A Project Report on

Predicting flight delays with error calculation using machine learning

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

in partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Allenki Krupa (18BD1A05A3)

C Soummith (18BD1A05AA)

G Mary Ashwitha (18BD1A05AG)

Preksha Addagatla (18BD1A05B9)

Under the guidance of

T Rupa Devi
Assistant Professor
Department of CSE

Department of Computer Science and Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad – 500029

2021-2022

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Accredited by NBA & NAAC, Approved By A.I.C.T.E., Reg by Govt of Telangana State & Affiliated to JNTU, Hyderabad)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **PREDICTING FLIGHT DELAYS WITH ERROR CALCULATION USING MACHINE LEARNING** being submitted by

A Krupa	**(18BD1A05A3)**
C Soummith	**(18BD1A05AA)**
G Mary Ashwitha	**(18BD1A05AG)**
Preksha A	**(18BD1A05B9)**

In partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** affiliated to the **Jawaharlal Nehru Technological University, Hyderabad** during the year 2021-22**.**

Internal Guide **Head of the Department**

(Mrs. T Rupa Devi) **(Dr. S. Padmaja)**

Submitted for Viva Voce Examination held on _____

Internal Examiner

Vision of KMIT

Producing quality graduates trained in the latest technologies and related tools and striving to make India a world leader in software and hardware products and services. To achieve academic excellence by imparting in depth knowledge to the students, facilitating research activities and catering to the fast growing and ever- changing industrial demands and societal needs.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish industry institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create newenterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand lndia'schallenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

PROGRAM OUTCOMES (POs)

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. Design/development of solutions: Design solutions for complex engineering problem and design system component or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation make effective presentations and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage

projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analysis the common business functions to design and develop appropriate Information Technology solutions for social up liftment.

PSO2: Shall have expertise on the evolving technologies like Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES

P1: Compares all models and finds the model with best accuracy.

P2: Helps the customers and airport authorities predict flight delays.

P3: Interactive web interface that helps the authorities to predict the delay

P4: Gives accurate time of the flight delay with given inputs

L –LOW

M – MEDIUM

H –HIGH

PROJECT OUTCOMES MAPPING PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	2	2	-	2	2	-	-	-	2	-	1	1
P2	-	-	-	2	1	-	-	-	1	-	1	1
P3	-	-	-	2	1	-	-	-	2	-	1	2
P4	2	2	-	2	2	-	-	-	2	-	1	1

PROJECT OUTCOMES MAPPING PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	1	2
P2	2	2
P3	2	2
P4	1	2

PROJECT OUTCOMES MAPPING PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1	1	2	1	1
P2	1	1	1	1
P3	1	1	1	1
P4	1	2	1	1

DECLARATION

We hereby declare that the Project Stage-1 report entitled "**PREDICTING FLIGHT DELAYS WITH ERROR CALCULATION USING MACHINE LEARNING"** is done in the partial fulfillment for the award of the Degree in Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

ALLENKI KRUPA (18BD1A05A3)

CHALLA SOUMMITH (18BD1A05AA)

GOPU MARY ASHWITHA (18BD1A05AG)

PREKSHA ADDAGATLA (18BD1A05B9)

ACKNOWLEDGMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. S. Anuradha**, Dean Academics for providing an excellent environment in the college.

We are also thankful to **Dr. S. Padmaja**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Mrs. T Rupa Devi**, for his/her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

ALLENKI KRUPA (18BD1A05A3)

CHALLA SOUMMITH (18BD1A05AA)

GOPU MARY ASHWITHA (18BD1A05AG)

PREKSHA ADDAGATLA (18BD1A05B9)

CONTENT

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 this paper, 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.

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

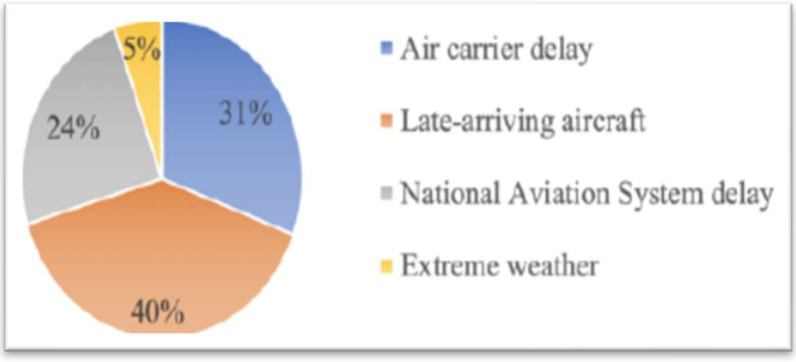
INTRODUCTION

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 [1] 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 Problem Statement

- Late-arriving aircraft: A single aircraft flies multiple flight legs each day in order to increase its utilization. Aircraft in the US typically operate 4-6 flight legs a day.
- Air carrier delays: This category includes all causes of delay that are considered to be within the control of an airline.
- NAS delays: These are delays due to air traffic control and traffic management initiatives.
- Extreme weather: This category includes severe meteorological conditions.

Fig. 1.1 Causes of flight delay in the US



1.2 Existing System

There have been too many studies in this area. For example, older Regression method has been used to compute delay propagation. For this model, the destination delay is highly dependent to arrival flights and the effective factors include; day, time, airport capacity and some factors are related to passenger loads. In addition, as the problem neglects the weather conditions, this model shows inefficiency in U.S.A but it is suitable for Europe. Where, only 1–4% of the Europe flights delayed due to weather condition, this value for U.S.A is between 70 and 75% an intelligent neural network has been designed which estimated the destination delay for actual applications in controlling traffic progress. This model employs factors of airport type, airplane type, date, and time, and flight path, flight frequency for network training and non-linear and linear for data analysis. As it is difficult to interpret neural network parameters, the way factor behavior and most important verification of the most important factors in flight is extremely difficult. Furthermore, older intelligent algorithm usually uses shadow learning models to solve conditions with a big data in complicated classifications. However, results of this analysis are very different with respect to ideal condition. Although model design can have a good or bad situation, response is highly dependent to experience and even happenstance and this procedure require too much time. Therefore, traditional simulation and modeling techniques is not suitable or even efficient for such problems. There is an ongoing subject of study which solves this problem and this paper also has tried to use that subject in modeling.

1.3 Proposed System

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 labeling along with airline airborne time are also provided. The data set consists of 25 columns and 59986 rows. The figure 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.

1.4 Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

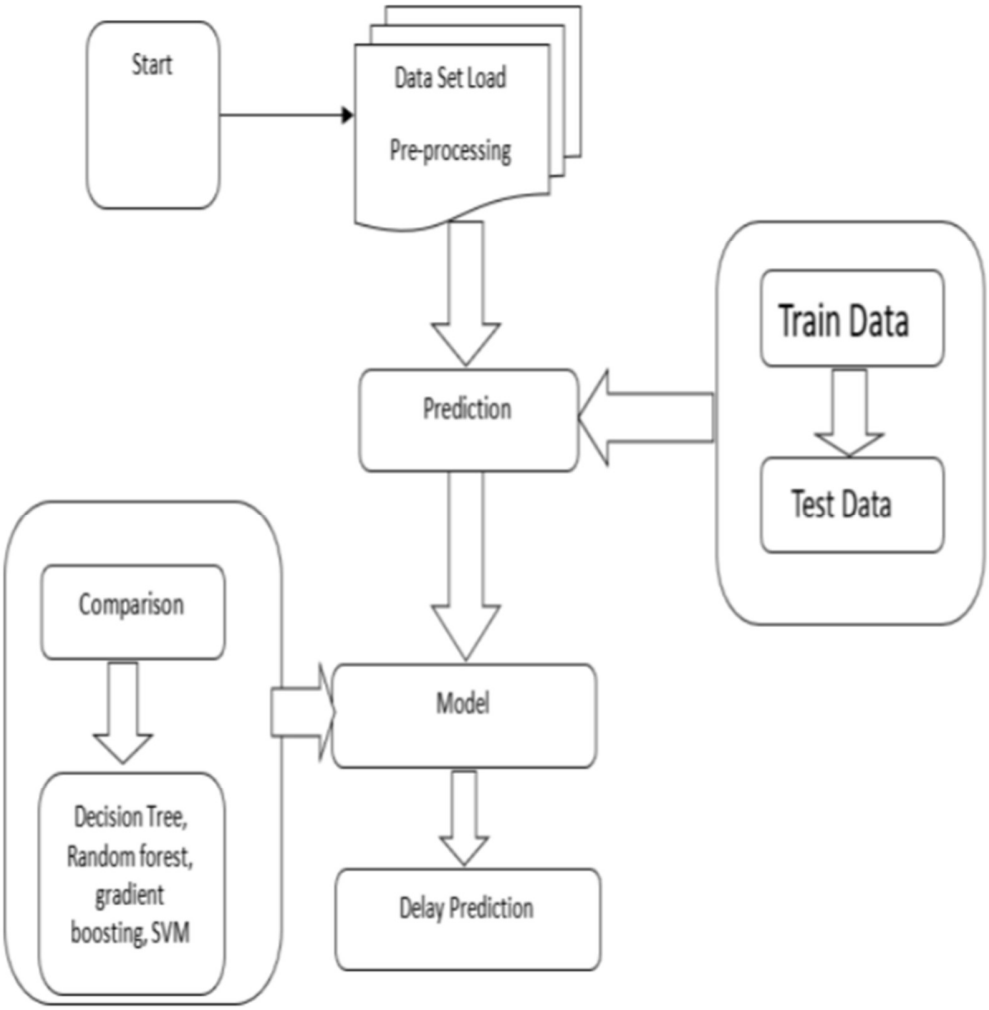
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

1.5 : Architecture diagram

In this section, we describe our technique for predicting flight arrival and departure time with error calculations using machine learning. The working process of the system is shown in the figure. When the program gets started the data set will be loaded and then will be pre-processed. the data will be trained and tested every time. Then will be sent to the prediction process. Comparison will be performed and the algorithm will be performed separately and the output will be fed to the model. Finally, delay prediction will be shown.

Fig. 1.2 Architecture Diagram



CHAPTER – 2

SOFTWARE REQUIREMENTS SPECIFICATIONS

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

2.1 System Analysis

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

Analysis:

Although the scale of this project is relatively small, to produce a professional solution is it imperative that the current problem is understood accurately. However, this task has been made doubly difficult by the lack of support from the company. Thankfully, the Application manager has been kind enough to spare me some of his own time to discuss the problem with me further. Therefore, this chapter is concerning with analyzing the current situation and expectations of the user for this system.

Requirements:

The minimum requirements of the project are listed below:

- Examine the tools and methodologies required to gain an overview of the system requirements for the proposed database.

- Examine suitable database management systems that can be used to implement the proposed database.
- Evaluate appropriate website authoring and web graphic creation tools that can be used to develop web-based forms for the proposed database
- Produce and apply suitable criteria for evaluating the solution

2.2 Requirement Analysis

Taking into account the comparative analysis stated in the previous section we could start specifying the requirements that our website should achieve. As a basis, an article on all the different requirements for software development was taken into account during this process. We divide the requirements in 2 types: functional and nonfunctional requirements.

2.3 Functional requirements

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet.
Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.
The functional specification describes what the system must do, how the system does it is described in the design specification. If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

2.4 Nonfunctional requirements

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e., how fast the system reacts to user commands.) or accuracy (i.e., how precise are the systems numerical answers.).

- Portability
- Reliability
- Usability
- Time Constraints
- Error messages
- Actions which cannot be undone should ask for confirmation
- Responsive design should be implemented
- Space Constraints
- Performance
- Standards
- Ethics
- Interoperability
- Security
- Privacy
- Scalability

2.5 Software requirements

- Programming Language : Python
- IDE : PyCharm/Jupyter

2.6 Hardware requirements

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

CHAPTER – 3

LITERATURE SURVEY

Much research has been done on studying flight delays. The prediction, analysis and cause of flight delays have been a major problem for air traffic control, decision-making by airlines and ground delay response programs. Studies are conducted on the delay propagation of the sequence. Also, studying the predictive model of arrival delay and departure delay with meteorological features is encouraged. In the past, researchers have tried to predict flight delays with Machine Learning. Chakrabarty et al. used supervised automatic learning algorithms (random forest, Gradient Boosting Classifier, Support Vector Machine and the k-nearest neighbor algorithm) to predict delays in the arrival of operated flights including the five busiest US airports. The maximum precision achieved was 79.7% with gradient booster as a classifier with a limited data set. Choi et al. applied machine learning algorithms like decision tree, random forest, AdaBoost and kNearest Neighbours to predict delays on individual flights. Flight schedule data and weather forecasts have been incorporated into the model. Sampling techniques were used to balance the data and it was observed that the accuracy of the classifier trained without sampling was more that of the trained classifier with sampling techniques. Cao et al. used a Bayesian Network model to analyze the turnaround time of a flight and delay prediction.

Juan José Rebollo and Hamsa Balakrishnan [8] used a hundred pairs of origin and destination to summarize the result of various regression and classification models. The find outs reveal that among all the methods used, random forest has the highest performance. However, predictability may additionally range because of factors such as the number of origin destination pairs and the forecast horizon. Sruti Oza, Somya Sharma [9] used multiple linear regression to predict weather induced flight delays in flight-data, as well as climatic factors and probabilities due to weather delays. The forecasts were based on some key attributes, such as carrier, departure time, arrival time, origin and destination. Anish M. Kalliguddi and Aera K. Leboulluec predicted both departure and arrival delays using regression models such as Decision Tree Regressor, Multiple Linear Regression and Random Forest Regressor in flight-data. It has been observed that the longer forecast horizon is useful for increasing the accuracy with a minimum forecast error for random

forests. Etani J Big Data A supervised model of on-schedule arrival fight is used using weather data and flight data. The relationship between flight data and pressure patterns of Peach Aviation is found. On-Schedule arrival flight is predicted with 77% accuracy using Random Forest as a Classifier.

1. Shao, Wei, et al. "Flight Delay Prediction using Airport Situational Awareness Map." Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2019.

- Wei shao and eight others worked on this paper for predicting flight delays with help of the airport situational awareness map.
- They mainly used a Data-driven framework where it is suitable for prediction of departure flight delays and also explored other different features that are taken from the awareness map.

2. Raj, Jennifer S., and J. Vijitha Ananthi. "Recurrent neural networks and nonlinear prediction in support vector machines." Journal of Soft Computing Paradigm (JSCP) 1.01 (2019): 33-40.

- Dr. Jennifer S. Raj and J. Vijitha Ananthi proposed to use an SVM optimized approach to RNN for solving nonlinear regression estimation.
- This model improved the speed and accuracy in identifying optimal values of the SVM parameters.

3. Ye, Bojia , et al. "A Methodology for Predicting Aggregate Flight Departure Delays in Airports Based on Supervised Learning." Sustainability 12.7 (2020): 2749.

- Predicting delay among flights using Supervised Learning model built by with local weather characteristics, Airport related aggregate, time, flight-plan and delay.

4. Musaddi , Roshni, et al. "Flight Delay Prediction using Binary Classification."

- This paper proposes Binary Classification for predicting delay in flight take offs.

- We have taken datasets of different airlines from different airports to check which one gets delayed more often.
- The delay can be due to climate conditions or traffic in airspace and airports or any other reason.

5. Khanmohammadi, Sina, Salih Tutun, and Yunus Kucuk . "A new multilevel input layer artificial neural network for predicting flight delays at JFK airport." Procedia Computer Science 95 (2016): 237-244.

- A new ANN structure (DMP-ANN) is introduced which is suitable for prediction of defects such as delays in operations.
- This structure is appropriate for problems with nominal variables, where traditional ANN models have difficulties.

6. Ding, Yi. "Predicting flight delay based on multiple linear regression." IOP Conference Series: Earth and Environmental Science. Vol. 81.No. 1.IOP Publishing, 2017.

- This paper considered the problem of predicting flight arrival delay and presented a prediction result.
- The problem was treated as both a regression and an ordinal classification task and a suitable approach, based on the multiple linear regression model, was used to predict the delay.

7. Kalliguddi, Anish M., and Aera K. Leboulluec. "Predictive modeling of aircraft flight delay." Universal Journal of Management 5.10 (2017): 485-491.

- This study is devoted to develop a predictive model to forecast flight delays.

- Models based on multiple linear regression, decision trees and random forest algorithms are created and tested in R-studio software concluding that Random forest model outperforms other two models based on the evaluation criteria.

8. Airline Delay Prediction using Machine Learning Techniques. Devansh Shah, Ayushi Lodaria, Danish Jain, Lynette D'Mello

- This study shows that machine learning algorithms can be efficaciously used to predict flight delays.
- The purpose of doing the above classification and analysis, is to gauge the delay not only to suffice the various purposes of mankind, but also analyze factors affecting delay such as "Weather Delay", "NAS Delay".

CHAPTER – 4

SYSTEM DESIGN

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

4.1 Software Design

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consist of hierarchy of modules and serve to partition into separate function.

2. **Coupling:** modules should have little dependence on other modules of a system.

3. **Cohesion:** modules should carry out in a single processing function.

4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provides.

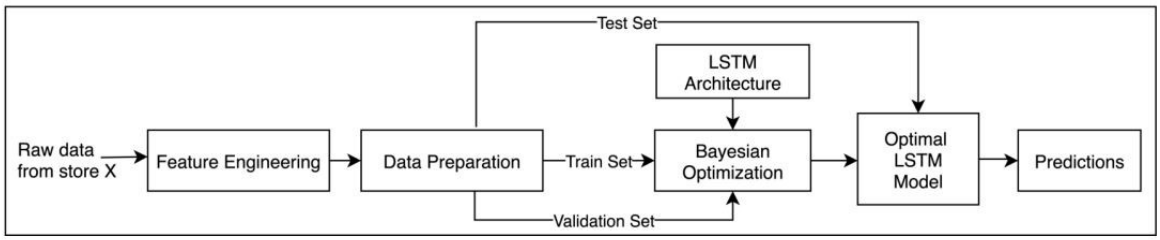
4.2 Data Flow Diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

Fig. 4.1 Data Flow Diagram



4.3 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

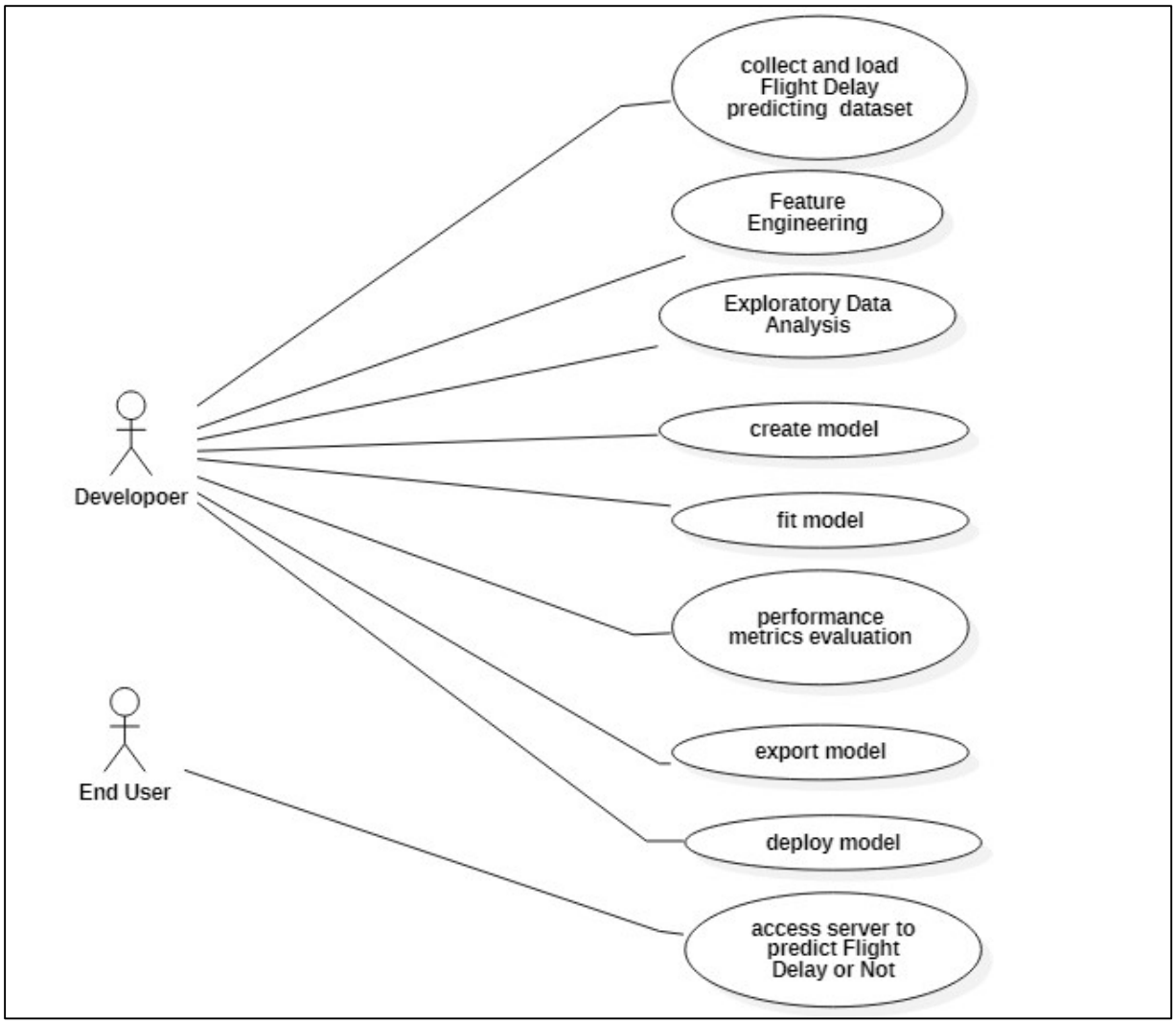
The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

4.4 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

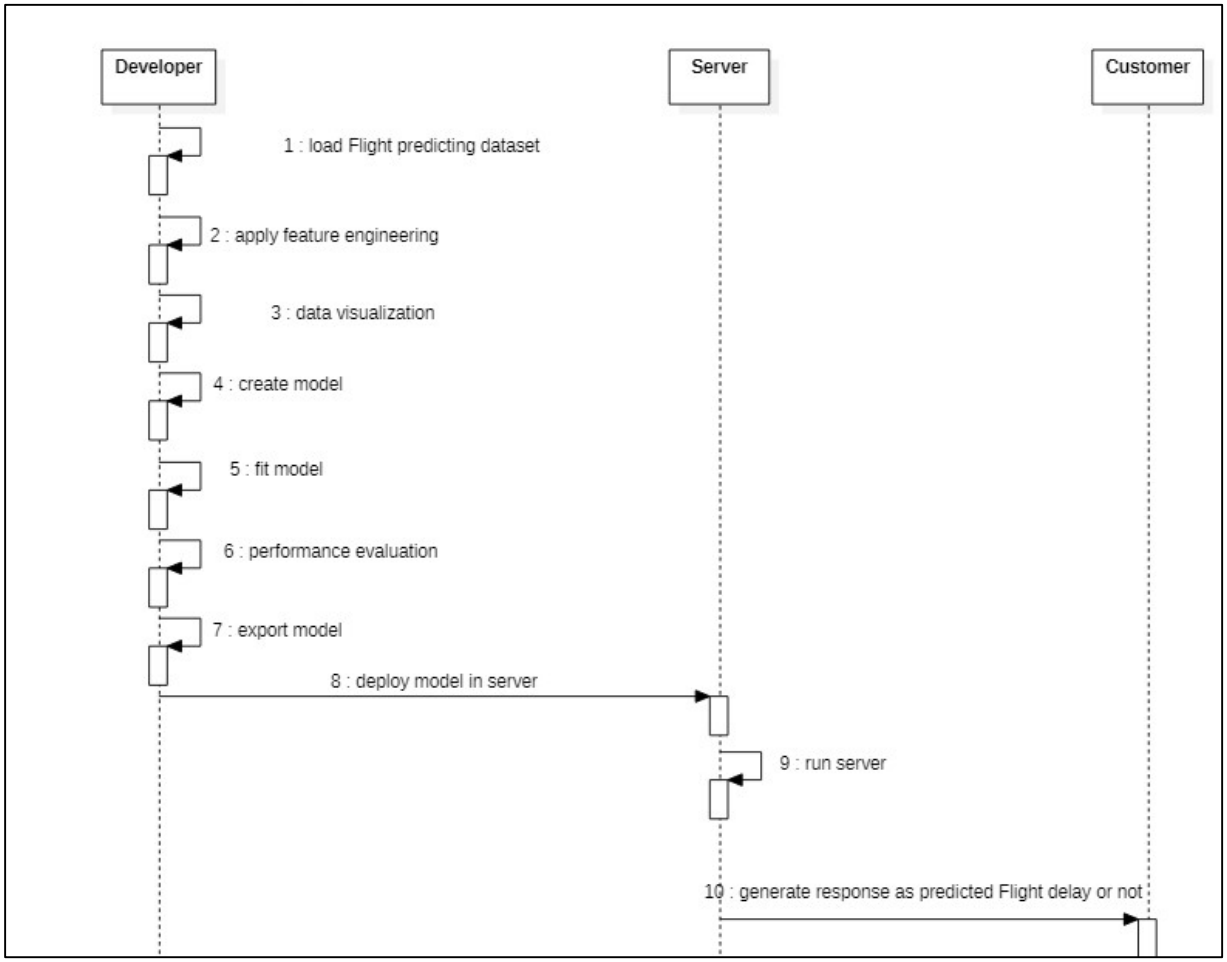
Fig. 4.2 Use Case Diagram



4.5 Sequence diagram:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment.

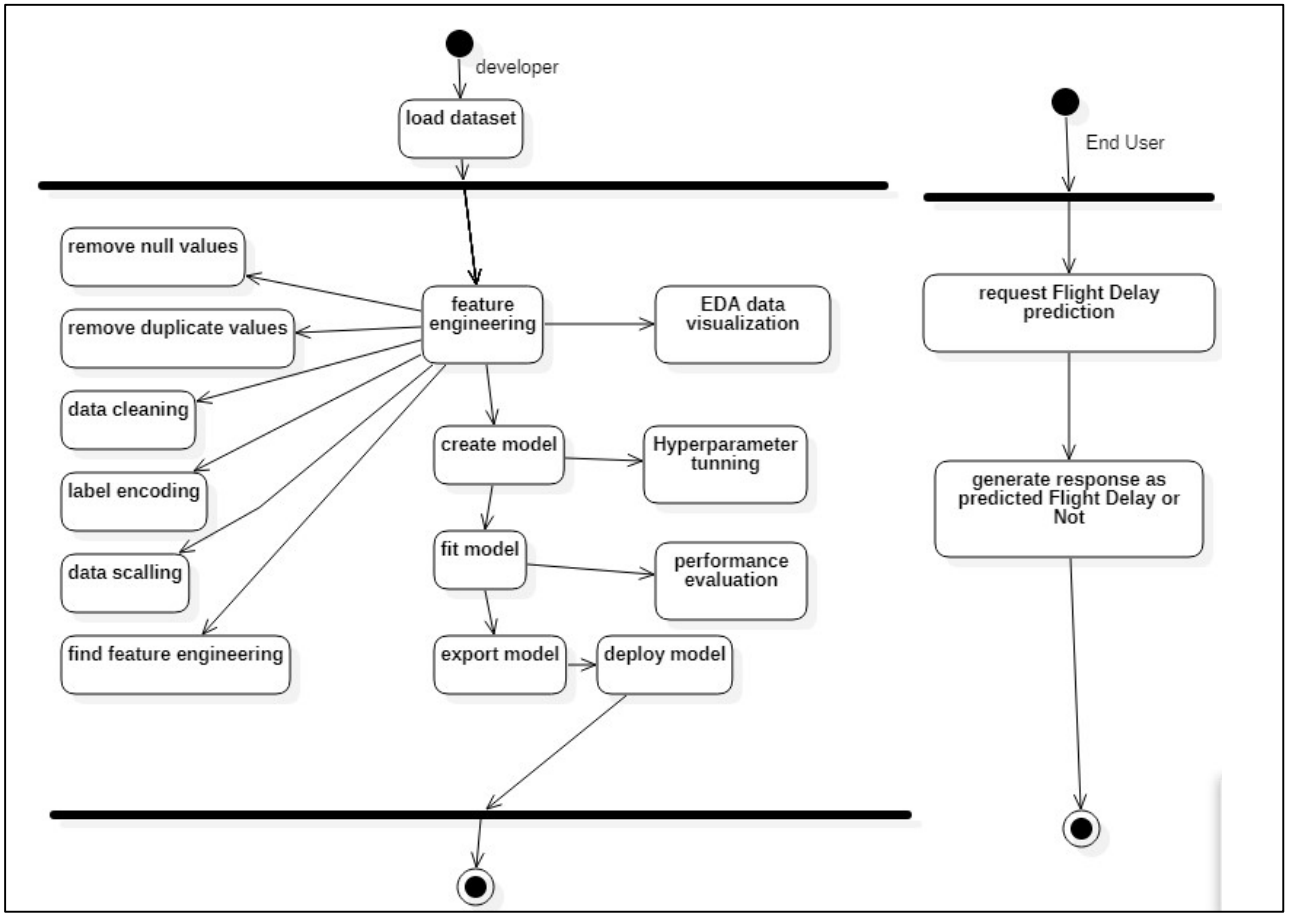
Fig. 4.3 Sequence Diagram



4.6 Activity diagram:

•Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

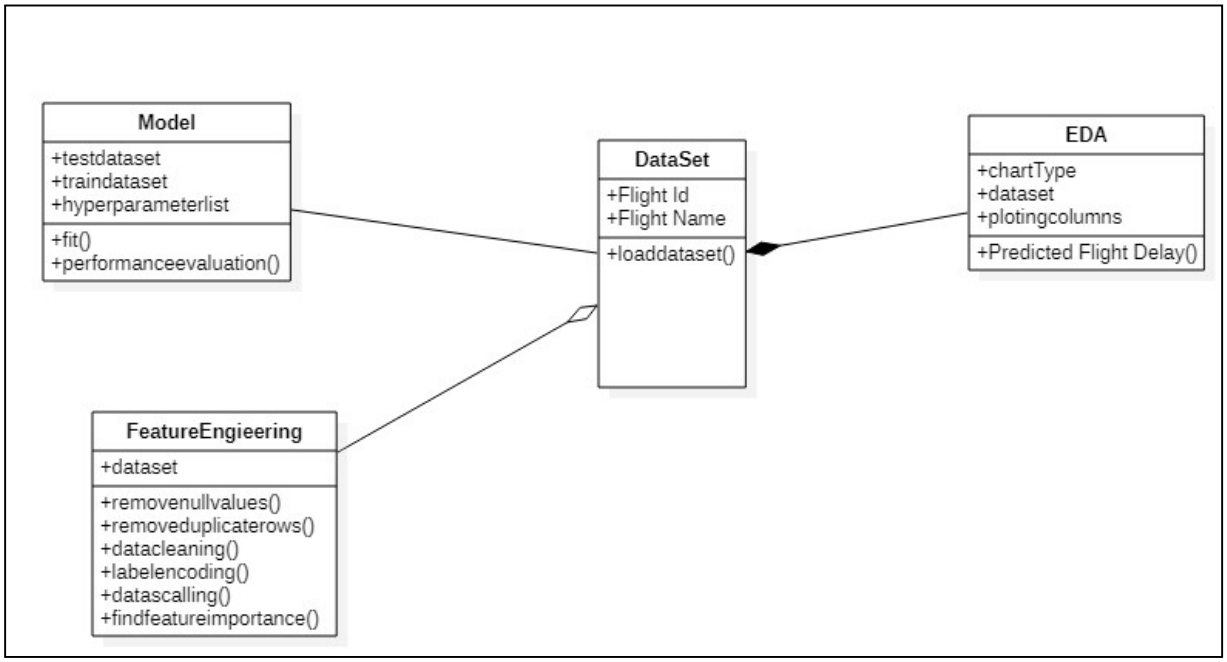
Fig. 4.4 Activity Diagram



4.7 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

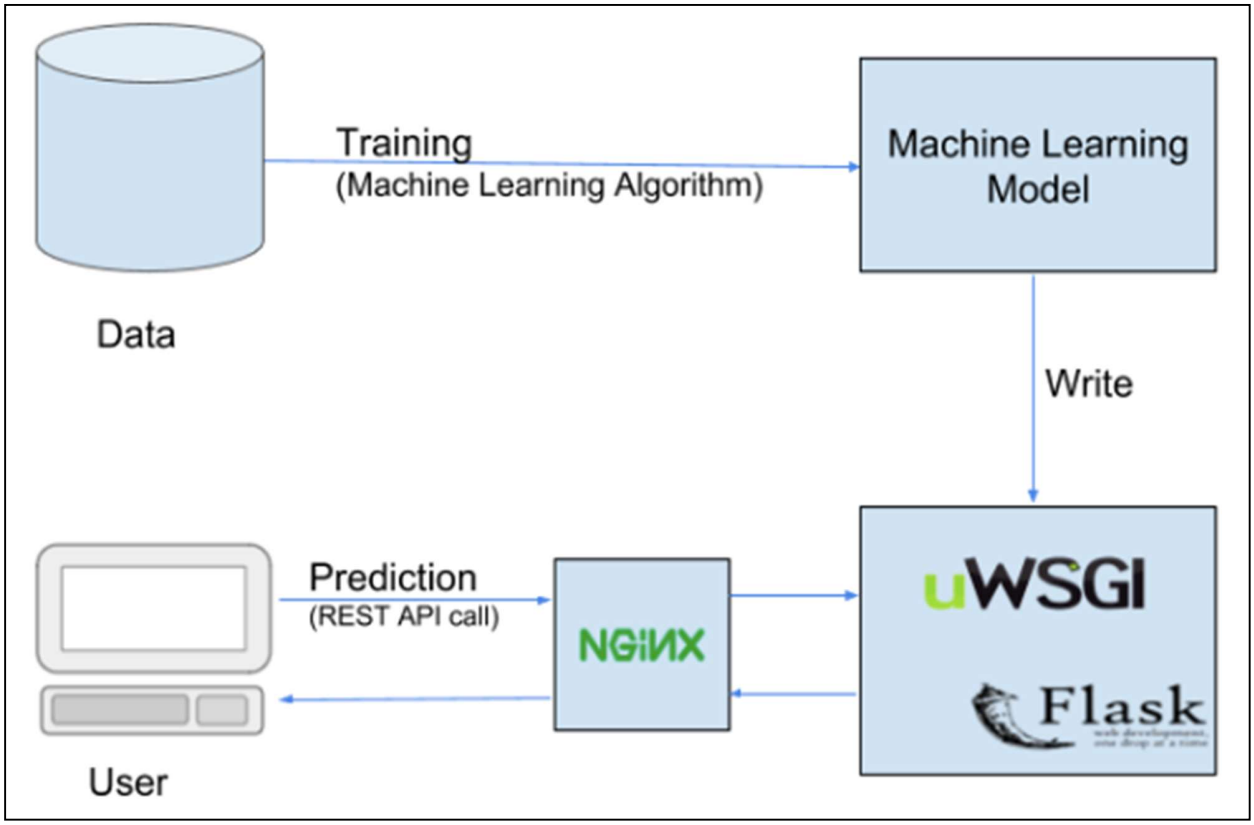
Fig. 4.5 Class Diagram



4.8. Deployment Diagram

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:

Fig. 4.6 Deployment Diagram



CHAPTER – 5

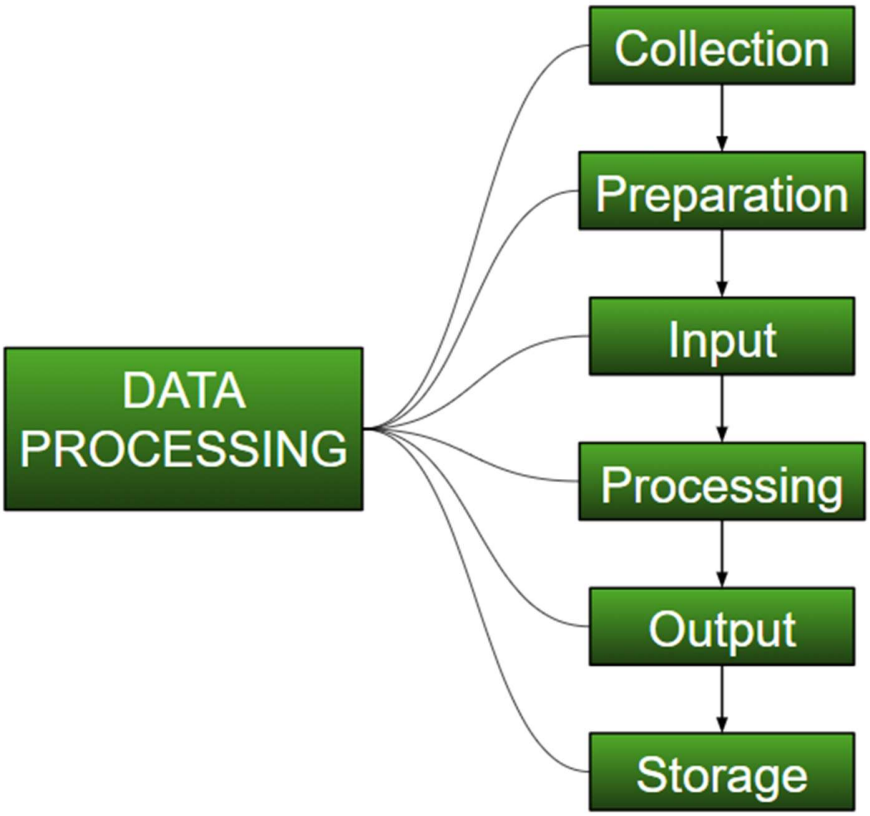
IMPLEMENTATION

We proposed as an alternative to the user-based neighborhood approach. We first consider the dimensions of the input and output of the neural network. In order to maximize the amount of training data we can feed to the network; we consider a training example to be a user profile (i.e., a row from the user-item matrix R) with one rating withheld. The loss of the network on that training example must be computed with respect to the single withheld rating. The consequence of this is that each individual rating in the training set corresponds to a training example, rather than each user. As we are interested in what is essentially a regression, we choose to use root mean squared error (RMSE) with respect to known ratings as our loss function. Compared to the mean absolute error, root mean squared error more heavily penalizes predictions which are further off. We reason that this is good in the context of recommender system because predicting a high rating for an item the user did not enjoy significantly impacts the quality of the recommendations. On the other hand, smaller errors in prediction likely result in recommendations that are still useful—perhaps the regression is not exactly correct, but at least the highest predicted rating are likely to be relevant to the user.

5.1. Data Processing

Data Processing is a task of converting data from a given form to a much more usable and desired form i.e., making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images and many more, depending on the task we are performing and the requirements of the machine. This might seem to be simple but when it comes to really big organizations like Twitter, Facebook, Administrative bodies like Parliament, UNESCO and health sector organizations, this entire process needs to be performed in a very structured manner.

Fig. 5.1 Data Processing



Collection:

The most crucial step when starting with ML is to have data of good quality and accuracy. Data can be collected from any authenticated source like data.gov.in, Kaggle or UCI dataset repository. For example, while preparing for a competitive exam, students' study from the best study material that they can access so that they learn the best to obtain the best results. In the same way, high-quality and accurate data will make the learning process of the model easier and better and at the time of testing, the model would yield state of the art results.

A huge amount of capital, time and resources are consumed in collecting data. Organizations or researchers have to decide what kind of data they need to execute their tasks or research. Example: Working on the Facial Expression Recognizer, needs a large number of images having a variety of human expressions. Good data ensures that the results of the model are valid and can be trusted upon.

Preparation:

The collected data can be in a raw form which can't be directly fed to the machine. So, this is a process of collecting datasets from different sources, analyzing these datasets and then constructing a new dataset for further processing and exploration. This preparation can be performed either manually or from the automatic approach. Data can also be prepared in numeric forms also which would fasten the model's learning.

Example: An image can be converted to a matrix of N X N dimensions; the value of each cell will indicate image pixel.

Input:

Now the prepared data can be in the form that may not be machine-readable, so to convert this data to readable form, some conversion algorithms are needed. For this task to be executed, high computation and accuracy is needed. Example: Data can be collected through the sources like MNIST Digit data(images), twitter comments, audio files, video clips.

Processing:

This is the stage where algorithms and ML techniques are required to perform the instructions provided over a large volume of data with accuracy and optimal computation.

Output:

In this stage, results are procured by the machine in a meaningful manner which can be inferred easily by the user. Output can be in the form of reports, graphs, videos, etc.

Storage:

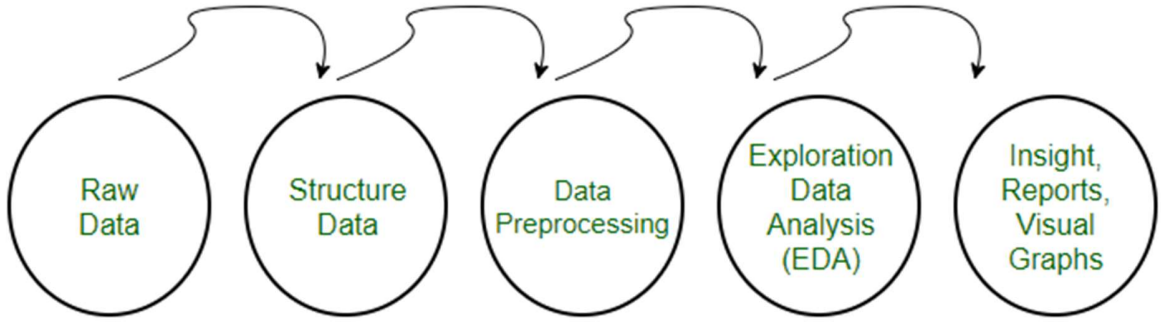
This is the final step in which the obtained output and the data model data and all the useful information are saved for the future use.

5.2. Data Preprocessing for Machine learning in Python

• Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.

• Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Fig. 5.2 Data Preprocessing



Need of Data Preprocessing

• For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

• Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

Rescale Data

- When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.
- This is useful for optimization algorithms in used in the core of machine learning algorithms like gradient descent.
- It is also useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like K-Nearest Neighbors.
- We can rescale your data using scikit-learn using the MinMaxScaler class.

Binarize Data (Make Binary)

- We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.
- This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.
- We can create new binary attributes in Python using scikit-learn with the Binarizer class.

Standardize Data

- Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
- We can standardize data using scikit-learn with the StandardScaler class.

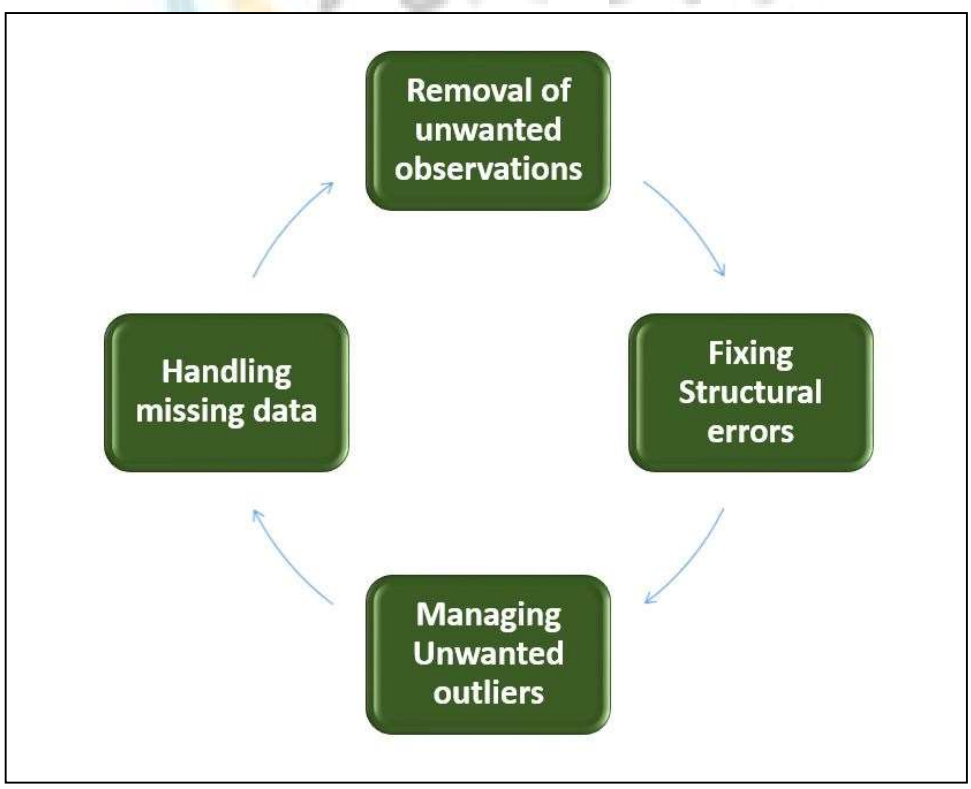
5.3 Data Cleansing

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. Data Cleaning is one of those things that everyone does but no one really talks about. It surely isn't the fanciest part of machine learning and at the same time, there aren't any hidden tricks or secrets to uncover. However, proper data cleaning can make or break your project. Professional data scientists usually spend a very large portion of their time on this step.

Because of the belief that, "Better data beats fancier algorithms". If we have a well-cleaned dataset, we can get desired results even with a very simple algorithm, which can prove very beneficial at times.

Obviously, different types of data will require different types of cleaning. However, this systematic approach can always serve as a good starting point.

Fig. 5.3 Steps involved in Data Cleaning



1. Removal of unwanted observations

This includes deleting duplicate/ redundant or irrelevant values from your dataset. Duplicate observations most frequently arise during data collection and irrelevant observations are those that don't actually fit the specific problem that you're trying to solve.

- Redundant observations alter the efficiency by a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results.
- Irrelevant observations are any type of data that is of no use to us and can be removed directly.

2. Fixing Structural errors

The errors that arise during measurement transfer of data or other similar situations are called structural errors. Structural errors include typos in the name of features, same attribute with different name, mislabeled classes, i.e., separate classes that should really be the same or inconsistent capitalization.

- For example, the model will treat America and America as different classes or values, though they represent the same value or red, yellow and red-yellow as different classes or attributes, though one class can be included in other two classes. So, these are some structural errors that make our model inefficient and gives poor quality results.

3. Managing Unwanted outliers

Outliers can cause problems with certain types of models. For example, linear regression models are less robust to outliers than decision tree models. Generally, we should not remove outliers until we have a legitimate reason to remove them. Sometimes, removing them improves performance, sometimes not. So, one must have a good reason to remove the outlier, such as suspicious measurements that are unlikely to be the part of real data.

4. Handling missing data

Missing data is a deceptively tricky issue in machine learning. We cannot just ignore or remove the missing observation. They must be handled carefully as they can be an indication of something important. The two most common ways to deal with missing data are:

1. Dropping observations with missing values.

Dropping missing values is sub-optimal because when you drop observations, you drop information.

- The fact that the value was missing may be informative in itself.
- Plus, in the real world, you often need to make predictions on new data even if some of the features are missing!

2. Imputing the missing values from past observations.

Imputing missing values is sub-optimal because the value was originally missing but you filled it in, which always leads to a loss in information, no matter how sophisticated your imputation method is.

- Again, "missingness" is almost always informative in itself, and you should tell your algorithm if a value was missing.
- Even if you build a model to impute your values, you're not adding any real information. You're just reinforcing the patterns already provided by other features.
- Both of these approaches are sub-optimal because dropping an observation means dropping information, thereby reducing data and imputing values also is sub-optimal as we fil the values that were not present in the actual dataset, which leads to a loss of information.
- Missing data is like missing a puzzle piece. If you drop it, that's like pretending the puzzle slot isn't there. If you impute it, that's like trying to squeeze in a piece from somewhere else in the puzzle.

- So, missing data is always informative and indication of something important. And we must aware our algorithm of missing data by flagging it. By using this technique of flagging and filling, you are essentially allowing the algorithm to estimate the optimal constant for missingness, instead of just filling it in with the mean.

Some data cleansing tools

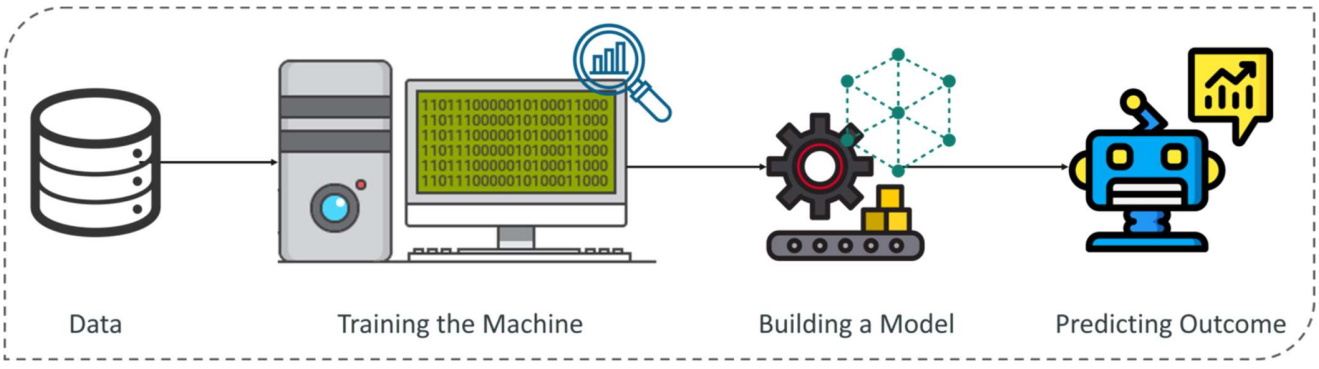
- Openrefine
- Trifacta Wrangler
- TIBCO Clarity
- Cloudingo
- IBM Infosphere Quality Stage

Conclusion

So, we have discussed four different steps in data cleaning to make the data more reliable and to produce good results. After properly completing the Data Cleaning steps, we'll have a robust dataset that avoids many of the most common pitfalls. This step should not be rushed as it proves very beneficial in the further process. Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing.

5.4 Machine Learning Process

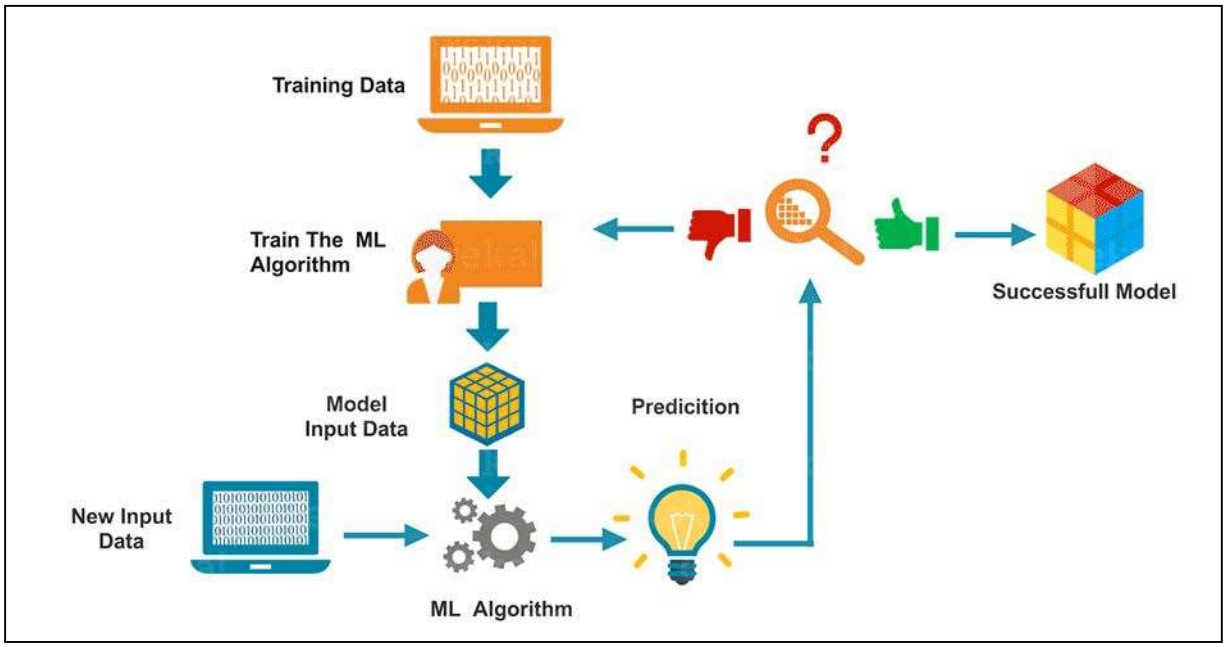
Fig. 5.4 Machine Learning Process



How does Machine Learning Work?

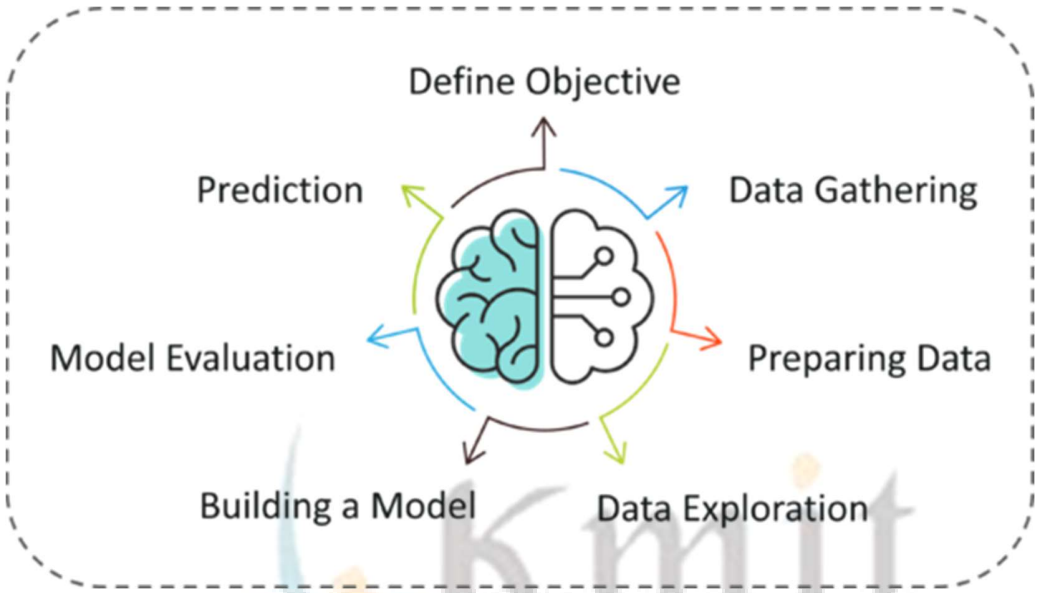
Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model.
The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set.

Fig. 5.5 Working of Machine Learning



The Machine Learning process involves building a Predictive model that can be used to find a solution for a Problem Statement. To understand the Machine Learning process let's assume that you have been given a problem that needs to be solved by using Machine Learning.

Fig. 5.6 Process in Machine Learning



The below steps are followed in a Machine Learning process:

Step 1: Define the objective of the Problem Statement

At this step, we must understand what exactly needs to be predicted. In our case, the objective is to predict the possibility of rain by studying weather conditions. At this stage, it is also essential to take mental notes on what kind of data can be used to solve this problem or the type of approach you must follow to get to the solution.

Step 2: Data Gathering

At this stage, you must be asking questions such as,

- What kind of data is needed to solve this problem?
- Is the data available?

- How can I get the data?

Once you know the types of data that is required, you must understand how you can derive this data. Data collection can be done manually or by web scraping. However, if you're a beginner and you're just looking to learn Machine Learning you don't have to worry about getting the data. There are 1000s of data resources on the web, you can just download the data set and get going.

Coming back to the problem at hand, the data needed for weather forecasting includes measures such as humidity level, temperature, pressure, locality, whether or not you live in a hill station, etc. Such data must be collected and stored for analysis.

Step 3: Data Preparation

The data you collected is almost never in the right format. You will encounter a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc. Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions. Therefore, at this stage, you scan the data set for any inconsistencies and you fix them then and there.

Step 4: Exploratory Data Analysis

Grab your detective glasses because this stage is all about diving deep into data and finding all the hidden data mysteries. EDA or Exploratory Data Analysis is the brainstorming stage of Machine Learning. Data Exploration involves understanding the patterns and trends in the data. At this stage, all the useful insights are drawn and correlations between the variables are understood.

For example, in the case of predicting rainfall, we know that there is a strong possibility of rain if the temperature has fallen low. Such correlations must be understood and mapped at this stage.

Step 5: Building a Machine Learning Model

All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model. This stage always begins by splitting the data set into two parts, training data, and testing data. The training data will be used to build and analyze the model. The logic of the model is based on the Machine Learning Algorithm that is being implemented.

Choosing the right algorithm depends on the type of problem you're trying to solve, the data set and the level of complexity of the problem. In the upcoming sections, we will discuss the different types of problems that can be solved by using Machine Learning.

Step 6: Model Evaluation & Optimization

After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome. Once the accuracy is calculated, any further improvements in the model can be implemented at this stage. Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

Step 7: Predictions

Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (e.g., True or False) or it can be a Continuous Quantity (eg. the predicted value of a stock).

In our case, for predicting the occurrence of rainfall, the output will be a categorical variable.

CHAPTER – 6

TESTING

6.1 Introduction to Testing

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.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

The actual 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 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.

6.2 Types of Testing

There are many types of testing methods are available in that mainly used testing methods are as follows

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.

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.

Functional testing

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

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.
Systems/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

System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

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

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, 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.

6.3 Test Cases

Table 6.1 Test cases:

Tested	**Test name**	**Inputs**	**Expected output**	**Actual Output**	**status**
1	Load Dataset	Csv file	Read dataset	Load dataset	success
2	Split dataset	Train80% and test20%	Divide the training set and Testing set	Split train and Test	success
	Train Model	Train dataset, random value, predicted class	Train with best accuracy	Train with best accuracy	success
4	Validate Model	No. of Epochs	Validate the Model with best fit	Model Generated	success
5	Predict accuracy and Error Rate	Accuracy	Plot expected accuracy and predicted accuracy	Plot expected predicted accuracy	success
6	Test Data	Test column	Predicted accuracy	Predicted accuracy	success

CHAPTER – 7

SCREEN SHOTS

Fig. 7.1 Mapping of Datasets:

12	31	4	B6	745	N828JB	JFK	PSE	2359	...
12	31	4	B6	1503	N913JB	JFK	SJU	2359	...
12	31	4	B6	333	N527JB	MCO	SJU	2359	...
12	31	4	B6	839	N534JB	JFK	BQN	2359	...

: 31 columns

```
airlinecompanies = airlines.set_index('IATA_CODE')['AIRLINE'].to_dict()
```

```
airlinecompanies
```

```
{'UA': 'United Air Lines Inc.',
 'AA': 'American Airlines Inc.',
 'US': 'US Airways Inc.',
 'F9': 'Frontier Airlines Inc.',
 'B6': 'JetBlue Airways',
 'OO': 'Skywest Airlines Inc.',
 'AS': 'Alaska Airlines Inc.',
 'NK': 'Spirit Air Lines',
 'WN': 'Southwest Airlines Co.',
 'DL': 'Delta Air Lines Inc.',
 'EV': 'Atlantic Southeast Airlines',
 'HA': 'Hawaiian Airlines Inc.',
 'MQ': 'American Eagle Airlines Inc.',
 'VX': 'Virgin America'}
```

Fig. 7.2 Classifying the delay:

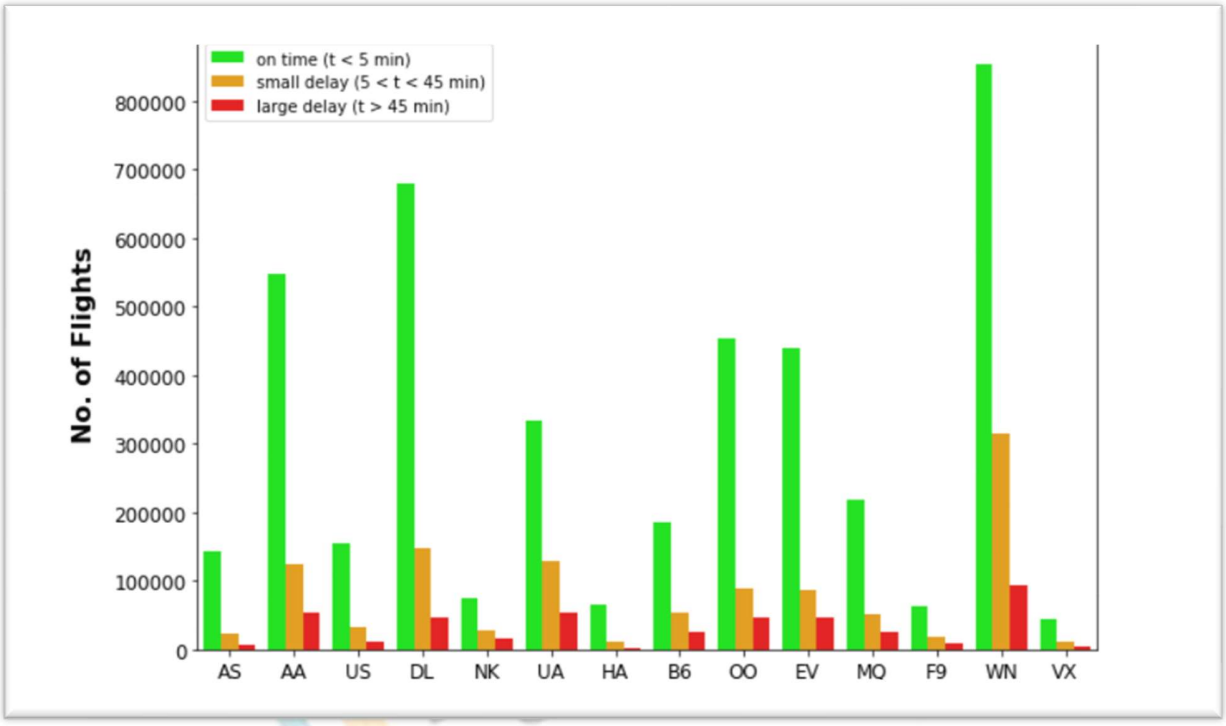


Fig. 7.3 Calculating the mean delay:

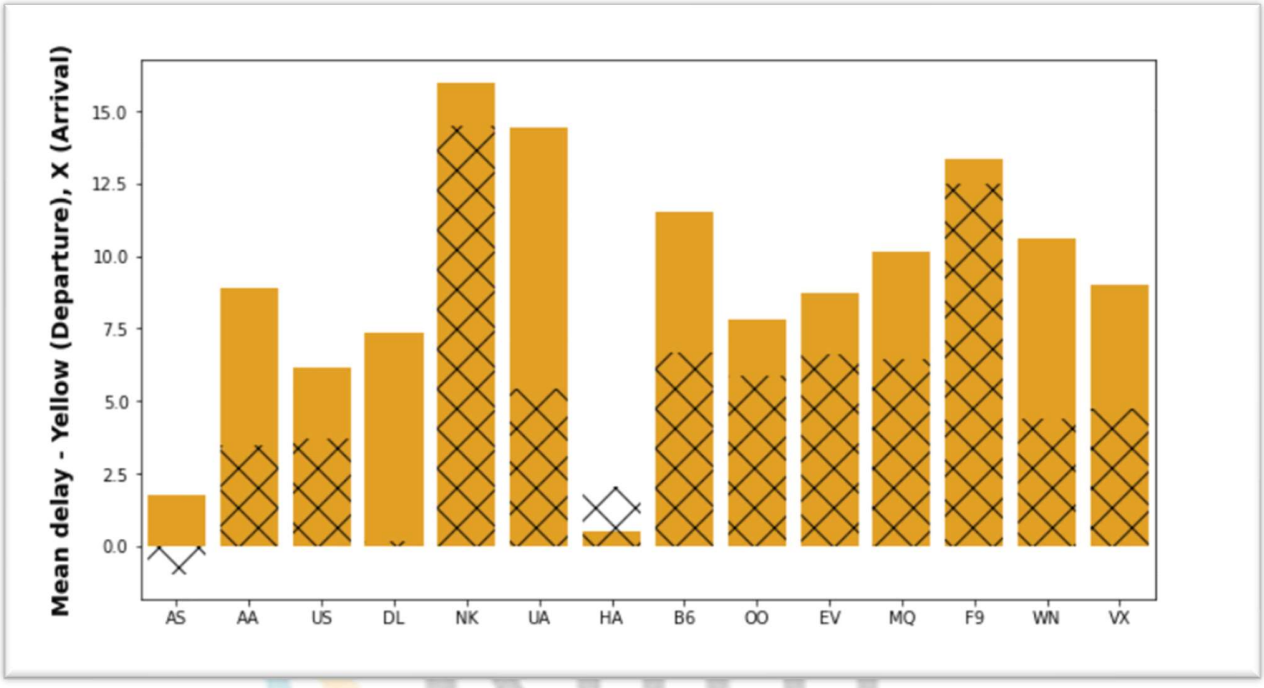


Fig. 7.4 Dropping the null values:

```
airport.isnull().sum()
```

```
IATA_CODE    0
AIRPORT      0
CITY         0
STATE        0
COUNTRY      0
LATITUDE     3
LONGITUDE    3
dtype: int64
```

```
airport = airport.dropna(subset = ['LATITUDE','LONGITUDE'])
```

Fig. 7.5 Finding percentage of null values:

```
YEAR                  0.000000
MONTH                 0.000000
DAY                   0.000000
DAY_OF_WEEK           0.000000
AIRLINE               0.000000
FLIGHT_NUMBER         0.000000
TAIL_NUMBER           0.252978
ORIGIN_AIRPORT        0.000000
DESTINATION_AIRPORT   0.000000
SCHEDULED_DEPARTURE   0.000000
DEPARTURE_TIME        1.480526
DEPARTURE_DELAY       1.480526
TAXI_OUT              1.530259
WHEELS_OFF            1.530259
SCHEDULED_TIME        0.000103
ELAPSED_TIME          1.805629
AIR_TIME              1.805629
DISTANCE              0.000000
WHEELS_ON             1.589822
TAXI_IN               1.589822
SCHEDULED_ARRIVAL     0.000000
ARRIVAL_TIME          1.589822
ARRIVAL_DELAY         1.805629
DIVERTED              0.000000
CANCELLED             0.000000
CANCELLATION_REASON  98.455357
AIR_SYSTEM_DELAY     81.724960
SECURITY_DELAY       81.724960
```

Fig. 7.6 Final shape after removing null values:

```
Int64Index: 5714008 entries, 0 to 5819078
Data columns (total 26 columns):
 #   Column               Dtype
---  ------               -----
 0   YEAR                 int64
 1   MONTH                int64
 2   DAY                  int64
 3   DAY_OF_WEEK          int64
 4   AIRLINE              object
 5   FLIGHT_NUMBER        int64
 6   TAIL_NUMBER          object
 7   ORIGIN_AIRPORT       object
 8   DESTINATION_AIRPORT  object
 9   SCHEDULED_DEPARTURE  int64
 10  DEPARTURE_TIME       float64
 11  DEPARTURE_DELAY      float64
 12  TAXI_OUT             float64
 13  WHEELS_OFF           float64
 14  SCHEDULED_TIME       float64
 15  ELAPSED_TIME         float64
 16  AIR_TIME             float64
 17  DISTANCE             int64
 18  WHEELS_ON            float64
 19  TAXI_IN              float64
 20  SCHEDULED_ARRIVAL    int64
 21  ARRIVAL_TIME         float64
 22  ARRIVAL_DELAY        float64
 23  DIVERTED             int64
 24  CANCELLED            int64
 25  DELAY_LEVEL          int64
dtypes: float64(11), int64(11), object(4)
memory usage: 1.1+ GB
```

Fig. 7.7 Final Data set:

	AIRLINE	Org_Airport_Name	Origin_city	Dest_Airport_Name	Destination_city	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DISTANCE	Actual_Depart
0	Alaska Airlines Inc.	Ted Stevens Anchorage International Airport	Anchorage	Seattle-Tacoma International Airport	Seattle	ANC	SEA	1448	23:54
1	Alaska Airlines Inc.	Ted Stevens Anchorage International Airport	Anchorage	Seattle-Tacoma International Airport	Seattle	ANC	SEA	1448	00:41
2	Alaska Airlines Inc.	Ted Stevens Anchorage International Airport	Anchorage	Seattle-Tacoma International Airport	Seattle	ANC	SEA	1448	01:40
3	Alaska Airlines Inc.	Ted Stevens Anchorage International Airport	Anchorage	Seattle-Tacoma International Airport	Seattle	ANC	SEA	1448	02:09
4	Alaska Airlines Inc.	Ted Stevens Anchorage International Airport	Anchorage	Seattle-Tacoma International Airport	Seattle	ANC	SEA	1448	04:57
...	...	...	...	...	...	...	...	...	
5221995	Atlantic Southeast Airlines	Meridian Regional Airport	Meridian	Hattiesburg-Laurel Regional Airport	Hattiesburg-Laurel	MEI	PIB	69	20:37
5221996	Atlantic Southeast Airlines	Meridian Regional Airport	Meridian	Hattiesburg-Laurel Regional Airport	Hattiesburg-Laurel	MEI	PIB	69	16:16
5221997	Atlantic Southeast Airlines	Meridian Regional Airport	Meridian	Hattiesburg-Laurel Regional Airport	Hattiesburg-Laurel	MEI	PIB	69	20:56
5221998	Atlantic Southeast Airlines	Meridian Regional Airport	Meridian	Hattiesburg-Laurel Regional Airport	Hattiesburg-Laurel	MEI	PIB	69	14:21
5221999	Atlantic Southeast Airlines	Meridian Regional Airport	Meridian	Hattiesburg-Laurel Regional Airport	Hattiesburg-Laurel	MEI	PIB	69	20:20

219244 rows × 22 columns

Fig. 7.8 Plotting the data-city vs delay:

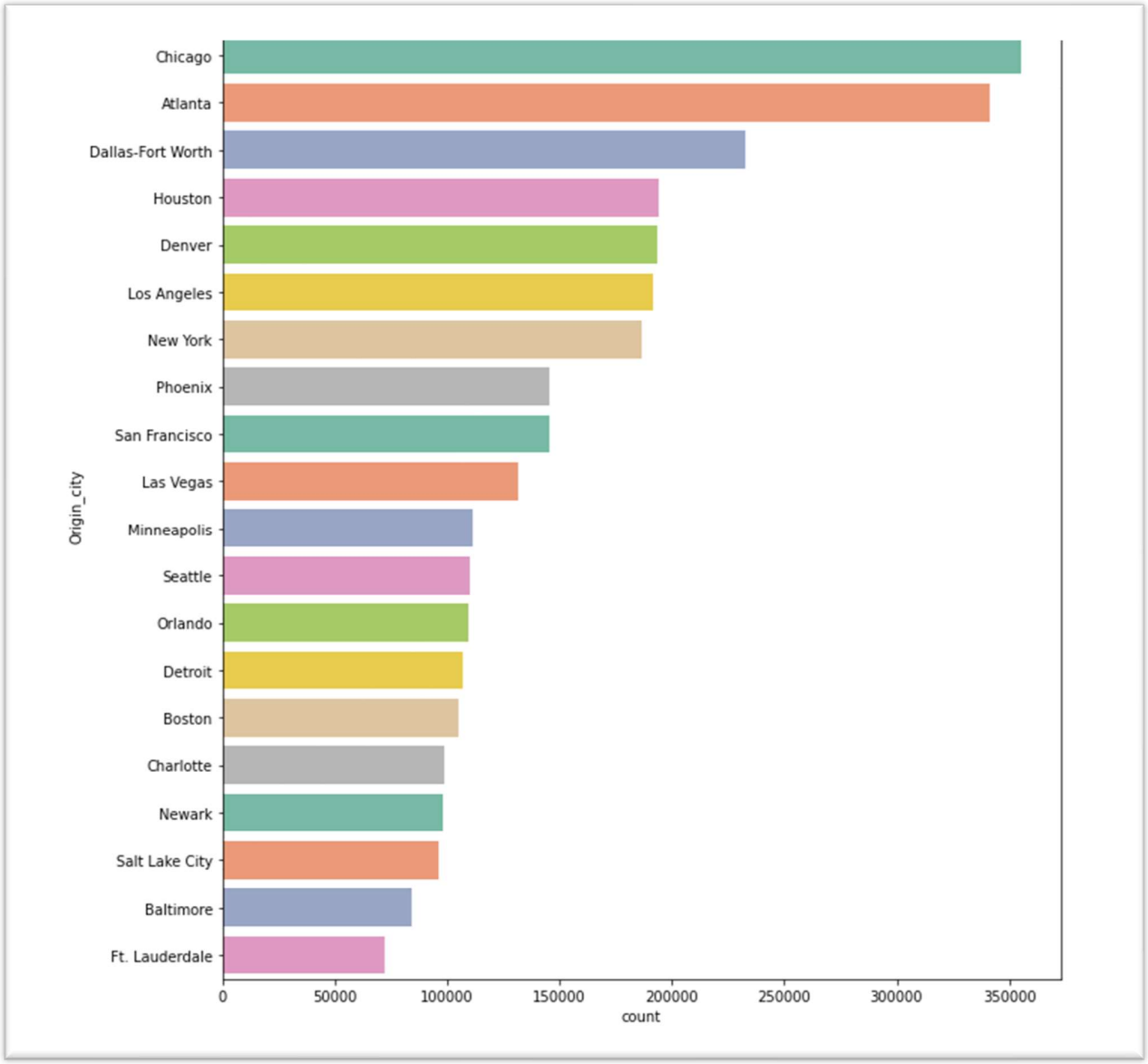


Fig. 7.9 Plotting the data-Airline vs Arrival Delay:

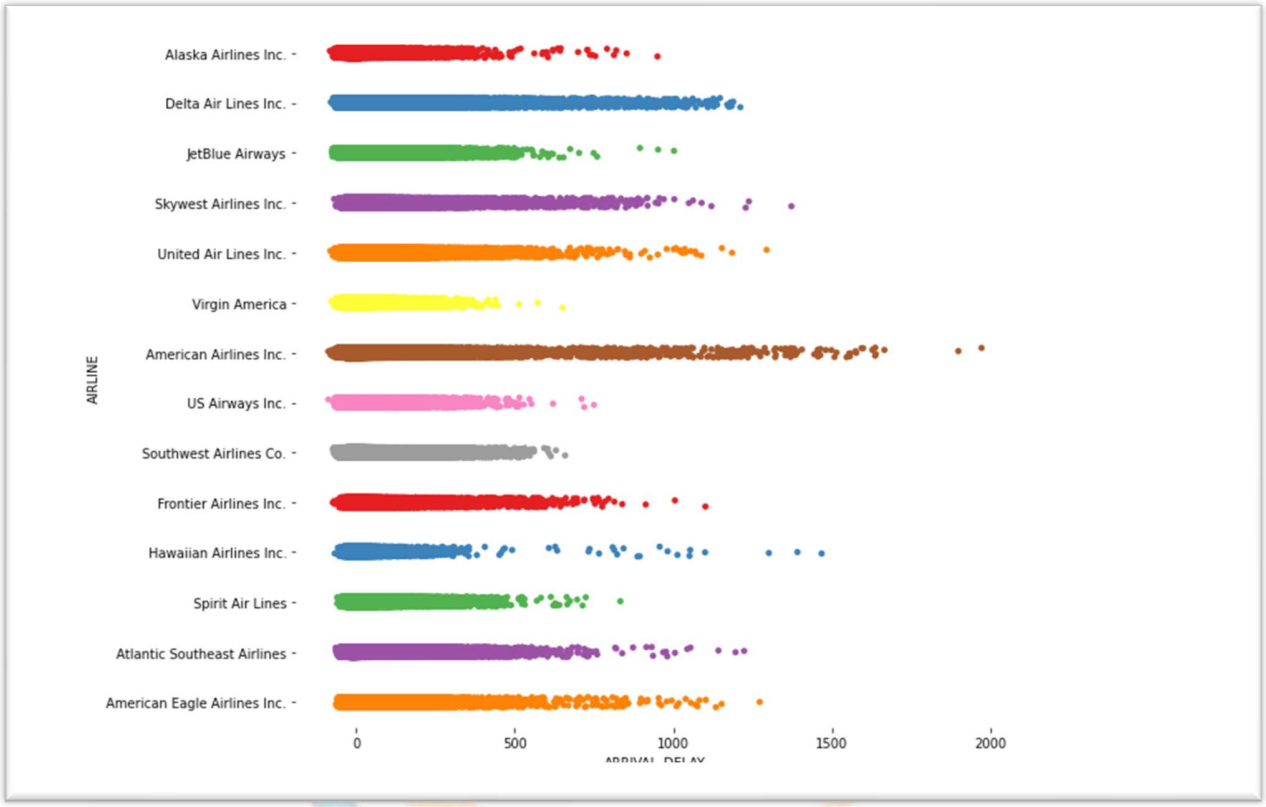


Fig. 7.10 Correlation Plot:

CHAPTER – 8

FUTURE ENHANCEMENTS

The future scope of this paper can include the application of more advanced, modern and innovative preprocessing 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.

CHAPTER – 9

CONCLUSION

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.

CHAPTER – 10

REFERENCES

[1] N. G. Rupp, "Further Investigation into the Causes of Flight Delays," in Department of Economics, East Carolina University, 2007.

[2] "Bureau of Transportation Statistics (BTS) Databases and Statistics," [Online]. Available: http://www.transtats.bts.gov.

[3] "Airports Council International, World Airport Traffic Report," 2015,2016.

[4] E. Cinar, F. Aybek, A. Caycar, C. Cetek, "Capacity and delay analysis for airport maneuvering areas using simulation," Aircraft Engineering and Aerospace Technology, vol. 86, no. No. 1,pp. 43-55, 2013.

[5] Navoneel, et al., Chakrabarty, "Flight Arrival Delay Prediction Using Gradient Boosting Classifier," in Emerging Technologies in Data Mining and Information Security, Singapore, 2019.

[6] Y. J. Kim, S. Briceno, D. Mavris, Sun Choi, "Prediction of weatherinduced airline delays based on machine learning algorithms," in 35th Digital Avionics Systems Conference (DASC), 2016.

[7] W.-d. Cao. a. X.-y. Lin, "Flight turnaround time analysis and delay prediction based on Bayesian Network," Computer Engineering and Design, vol. 5, pp. 1770-1772, 2011

. [8] J. J. Robollo, Hamsa, Balakrishnan, "Characterization and Prediction of Air Traffic Delays"

. [9] S. Sharma, H. Sangoi, R. Raut, V. C. Kotak, S. Oza, "Flight Delay Prediction System Using Weighted Multiple Linear Regression," International Journal of Engineering and Computer Science, vol. 4, no. 4, pp. 11668 - 11677, April 2015. [10] A. M. Kalliguddi, Area K., Leboulluec, "Predictive Modelling of Aircraft Flight Delay," Universal Journal of Management, pp. 485 - 491, 2017.

[11] Noriko, Etani, "Development of a predictive model for on-time arrival fight of airliner by discovering correlation between fight and weather data," 2019.

[12] [Online]. Available: https://towardsdatascience.com/metrics-toevaluate-your-machine-learning-algorithm-f10ba6e38234.

[13] C. J. Willmott, Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square(RMSE) in assessing average model performance," Climate Research, vol. 30, no. 1, pp. 79 - 82, 2005.

[14] [Online]. Available: http://scikitlearn.org/stable/modules/classes.html?source=post_page/sklearn-metrics-metrics.