercial air transport: A deep learning approach
Authors: SobhanAsian - 2019

Description:

This study analyzes high-dimensional data from Beijing International Airport and presents a practical flight delay prediction model. Following a multifactor approach, a novel deep belief network method is employed to mine the inner patterns of flight delays. Support vector regression is embedded in the developed model to perform a supervised fine-tuning within the presented predictive architecture. The proposed method has proven to be highly capable of handling the challenges of large datasets and capturing the key factors influencing delays. This ultimately enables connected airports to collectively alleviate delay propagation within their network through collaborative efforts (e.g., delay prediction synchronization).

[3] TITLE: A Review on Flight Delay Prediction

Authors: Alice sternberg, Jorge Soares, Diego Carvalho, Eduardo Ogasawara – 2017

Description:

Flight delays hurt airlines, airports, and passengers. Their 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. In this context, this paper presents a thorough literature review of approaches used to build flight delay prediction models from the Data Science perspective. We propose a taxonomy and summarize the initiatives used to address the flight delay prediction problem, according to scope, data, and computational methods, giving particular attention to an increased usage of machine learning methods. Besides, we also present a timeline of significant works that depicts relationships between flight delay prediction problems and research trends to address them.

[4] Title: Flight Arrival Delay Prediction And Analysis Using Ensemble Learning
Authors: Xiaotong Dou - 2020

Description:

With the development of the civil aviation transportation industry in recent years, the volume of civil aviation transportation has increased rapidly. Increased carrier costs and reduced airport operating efficiency caused by flight delays have become issues that need to be addressed. How to improve the accuracy of predicting flight arrival delay time is of great significance for

improving airport transportation efficiency, rationally scheduling flights and improving passenger comfort. In this paper, the Cat-boost model is utilized on the U.S Domestic airline on-time performance data from U.S. Transportation Administration, combined with the characteristics of the model to determine the influencing factors, and to predict the arrival delays of flights within the United States. The accuracy;precision and some other criterion of the model are given to evaluate the performance on the data. A better effect is obtained: the accuracy reach 80.44% in this case. Finally, the specific delay time is predicted, we found that the support vector machine has the best prediction result for the flight delay time, the average prediction error is 9.733 min, which has a certain reference value for flight operation and airport scheduling.

[5] Title: A statistical approach to predict flight delay using
gradient boosted decision tree

Authors: Suvojit Manna , Sanket Biswas , Riyanka Kundu , Somnath Rakshit

Description:

Supervised machine learning algorithms have been used extensively in different domains of machine learning like pattern recognition, data mining and machine translation. Similarly, there has been several attempts to apply the various supervised or unsupervised machine learning algorithms to the analysis of air traffic data. However, no attempts have been made to apply Gradient Boosted Decision Tree, one of the famous machine learning tools to analyse those air traffic data. This paper investigates the effectiveness of this successful paradigm in the air traffic delay prediction tasks. By combining this regression model based on the machine learning paradigm, an accurate and sturdy prediction model has been built which enables an elaborated analysis of the patterns in air traffic delays. Gradient Boosted Decision Tree has shown a great accuracy in modeling sequential data. With the help of this model, day-to-day sequences of the departure and arrival flight delays of an individual airport can be predicted efficiently. In this paper, the model has been implemented on the Passenger Flight on-time Performance data taken from U.S. Department of Transportation to predict the arrival and departure delays in flights. It shows better accuracy as compared to other methods.

CHAPTER 2

PROBLEM STATEMENT

2.1 EXISTING SYSTEM

The Existing system proposed that, The expected growth in air travel demand and the positive correlation with the economic factors highlight the significant contribution of the aviation community to the U.S. economy. On-time operations play a key role in airline performance and passenger satisfaction. Thus, an accurate investigation of the variables that cause delays is of major importance. The application of machine learning techniques in data mining has seen explosive growth in recent years and has garnered interest from a broadening variety of research domains including aviation. This study employed a support vector machine (SVM) model to explore the non-linear relationship between flight delay outcomes. These findings provide insight for better understanding of the causes of departure delays and the impacts of various explanatory factors on flight delay patterns.

The primary contribution of Existing study is to investigate the possibility of using SVM models for analysis of the causes of flight delay and investigation of flight delay patterns. The maximum precision achieved was 79.7% with gradient booster as a classifier with a limited data set

2.1.1 DISADVANTAGES

- There is import of training and testing dataset is very small to predict the flight delays.
- Less accuracy prediction when testing the new dataset.
- Taken huge time to predict the Flight error.

CHAPTER 3

DEVELOPMENT PROCESS

3.1. REQUIREMENT ANALYSIS

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system's purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

3.1.1 PYTHON:

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.

For example, `x=10` .Here, x can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear *syntax* and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of *operating systems*, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users.

Features in Python

There are many features in Python, some of which are discussed below

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support

- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted Language

3.2. RESOURCE REQUIREMENTS:

SOFTWARE REQUIREMENTS:

Operating System	Windows 7 or later
Simulation Tool	Anaconda (Jupyter notebook)
Documentation	Ms – Office

HARDWARE REQUIREMENTS:

CPU type	Intel Pentium
Ram size	4GB
Hard disk capacity	80 GB
Keyboard type	Internet keyboard
Monitor type	15 Inch colour monitor
CD -drive type	52xmax

3.3 SYSTEM DESIGN

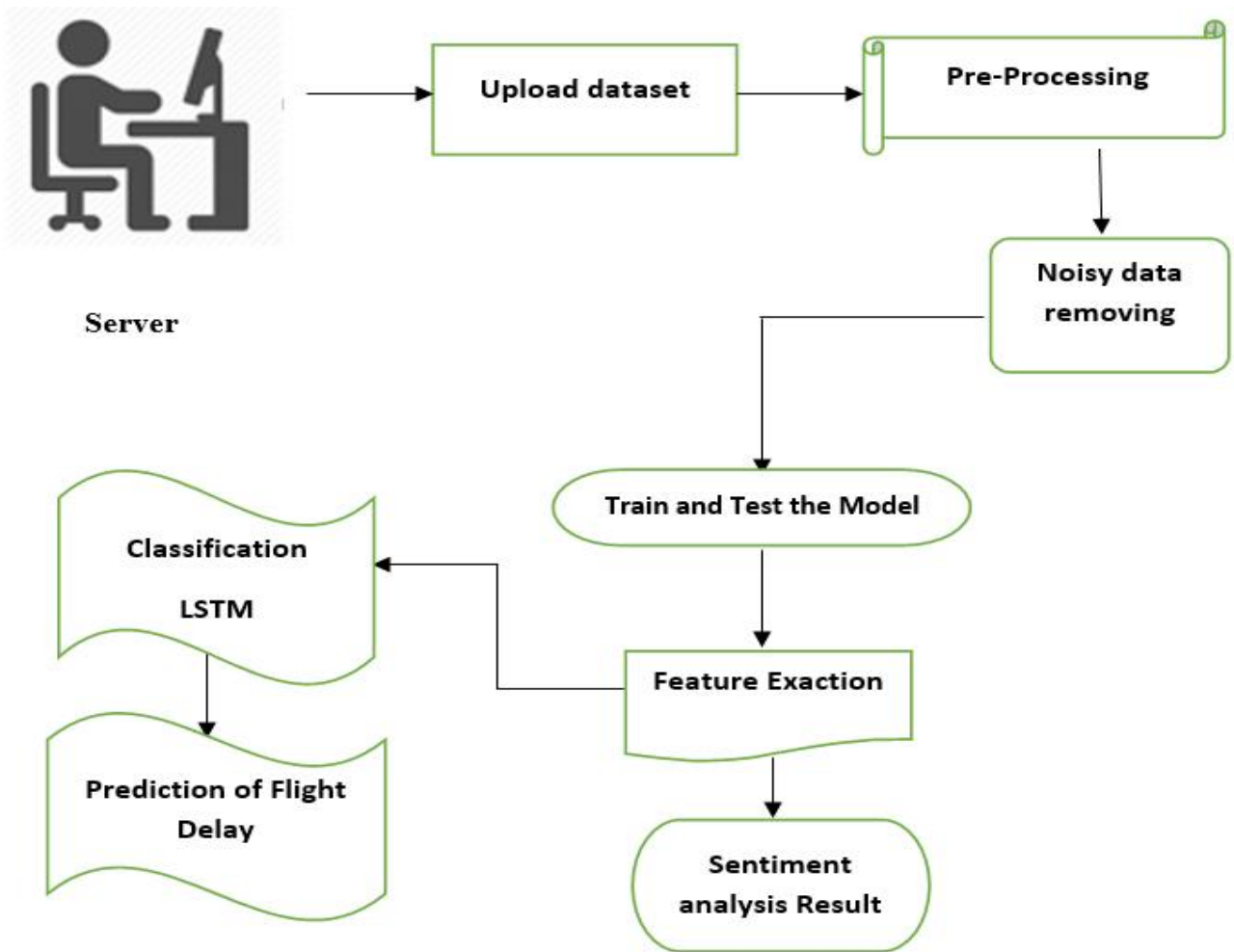


Fig. 1 System Architecture

3.3.1 PROPOSED SYSTEM

- Our proposed model does everything possible in the prevention or avoidance of delays and cancellations of flights by taking some measures.
- In this model, using machine learning models such as Logistic Regression, Decision Tree Regression, Bayesian Ridge, Random Forest Regression and Gradient Boosting Regression
- We predict whether the arrival of a particular flight will be delayed or not.
- We develop a system that predicts for a delay in flight departure based on certain parameters. We train our model for forecasting using various attributes of a particular flight, such as arrival performances, flight summaries, origin/destination, etc.

3.3.2 ADVANTAGES

- The system collects huge number of dataset to train to the model and predict the flight delay error calculation.
- Speed and accuracy score is high.
- Prediction rate is high.

3.4 SYSTEM ARCHITECTURE

3.5 MODULE DESCRIPTION:

1. Module 1: Data collection
2. Module 2: Pre-Processing
3. Module 3: Feature Extraction
4. Module 4: Evaluation

Module 1: Data collection

To predict flight delays to train models, we have collected data accumulated by the Bureau of Transportation; U.S. Statistics of all the domestic flights taken in 2015 was used. The US Bureau of Transport Statistics provides statistics of arrival and departure that includes actual departure time, scheduled departure time, and scheduled elapsed time, wheels-off time, departure delay and taxi-out time per airport. Cancellation and Rerouting by the airport and the airline with

the date and time and flight labelling along with airline airborne time are also provided. The data set consists of 25 columns and 59986 rows. Fig. 1 shows some of the fields of the original dataset. There were many lines with missing and null values. The data must be pre-processed for later use

The methodology here uses the supervised learning technique to gather the advantages of having the schedule and real arrival time. Initially, some specific monitoring algorithms with a light computation cost were considered candidates and therefore the best candidate was perfected for the final model.

We develop a system that predicts for a delay in flight departure based on certain parameters. We train our model for forecasting using various attributes of a particular flight, such as arrival performances, flight summaries, origin/destination, etc

	A	B	C	D	E	F	G	H	I
1	MONTH	DAY_OF_M	DAY_OF_V	OP_UNIQUE	ORIGIN	DEST	DEP_TIME	DEP_DEL1	DISTANCE
2	2	1	6	MQ	CLT	LYH	1430	0	175
3	2	8	6	MQ	CLT	LYH	1442	0	175
4	2	13	4	MQ	DFW	SHV	2247	0	190
5	2	14	5	MQ	DFW	SHV	2230	0	190
6	2	15	6	MQ	DFW	SHV	2246	0	190
7	2	16	7	MQ	DFW	SHV	2230	0	190
8	2	17	1	MQ	DFW	SHV	2237	0	190
9	2	18	2	MQ	DFW	SHV	2230	0	190
10	2	19	3	MQ	DFW	SHV	2240	0	190
11	2	20	4	MQ	DFW	SHV	2226	0	190
12	2	21	5	MQ	DFW	SHV	2230	0	190
13	2	22	6	MQ	DFW	SHV	2231	0	190
14	2	23	7	MQ	DFW	SHV	2231	0	190
15	2	24	1	MQ	DFW	SHV	2231	0	190
16	2	25	2	MQ	DFW	SHV	2232	0	190
17	2	26	3	MQ	DFW	SHV	2228	0	190
18	2	27	4	MQ	DFW	SHV	2225	0	190
19	2	28	5	MQ	DFW	SHV	2223	0	190
20	2	29	6	MQ	DFW	SHV	2232	0	190

1	tweet_id	airline_ser	airline_ser	negative	negative	airline	airline_ser	name	negative	retweet_c	text	tweet_coc	tweet_cre	tweet_loc	user_timezone
2	5.7E+17	neutral	1			Virgin America	cairdin			0	@VirginAmerica Wha	#####			Eastern Time (US & Canada)
3	5.7E+17	positive	0.3486			Virgin America	jnardino			0	@VirginAmerica plus	#####			Pacific Time (US & Canada)
4	5.7E+17	neutral	0.6837			Virgin America	yvonnalynn			0	@VirginAmerica I did	#####	Lets Play		Central Time (US & Canada)
5	5.7E+17	negative	1	Bad Flight	0.7033	Virgin America	jnardino			0	@VirginAmerica it's r	#####			Pacific Time (US & Canada)
6	5.7E+17	negative	1	Can't Tell	1	Virgin America	jnardino			0	@VirginAmerica and i	#####			Pacific Time (US & Canada)
7	5.7E+17	negative	1	Can't Tell	0.6842	Virgin America	jnardino			0	@VirginA	#####			Pacific Time (US & Canada)
8	5.7E+17	positive	0.6745			Virgin America	cjmoginnis			0	@VirginAmerica yes, i	#####	San Franci		Pacific Time (US & Canada)
9	5.7E+17	neutral	0.634			Virgin America	pilot			0	@VirginAmerica Reall	#####	Los Angele		Pacific Time (US & Canada)
10	5.7E+17	positive	0.6559			Virgin America	dhepburn			0	@virginamerica Well,	#####	San Diego		Pacific Time (US & Canada)
11	5.7E+17	positive	1			Virgin America	YupitsTate			0	@VirginAmerica it wa	#####	Los Angele		Eastern Time (US & Canada)
12	5.7E+17	neutral	0.6769			Virgin America	idk_but_youtube			0	@VirginAmerica did y	#####	1/1 loner s		Eastern Time (US & Canada)
13	5.7E+17	positive	1			Virgin America	HyperCamiLax			0	@VirginAmerica I <	#####	NYC	America/New_York	
14	5.7E+17	positive	1			Virgin America	HyperCamiLax			0	@VirginAmerica This	#####	NYC	America/New_York	
15	5.7E+17	positive	0.6451			Virgin America	mollanderson			0	@VirginAmerica @vir	#####			Eastern Time (US & Canada)
16	5.7E+17	positive	1			Virgin America	sjespers			0	@VirginAmerica Than	#####	San Franci		Pacific Time (US & Canada)
17	5.7E+17	negative	0.6842	Late Flight	0.3684	Virgin America	smartwatermelon			0	@VirginAmerica SFO-	#####	palo alto,		Pacific Time (US & Canada)
18	5.7E+17	positive	1			Virgin America	ItzBrianHunty			0	@VirginAmerica So er	#####	west covin		Pacific Time (US & Canada)
19	5.7E+17	negative	1	Bad Flight	1	Virgin America	heatherovieda			0	@VirginAmerica I fle	#####	this place		Eastern Time (US & Canada)
20	5.7E+17	positive	1			Virgin America	thebrandiray			0	I âߝ flying @VirginA	#####	Somewher		Atlantic Time (Canada)

Fig. 2. Snapshot of Dataset

Module 2: Pre-Processing

Once the data is extracted from the twitter source as the datasets, this information has to be passed to the classifier. The classifier cleans the dataset by removing redundant data like stop words, emoticons in order to make sure that non textual content is identified and removed before the analysis.

Text pre-processing is an essential a part of any NLP method and the significance of the NLP pre-processing are

- To minimize indexing (or knowledge) records dimension of the textual content records
 1. Stop words bills 20-30% of total phrase counts in a special textual content record
 2. Stemming may just diminish indexing size as much as forty- 50%

- To make stronger the efficiency and effectiveness of the IR method
 1. Stop words aren't valuable for shopping or textual content mining
 2. Stemming used for matching the similar words in a text record

Tokenization:

Tokenization is the process of breaking a circulate of textual content into phrases, phrases, symbols, or different significant factors called tokens .The aim of the tokenization is the exploration of the phrases in a sentence. The list of tokens turns into input for further processing akin to parsing or textual content mining. Tokenization is valuable both in linguistics (where it's a form of textual content segmentation), and in laptop science, the place it forms a part of lexical analysis. Textual knowledge is simplest a block of characters at the starting.

All strategies in know-how retrieval require the words of the data set. For that reason, the requirement for a parser is a tokenization of records. This might be sound trivial because the text is already saved in computing device-readable codecs. However, some problems are nonetheless left, like the removing of punctuation marks. Different characters like brackets, hyphens, and so on require processing as well.

Stop word Removal:

Stop phrases are very more often than not used fashioned phrases like ‘and’, ‘are’, ‘this’ etc. They don't seem to be useful in classification of records. So they must be removed. However, the development of such stop phrases record is problematic and inconsistent between textual sources. This process also reduces the text knowledge and improves the approach performance. Each textual content report offers with these phrases which are not vital for text mining applications.

Stemming and Lemmatization:

The aim of both stemming as well as lemmatization is to scale down inflectional types & mostly derivationally associated varieties of a phrase to a fashioned base kind.

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of accomplishing this goal accurately more often than not, and quite often involves the removal of derivational affixes.

Lemmatization often refers to doing matters competently with the usage of a vocabulary and morphological analysis of phrases, in most cases aiming to eliminate inflectional endings only and to come back the base or dictionary type of a word, which is often called the lemma.

```
tweet_id          0
airline_sentiment 0
airline_sentiment_confidence 0
negativereason    5462
negativereason_confidence 4118
airline           0
airline_sentiment_gold 14600
name             0
negativereason_gold 14608
retweet_count     0
text             0
tweet_coord      13621
tweet_created     0
tweet_location   4733
user_timezone    4820
dtype: int64
```

Fig 3. Flight Delay Prediction

MONTH	574268
DAY_OF_MONTH	574268
DAY_OF_WEEK	574268
OP_UNIQUE_CARRIER	574268
ORIGIN	574268
DEST	574268
DEP_TIME	569330
DEP_DEL15	569317
DISTANCE	574268
Unnamed: 9	0
dtype: int64	

Fig 4. Flight Delay error

Module 3: Feature Extraction

We have studied from various sources to find out which parameters will be most appropriate to predict the departure and arrival delays. After several searches, we conclude the following parameters:

- Day Departure
- Delay Airline
- Flight Number
- Destination Airport
- Origin Airport

- Day of Week
- Taxi out

Module 4: Evaluation

After pre-processing and feature extraction of our dataset, 60% of the dataset was selected for training and 40% of the dataset was selected for testing. For error calculation, we are using scikit-learn metrices. Results are divided between two sections, Departure Delay(A) and Arrival Delay(B).

A. Departure Delay

our results for departure delay which compares different Machine Learning models, i.e. Logistic Regression, Decision Tree Regressor, Bayesian Ridge, Random Forest Regressor and Gradient Boosting Regressor, based on various evaluation metrics. Further, we compare each model concerning one evaluation metric at a time.

B. Arrival Delay

our results for arrival delay which compares different Machine Learning models, i.e. Logistic Regression, Decision Tree Regressor, Bayesian Ridge, Random Forest Regressor and Gradient Boosting Regressor, based on various evaluation metrics. Further, we compare each model concerning one evaluation metric at a time.

CHAPTER 4

SYSTEM STUDY

4.1. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth N with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a

burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- i. Economical Feasibility
- ii. Technical Feasibility
- iii. Social Feasibility

4.1.1. Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.1.2. Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.1.3. Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub – assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1. TYPES OF TESTS

5.1.1. 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 and 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. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.1.2. 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.

5.1.3. FUNCTIONAL TEST

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

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.1.4. SYSTEM TEST

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.

5.1.5. WHITE BOX TESTING

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

5.1.6. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.1.7. UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

RESULT :

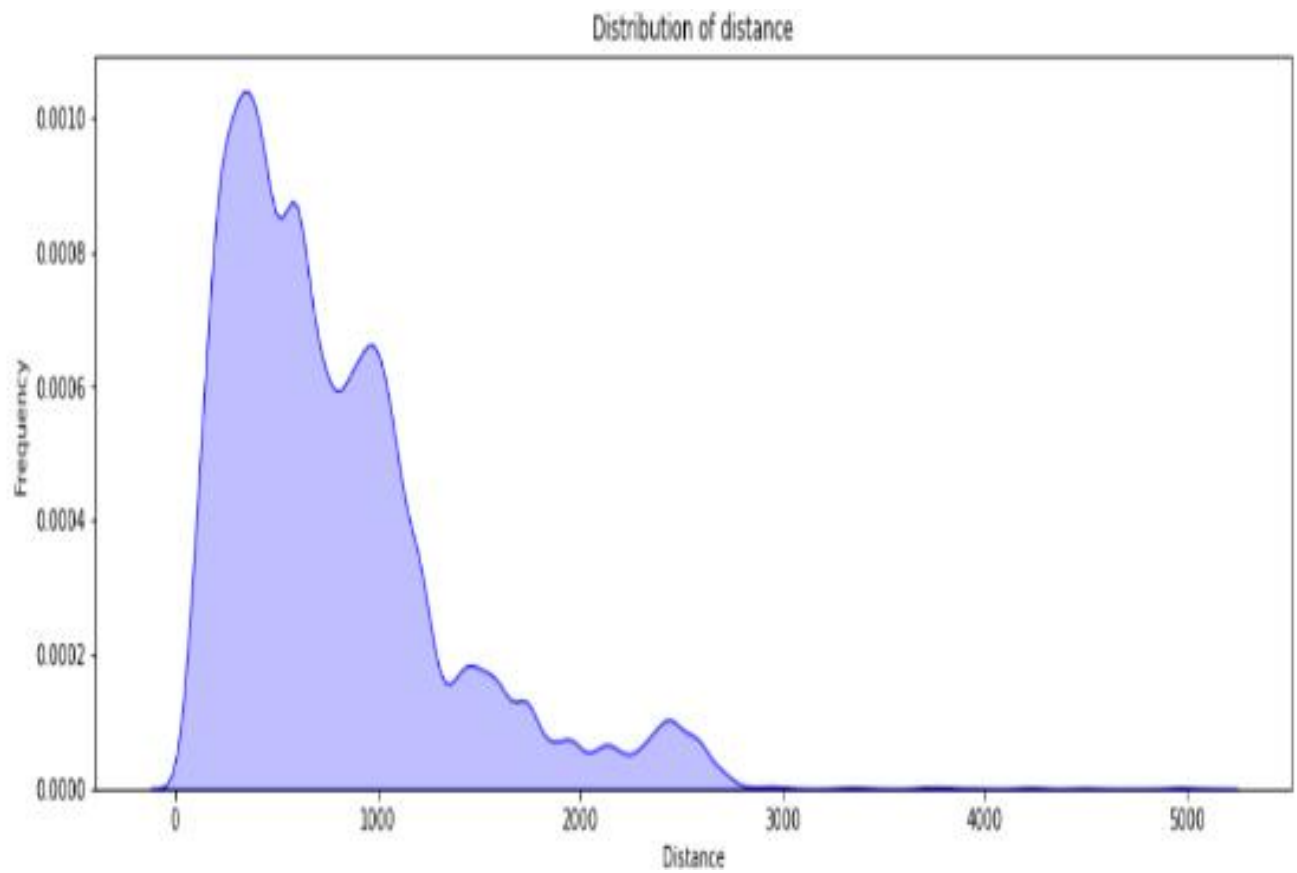


Fig 5. Variable distribution's distance

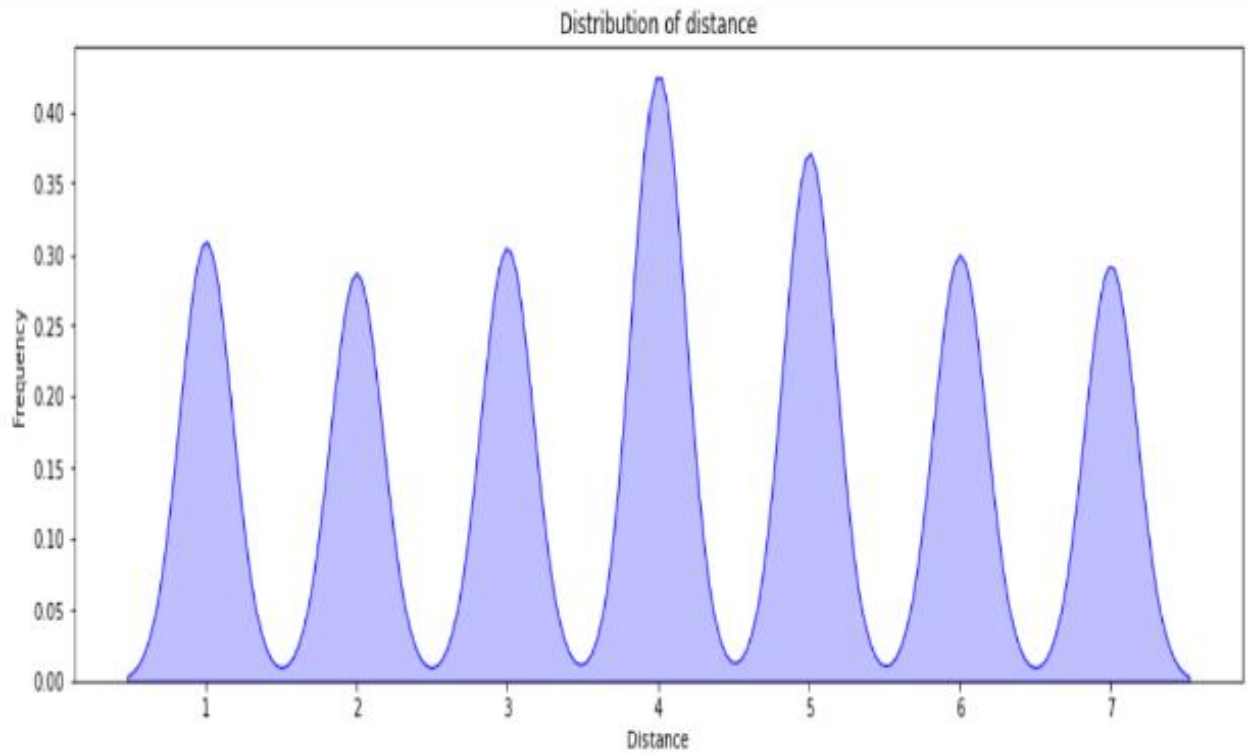


Fig 6. The week's day

By using Pandas, NumPy, Statistical Methods, and Data Visualisation packages, this is an unavoidable and important step in fine-tuning the dataset in a different method of evaluation to understand the additional insight of the key characteristics of various organisations of the data set such as column(s), row(s), and data visualisation packages.

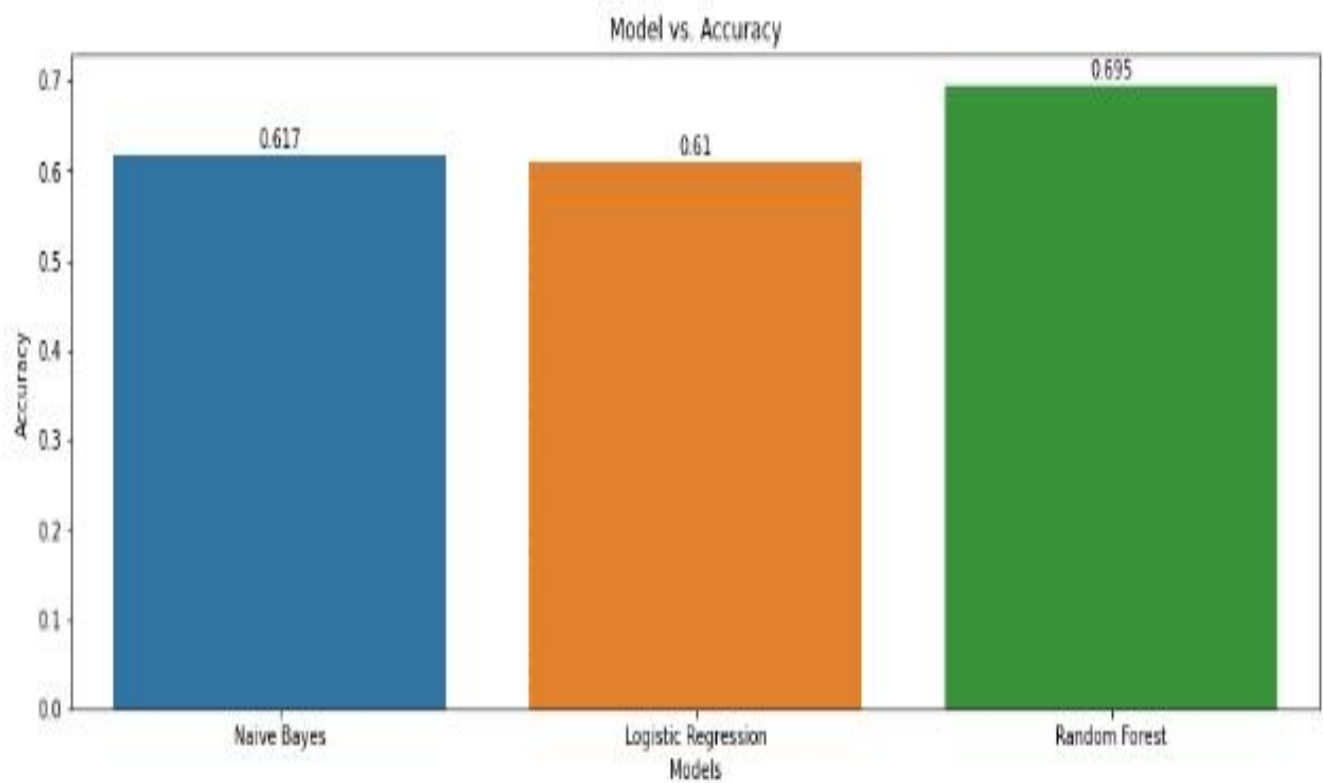


Fig.7. Graph of accuracy algorithms

- i. Logistic regression is a technique for supervised machine learning. Random Forest is a technique for learning in groups.
- ii. The Gaussian Nave Bayes classifier is a common machine learning classifier that is both easy and effective to use.
- iii. A training algorithm for learning classification and regression rules from data is referred to as logistic regression.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Machine learning algorithms were applied progressively and successively to predict flight arrival & delay. We built five models out of this. We saw for each evaluation metric considered the values of the models and compared them. We found out that: -

In Departure Delay, Random Forest Regressor was observed as the best model with Mean Squared Error 2261.8 and Mean Absolute Error 24.1, which are the minimum value found in these respective metrics. In Arrival Delay, Random Forest Regressor was the best model observed with Mean Squared Error 3019.3 and Mean Absolute Error 30.8, which are the minimum value found in these respective metrics.

In the rest of the metrics, the value of the error of Random Forest Regressor although is not minimum but still gives a low value comparatively. In maximum metrics, we found out that Random Forest Regressor gives us the best value and thus should be the model selected.

The future scope of this paper can include the application of more advanced, modern and innovative pre-processing techniques, automated hybrid learning and sampling algorithms, and deep learning models adjusted to achieve better performance. To evolve a predictive model, additional variables can be introduced. e.g., a model where meteorological statistics are utilized in developing error-free models for flight delays. In this paper we used data from the US only, therefore in future, the model can be trained with data from other countries as well. With the use of models that are complex and hybrid of many other models provided with appropriate processing power and with the use of larger detailed datasets, more accurate predictive models can be developed. Additionally, the model can be configured for other airports to predict their flight delays as well and for that data from these airports would be required to incorporate into this research.

6.2.APPENDIX:

6.2.1 SAMPLE CODE

```
#import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
```

```

#pip install numpy
from sklearn.model_selection import train_test_split
from sklearn import metrics
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SimpleRNN, SpatialDropout1D
flight = pd.read_csv("Tweets.csv")
flight
flight.shape
flight.head()
flight.tail()
flight.describe()
flight.info()
flight.isnull().sum()
flight = flight[flight['airline_sentiment_confidence'] > 0.6]
flight
flight = flight[['text', 'airline_sentiment']]
flight.head()
def clean_train_data(x):
    text = x
    text = text.lower()
    text = re.sub('\[.*?\]', "", text) # remove square brackets
    text = re.sub(r'^\w\s|$', "", text) # remove punctuation
    text = re.sub('\w*\d\w*', "", text) # remove words containing numbers
    text = re.sub('\n', "", text)
    return text
flight['text'] = flight.text.apply(lambda x : clean_train_data(x))
flight.head()
data = flight.copy()
data
flight = flight[flight['airline_sentiment'] != 'neutral']
flight.head()
print("POsitive:",len(flight[flight['airline_sentiment'] == 'positive']))

```

```
print("\nNegative",len(flight[ flight['airline_sentiment'] == 'negative']))
```

```
print("\nNeutral",len(flight[ flight['airline_sentiment'] == 'neutral']))
```

```
model1_data = flight.copy()
```

```
model1_data
```

```
max_features = 2000
```

```
token = Tokenizer(num_words=max_features, split = ' ')
```

```
token.fit_on_texts(flight['text'].values)
```

```
X = token.texts_to_sequences(flight['text'].values)
```

```
X = pad_sequences(X)
```

```
X.shape
```

```
embed_dim = 128
```

```
lstm_out = 196
```

```
model = Sequential()
```

```
model.add(Embedding(max_features, embed_dim, input_length = X.shape[1]))
```

```
model.add(SpatialDropout1D(0.4))
```

```
model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
```

```
model.add(Dense(2, activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
model.summary()
```

```
Y = pd.get_dummies(flight['airline_sentiment']).values
```

```
Y.shape
```

```
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size=0.33, random_state=42)
```

```
X_train
```

```
X_test
```

```
y_train
```

```
y_test
```

```
batch_size = 25
```

```
history = model.fit(X_train, y_train, epochs=10, batch_size=batch_size, verbose=2)
```

```
# score = model.predict(X_test)
```

```
score, acc = model.evaluate(X_test, y_test, batch_size=batch_size, verbose=2)
```

```
print('score', score)
```

```

print('accuracy', acc)
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import emoji
#!pip install emoji
!pip install catboost
from sklearn.model_selection import train_test_split, GridSearchCV
from catboost import CatBoostClassifier, Pool
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
data = pd.read_csv('2020_feb_flight_delay.csv')
data
data.shape
data.head()
data.tail()
data.info()
data.describe()
data.columns
data.notnull().sum()
#drop unnamed column
data = data.drop(['Unnamed: 9'],axis=1)
data
# distribution of our target
data['DEP_DEL15'].value_counts()
# Split the data into positive and negative
positive_rows = data.DEP_DEL15 == 1.0
data_pos = data.loc[positive_rows]
data_neg = data.loc[~positive_rows]
positive_rows.shape

```

```

data_neg.shape
data_pos.shape
data.shape
# Merge the balanced data
data = pd.concat([data_pos, data_neg.sample(n = len(data_pos))], axis = 0)
data
# Shuffle the order of data
data = data.sample(n = len(data)).reset_index(drop = True)
data
data.isna().sum()
data = data.dropna(axis=0)
data
data.info()
plt.figure(figsize=(15,5))
sns.distplot(data['DISTANCE'], hist=False, color="b", kde_kws={"shade": True})
plt.xlabel("Distance")
plt.ylabel("Frequency")
plt.title("Distribution of distance")
plt.show()
plt.figure(figsize=(15,5))
sns.distplot(data['DEP_TIME'], hist=False, color="b", kde_kws={"shade": True})
plt.xlabel("Distance")
plt.ylabel("Frequency")
plt.title("Distribution of distance")
plt.show()
plt.figure(figsize=(15,5))
sns.distplot(data['DEP_DEL15'], hist=False, color="b", kde_kws={"shade": True})
plt.xlabel("Distance")
plt.ylabel("Frequency")
plt.title("Distribution of distance")
plt.show()
plt.figure(figsize=(15,5))
sns.distplot(data['DAY_OF_WEEK'], hist=False, color="b", kde_kws={"shade": True})
plt.xlabel("Distance")
plt.ylabel("Frequency")

```

```

plt.title("Distribution of distance")
plt.show()
print(f"Average distance for delay {data[data['DEP_DEL15'] == 1]['DISTANCE'].values.mean()} miles")
print(f"Average distance for no delay {data[data['DEP_DEL15'] == 0]['DISTANCE'].values.mean()} miles")
#Count of carriers in the dataset
plt.figure(figsize=(15,10))
sns.countplot(x=data['OP_UNIQUE_CARRIER'], data=data)
plt.xlabel("Carriers")
plt.ylabel("Count")
plt.title("Count of unique carrier")
plt.show()
plt.figure(figsize=(15,10))
sns.countplot(x=data['DAY_OF_WEEK'], data=data)
plt.xlabel("Day of Week")
plt.ylabel("Count")
plt.title("Count of Day of Week")
plt.show()
data = data.rename(columns={'DEP_DEL15':'TARGET'})
data
def label_encoding(categories):
    """
    To perform mapping of categorical features
    """
    categories = list(set(list(categories.values)))
    mapping = {}
    for idx in range(len(categories)):
        mapping[categories[idx]] = idx
    return mapping
data['OP_UNIQUE_CARRIER'] =
data['OP_UNIQUE_CARRIER'].map(label_encoding(data['OP_UNIQUE_CARRIER']))
data.head()
data['ORIGIN'] = data['ORIGIN'].map(label_encoding(data['ORIGIN']))
data.head()

```

```

data['DEST'] = data['DEST'].map(label_encoding(data['DEST']))
data.head()
data['TARGET'].value_counts()
X = data[['DAY_OF_MONTH', 'DAY_OF_WEEK', 'OP_UNIQUE_CARRIER', 'ORIGIN',
'DEST', 'DEP_TIME', 'DISTANCE']].values
y = data[['TARGET']].values
X.shape
y.shape
# Splitting Train-set and Test-set
X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.2, random_state=41)

# Splitting Train-set and Validation-set
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25,
random_state=41)
X_train.shape
X_val.shape
y_train.shape
y_val.shape
# Formula to get accuracy
def get_accuracy(y_true, y_preds):
    # Getting score of confusion matrix
    true_negative, false_positive, false_negative, true_positive = confusion_matrix(y_true,
y_preds).ravel()
    # Calculating accuracy
    accuracy = (true_positive + true_negative)/(true_negative + false_positive +
false_negative + true_positive)
    return accuracy
gnb = GaussianNB()
gnb.fit(X_train, y_train)
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state=0).fit(X_train, y_train)
rf = RandomForestClassifier(random_state=0)
rf.fit(X_train, y_train)
models = [gnb,lr,rf]
acc = []

```

```

for model in models:
    preds_val = model.predict(X_val)
    accuracy = get_accuracy(y_val, preds_val)
    acc.append(accuracy)
model_name = ['Naive Bayes', 'Logistic Regression', 'Random Forest']
accuracy = dict(zip(model_name, acc))
print(accuracy)
plt.figure(figsize=(15,5))
ax = sns.barplot(x = list(accuracy.keys()), y = list(accuracy.values()))
for p, value in zip(ax.patches, list(accuracy.values())):
    _x = p.get_x() + p.get_width() / 2
    _y = p.get_y() + p.get_height() + 0.008
    ax.text(_x, _y, round(value, 3), ha="center")
plt.xlabel("Models")
plt.ylabel("Accuracy")
plt.title("Model vs. Accuracy")
plt.show()
test_preds = rf.predict(X_test)
get_accuracy(y_test, test_preds)

```

Flight delays

ORIGINALITY REPORT

14%

SIMILARITY INDEX

8%

INTERNET SOURCES

10%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	Priyanka Meel, Mukul Singhal, Mukul Tanwar, Naman Saini. "Predicting Flight Delays with Error Calculation using Machine Learned Classifiers", 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020 Publication	3%
2	www.jetir.org Internet Source	3%
3	www.coursehero.com Internet Source	2%
4	Submitted to Asian Institute of Technology Student Paper	2%
5	Submitted to University of Stirling Student Paper	1%
6	Chuanzhi Zhuang, Ningyu Zhang, Xiaolong Jin, Zhong Li, Shumin Deng, Huajun Chen. "Joint Extraction of Triple Knowledge Based on Relation Priority", 2020 IEEE Intl Conf on Parallel & Distributed Processing with	1%

Applications, Big Data & Cloud Computing,
Sustainable Computing & Communications,
Social Computing & Networking
(ISPA/BDCloud/SocialCom/SustainCom), 2020

Publication

7	Submitted to National College of Ireland Student Paper	1 %
8	link.springer.com Internet Source	<1 %
9	www.techscience.com Internet Source	<1 %
10	journalofbigdata.springeropen.com Internet Source	<1 %
11	ijrsrset.com Internet Source	<1 %
12	www.cister.isep.ipp.pt Internet Source	<1 %
13	www.hindawi.com Internet Source	<1 %
14	"Intelligent Systems in Big Data, Semantic Web and Machine Learning", Springer Science and Business Media LLC, 2021 Publication	<1 %

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

Error Calculation for Prediction of Flight Delays using Machine Learned Classifiers

Baranidaran GT, Balamurugan R

2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

GENERAL TERMS

1. The undersigned represents that he/she has the power and authority to make and execute this form.
2. The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
3. The undersigned agrees that publication with IEEE is subject to the policies and procedures of the [IEEE PSPB Operations Manual](#).
4. In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall be null and void. In this case, IEEE will retain a copy of the manuscript for internal administrative/record-keeping purposes.
5. For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
6. The author hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the author has obtained any necessary permissions. Where necessary, the author has obtained

all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE

You have indicated that you DO wish to have video/audio recordings made of your conference presentation under terms and conditions set forth in "Consent and Release."

CONSENT AND RELEASE

1. In the event the author makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the author, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.
2. In connection with the permission granted in Section 1, the author hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON,
YOU CERTIFY THAT SUCH ACTION

CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE
WITH UNITED STATES LAW, WHICH

AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A
USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

Baranidaran GT

Signature

08-03-2022

Date (dd-mm-yyyy)

AUTHOR RESPONSIBILITIES

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at

http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html

Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

RETAINED RIGHTS/TERMS AND CONDITIONS

- Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
- Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
- Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.
- Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

AUTHOR ONLINE USE

- **Personal Servers.** Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
- **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's

teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.

- **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

Questions about the submission of the form or manuscript must be sent to the publication's editor.

Please direct all questions about IEEE copyright policy to:

IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966

Letter of Acceptance

To

Balamurugan R, Maria Anu V, Baranidaran GT, L.MaryGladence, S.Revathy
Dept. of IT, Sathyabama Institute of Science and Technology.

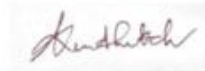
Greetings!!

On behalf of the SCAD College of Engineering and Technology Tirunelveli, we take great pleasure in inviting you to attend the “**6th International Conference on Trends in Electronics and Informatics (ICOEI 2022)**” which is going to be held during **28-30, April 2022** at Tirunelveli, India.

We are glad to inform you that your submitted article entitled “**Error Calculation for Prediction of Flight Delays using Machine Learned Classifiers**” [**Paper ID: ICOEI-091**] has been accepted for publication at ICOEI 2022. We welcome you to join us and share your research and views. This conference will offer you an unforgettable experience in exploring new opportunities.

With Thanks,

Yours' Sincerely



Dr. R. Karthik Ganesh,
Organizing Secretary
ICOEI 2022.

Previous Publications

[ICOEI 2021](#)
[ICOEI 2020](#)
[ICOEI 2019](#)
[ICOEI 2018](#)
[ICOEI 2017](#)

Proceedings by



REFERENCES

1. Chakrabarty, Navoneel, Tuhin Kundu, Sudipta Dandapat, Apurba Sarkar, and Dipak Kumar Koley. "Flight arrival delay prediction using gradient boosting classifier." In *Emerging Technologies in Data Mining and Information Security*, pp. 651-659. Springer, Singapore, 2019.
2. Chakrabarty, Navoneel. "A data mining approach to flight arrival delay prediction for american airlines." In *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, pp. 102-107. IEEE, 2019.
3. Kim, Y.J., Choi, S., Briceno, S. and Mavris, D., 2016, September. A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* (pp. 1-6). IEEE.
4. Sternberg, A., Soares, J., Carvalho, D. and Ogasawara, E., 2017. A review on flight delay prediction. arXiv preprint arXiv:1703.06118.
5. Ding, Y., 2017, August. Predicting flight delay based on multiple linear regression. In *IOP conference series: Earth and environmental science* (Vol. 81, No. 1, p. 012198). IOP Publishing.
6. Manna, Suvojit, Sanket Biswas, Riyanka Kundu, Somnath Rakshit, Priti Gupta, and Subhas Barman. "A statistical approach to predict flight delay using gradient boosted decision tree." In *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pp. 1-5. IEEE, 2017.
7. Dou, Xiaotong. "Flight arrival delay prediction and analysis using ensemble learning." In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, pp. 836-840. IEEE, 2020.
8. Chen, Jun, and Meng Li. "Chained predictions of flight delay using machine learning." In *AIAA Scitech 2019 forum*, p. 1661. 2019.
9. Rodríguez-Sanz, Álvaro, Fernando Gómez Comendador, Rosa Arnaldo Valdés, Javier Pérez-Castán, Rocío Barragán Montes, and Sergio Cámara Serrano. "Assessment of airport arrival congestion and delay: Prediction and reliability." *Transportation Research Part C: Emerging Technologies* 98 (2019): 255-283.
10. Kuhn, Nathalie, and Navaneeth Jamadagni. "Application of machine learning algorithms to predict flight arrival delays." *CS229* (2017).