OPTIMIZING FLIGHTS DELAYS PREDICTIONS: ERROR EVALUATION IN MACHINE LEARNING MODELS

1. Donga Meghana, 2. Jala Sai Chandana, 3.Bhupathi Swathi

B -Tech, 4th year students Department of Data Science (CSE) Sphoorthy Engineering College.

meghanadonga001@gmail.com, jschandana104@gmail.com, bhupathiswathi571@gmail.com.

Under the supervision of Mrs.K. Sneha (Asst. Professor).

kunasneha92@gmail.com

ABSTRACT

Flight delays pose significant challenges in the aviation sector due to air traffic congestion and its adverse effects on both economic and environmental aspects. This paper addresses the issue by employing machine learning classifiers to predict the likelihood of flight delays. The models utilized include Logistic Regression, Decision Tree Regression, Bayesian Ridge, Random Forest Regression, and Gradient Boosting Regression. The aviation industry, experiencing substantial growth over the past two decades, has witnessed increased air traffic congestion, leading to frequent flight delays. These delays not only result in financial losses for airlines but also have environmental repercussions. In response, airlines strive to prevent or minimize delays through various measures. The focus of this study is to predict the occurrence of flight delays using machine learning techniques. The selected classifiers—Logistic Regression, Decision Tree Regression, Bayesian Ridge, Random Forest Regression, and Gradient Boosting Regression—are applied to analyze relevant data and forecast whether a specific flight will experience a delay or not. The paper aims to contribute valuable insights into developing effective strategies for mitigating flight delays. By leveraging machine learning algorithms, the aviation industry can enhance its ability to predict and address delays, thereby improving overall operational efficiency and minimizing the negative impact on both the industry and the environment. The analysis includes error calculations to assess the accuracy of the models, providing a comprehensive evaluation of their predictive capabilities.

INTRODUCTION:

Statistical modelling is a mathematical way of making approximations from input data. These approximations are then used to make predictions. Statistical models help in predicting the future probabilistic behavior of a system based on past statistical data. Predictive modelling has been used in many fields, for example in crime cases to detect the likeliness of an email being spam and flight delays. In evaluation of how different models perform in modelling of flight delays, regression models have been found efficient in predicting flight delays since they highlighted the various causes of flight delays. However, they could not categorize complex data. Econometric models have been used to model scheduled flight cancellation and to show how delays from one airport were propagated to other destinations. These models did not provide a complete vindication since they ignored variables that were difficult to quantify. When subjected to social-economic situations, the models showed discriminative and subjective results. Among the models used, random forest has been found to have superior performance. Prediction accuracy may vary due to factors such as time of forecast and airline dynamics. A developed multiple regression model has shown that distance, day and scheduled departure are key factors in predicting flight delay. However, though the model gives flagged out the significant factors, its prediction accuracy was poor. Moreover, the model is limited to only one flight route. Comparison of other models, such as the K-means clustering Algorithms and Fourier fit model, have shown that Fourier fit model could predict flight delays with a high precision. However, the two models were found to be suitable a single airport, but not prediction applied to multiple airports. Probability models such as the normal distribution and the Poisson distribution have been used to Model flight departure and arrival delays. However, the prediction accuracy varied depending on variables such as time duration and the number of airports considered. Normal distribution was observed to model flight departure delays better while arrival delays were modeled better by the Poisson distribution. However, these models are parametric and assume that the response takes a particular functional form. If this form is not met by the training data set, the resulting model will not fit the data well and the estimates from this model will be poor. Logistic regression model has been used to model flight on-time performance. The model showed good performance with the training data set and the testing data set. The variance of the model was also low. However, its parametric nature can be a weakness if the training data set will not meet the assumed functional form. Neural networks performed better than logistic regression model in prediction of death in patients with suspected sepsis in an emergency room. This was attributed to the neural networks having few features to be verified before model construction and its ability to fit non-linear relationship between dependent and independent variables. Support Vector Machine (SVM) model was fitted and it was observed to fit all the training data set correctly. In prediction of auto-ignition temperatures of organic compounds, SVM performed better than multiple linear regression and back propagation neural network. Random forests have been used to model delay innovation. Results from this study showed that more decision trees were better but up to a certain critical value. Prediction of new vehicle prediction approach in computational toxicology led to results with random forest performing better than decision tree. Random forests and SVM are classified under machine learning. Under machine learning, the training data is divided into several samples. At each sample, a model is fitted and tested against the testing data set. The sample that yields. The best model is obtained from a plot of the train errors and the test errors against the sample size. Their overall advantage of the SVM and the random forest is their non-parametric nature in that they do not assume a particular functional form of the response under investigation. This makes them very flexible since they fit a wider range of shapes of the response. Modelling studies on flight delays are not available for Kenya aviation industry. The aim of this study is to compare the prediction power of models that have been used to predict flight delays at Jomo Kenyatta International Airport. Secondary data that was obtained from Kenya Airports Authority on flights at Jomo Kenyatta International Airport. The data was for the year 2017/2018 where the year started on March 2017 and ended on March 2018. The variables used included; the day of the flight (that is, Monday to Sunday), the month (that is, January to December), the airline, the flight class (that is, domestic or international), season (that is, summer (March to October) or winter (October to

March), capacity of the aircraft, flight ID (tail number) and whether the flight had flown at night or during the day. The data was analyzed using R-Score statistical software. The time difference between the scheduled time and the actual time for flights was calculated. A time difference of more than 15 minutes was classified as a delay and it was given a value 1 and a time difference of less than 15 minutes was classified a non-delay and given the value 0. The three models, logistic regression model, SVM model and Random Forest, were fitted by machine learning. The entire data set was divided into a training data set of 15000 flights and a testing data set of 5000 flights. In fitting the models, different random samples were created from the training data by the programmed laptop used. For each sample, a model was fitted and tested using the testing data.

LITERATURE SURVEY:

Flight delay prediction has recently gained growing popularity due to the significant role plays in efficient airline and airport operation. Most of the previous prediction works consider the single-airport scenario, which overlooks the time-varying spatial interactions hidden in airport networks. In this paper, the flight delay prediction problem is investigated from a network perspective (i.e., multi -airport scenario). To model the time-evolving and periodic graph-structured information in the airport network, a flight delay prediction approach based on the graph convolutional neural network (GCN) is developed in this paper. More specifically, regarding that GCN cannot take both delay time series and time-evolving graph structures as inputs, a temporal convolutional block based on the Markov property is employed to mine the time-varying patterns of flight delays through a sequence of graph snapshots. Moreover, considering that unknown occasional air routes under emergency may result in incomplete graph-structured inputs for GCN, an adaptive graph convolutional block is embedded into the proposed method to expose spatial interactions hidden in airport networks. Through extensive experiments, it has been shown that the proposed approach outperforms benchmark methods with a satisfying accuracy improvement at the cost of acceptable execution time. The obtained results reveal that deep learning approach based on graph-structured inputs have great potentials in the flight delay prediction problem. , Member, IEEE. These delays have led into inevitable consequences such as unpleasant passenger experiences, followed by economic losses of relevant airspace users. The annual cost of flight delays to the global economy was estimated to be $50 billion in 2019 [5]. Such high loss motivates the analysis of air traffic delays and the development of more advanced flight delay prediction approaches in both industry and academia [6]. Index Terms—Flight delay prediction, time-evolving airport network, graph-structured information, graph convolutional neural network.

EXISTING SYSTEM:

Supervised automatic learning algorithms Support Vector Machine and the k-nearest neighbor to predict delays in the arrival of operated flights including the five busiest US airports. The precision achieved was very low with gradient booster as a classifier with a limited data set. Applied machine learning algorithms k-Nearest Neighbors 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.

DISADVANTAGES OF EXISTING SYSTEM:

• Non-parametric nature do not assume a particular functional form of the response under investigation data.

• The predictability may additionally range because of factors such as the number of origin destination pairs and the forecast horizon.

• The forecasts were based on some key attributes.

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 31 columns and 20277 and it can grow able by our implementation. By using pandas library we can fill the missing values which is essential for processing data for model.

ADVANTAGES OF PROPOSED SYSTEM:

• Supervised learning technique to gather the advantages of having the schedule and real arrival time.

• Algorithms are light computation cost will take.

• We develop a system that predicts for a delay in flight departure based on certain parameters.

SYSTEM DESIGN:

• System architecture involves using various diagrams like Data Flow, UML, Use Case, Class, Sequence, and Activity to model and visualize system components, interactions, and processes.

• System architecture involves designing the structure and behavior of complex systems, including hardware, software, networks, and other components, to ensure they work together effectively to meet specific requirements and objectives.

• Error calculation in flight delay prediction means measuring how much the predicted flight delay times differ from the actual delay times using metrics like Mean Absolute Error or Root Mean Square Error.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

System: Intel Core i7.

Hard Disk: 1 TB. Monitor: 15’’ L

ED Input

Devices: Keyboard, Mouse Ram: 16 GB.

SOFTWARE REQUIREMENTS:

Operating system: Windows 10.

Coding Language: Python

Tool: PyCharm, Visual Studio Code Database: MySQL

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth 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.

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL 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.

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.

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.

SYSTEM ARCHITECTURE :

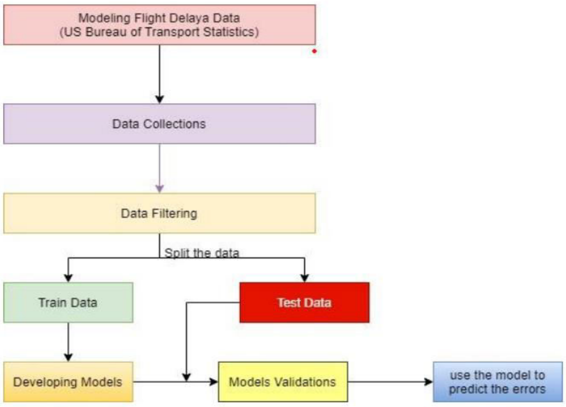


Fig1: System Architecture

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.

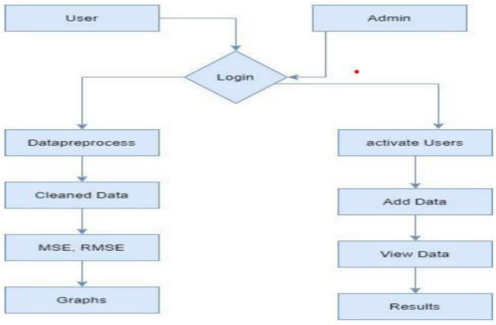


Fig2: Data Flow Diagram

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.

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.

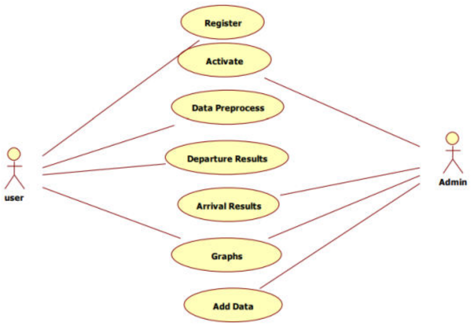


Fig3: Use Case Diagram

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.

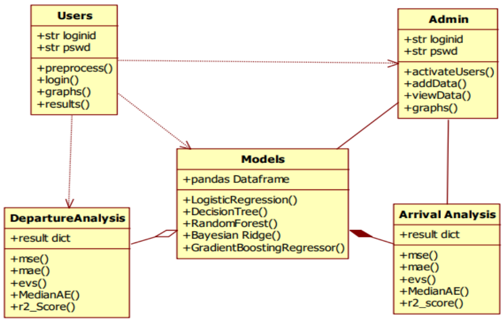


Fig4: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

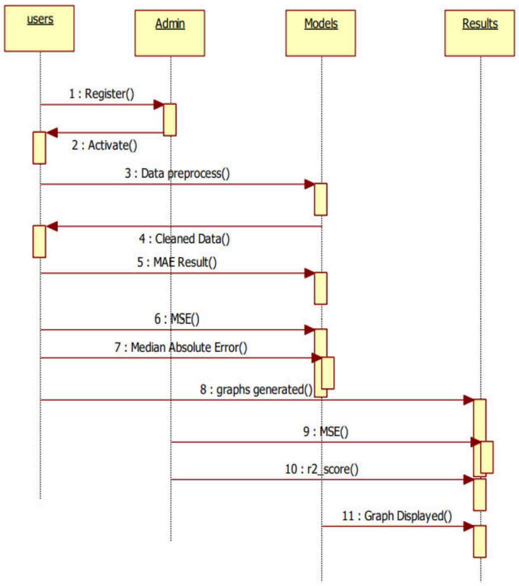


Fig5: Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of work flows 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 work flows of components in a system. An activity diagram shows the overall flow of control.

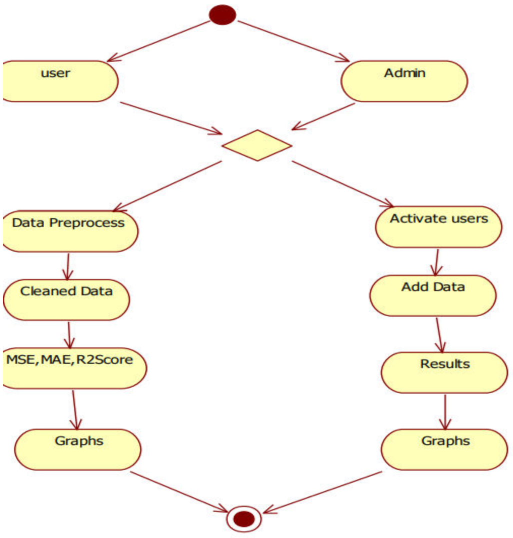


Fig6: Activity Diagram

IMPLEMENTATION:

MODULES:

- User
- Admin
- Data pre-process
- Model Execution

MODULES DESCRIPTION:

User:

The User can register the first. While registering he required a valid user email and mobile for further communications. Once the user register then admin can activate the customer. Once admin activated the User then user can login into our system. The dataset collected from US Bureau of Transport is not directly processed. Before process we need to clean the data. Once clean the data then user can test the departure delay performance based on selected models. The user can see the results in the browser. The all error scores displayed and graphical representation can be displayed.

Admin:

Admin can login with his credentials. Once he login he can activate the users. The activated user only login in our applications. 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 dataset parameters are Day, Departure Delay, Airline, Flight Number, Destination Airport, Origin Airport, Day of Week, and Taxi out. So this data we consider for further process.

Data Pre-process:

The admin provided data has been stored in the SQLite database. To process our methodology we need to perform data cleaning process. By using pandas data frame we can fill the missing values with its mean type. Once data cleaned the data will be displayed on the browser.

Model Execution

Machine learning models such as Logistic Regression, Decision Tree Regression, Bayesian Ridge, Random Forest Regression and Gradient Boosting Regression we predict result. The MSE is appropriate for our regression problems since it is differentiable, contributing to the stability of the algorithms. It also heavily punishes the bigger errors over smaller errors. MAE is a risk providing metric which tells the expected value of the absolute error loss. Explained Variance Score proportion with which our machine learning model explains the scattering of the dataset is measured by this technique. R2 Score Goodness of fit is indicated by this metric and hence it measures the probability of the model to predict unknown samples, through the proportion of explained variance. The best score can be 1.0 and the score can also be negative.

SOFTWARE ENVIRONMENT

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using white space indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

SYSTEM TEST

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.

TYPES OF TESTS

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.

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 per-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.

INPUT AND OUTPUT DESIGN:

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

➢What data should be given as input?

➢How the data should be arranged or coded? ➢The dialog to guide the operating personnel in providing input.

➢ Methods for preparing input validations and steps to follow when error occur.

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.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements. 2. Select methods for presenting information. 3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

• Convey information about past activities, current status or projections of the

• Future.

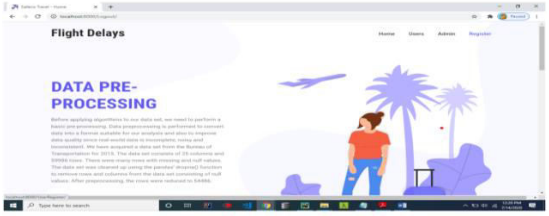
• Signal important events, opportunities, problems, or warnings.

• Trigger an action.

• Confirm an action.

RESULTS:

Home page

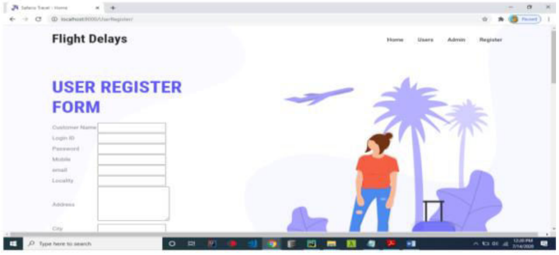


DATA PRE-PROCESSING

Before applying algorithms to our data set, we need to perform a basic pre-processing. Data preprocessing is performed to convert data into a format suitable for our analysis and also to improve data quality since real-world data is incomplete, noisy and inconsistent. We have acquired a data set from the Bureau of Transportation for 2015. The data set consists of 25 columns and 59986 rows. There were many rows with missing and null values. The data set was cleaned up using the pandas' dropna() function to remove rows and columns from the data set consisting of null values. After preprocessing, the rows were reduced to 54486.

Fig7: Home Page

Register Form



USER REGISTER FORM

Customer Name
Login ID
Password
Mobile
email
Locality
Address
City

Fig8: Register Form

User Login Form:

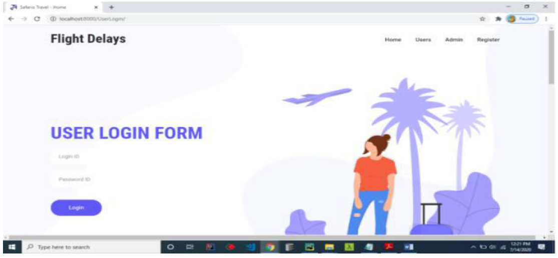
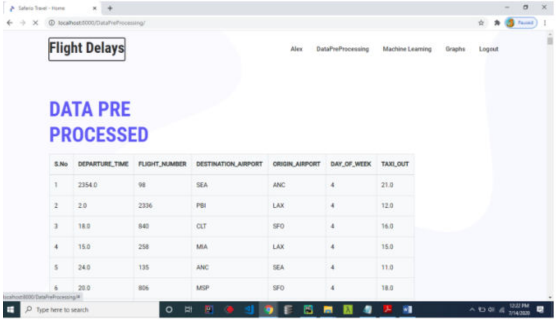


Fig9: User Login Form

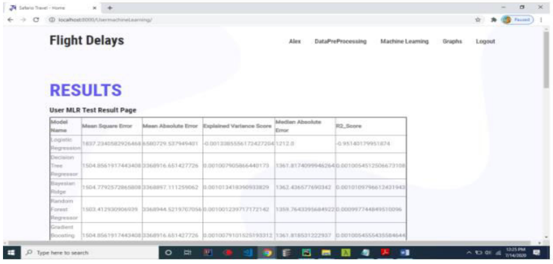
Preprocessed Data:



S.No	DEPARTURE_TIME	FLIGHT_NUMBER	DESTINATION_AIRPORT	ORIGIN_AIRPORT	DAY_OF_WEEK	TAXI_OUT
1	2354.0	98	SEA	ANC	4	21.0
2	2.0	2336	PBI	LAX	4	12.0
3	18.0	840	CLT	SFO	4	16.0
4	15.0	258	MIA	LAX	4	15.0
5	24.0	135	ANC	SEA	4	11.0
6	20.0	806	MSP	SFO	4	18.0

Fig10: Pre-processed Data

Algorithm codes:



User MLR Test Result Page

Model Name	Mean Square Error	Mean Absolute Error	Explained Variance Score	Median Absolute Error	R2_Score
Logistic Regression	1637.2340582926464	6580729.537949401	-0.0013385556172427204	1212.0	-0.95140179951874
Decision Tree Regressor	1504.8561917443408	3368916.651427726	0.0010079058664401 73	1361.8174099946264	0.001005451250667310 8
Bayesian Ridge	1504.7792572865808	3368897.111259062	0.0010134183909 33829	1362.436577690342	0.0010109796612431943
Random Forest Regressor	1503.41293090 6939	3368944.5219767056	0.0010012397171 72142	1359.7643395684922	0.000997744849510096
Gradient Boosting	1504.8561917443408	3368916.651427726	0.0010079101525193312	1361.818531222937	0.0010054555435584644

Fig11: Result

User Side graphs

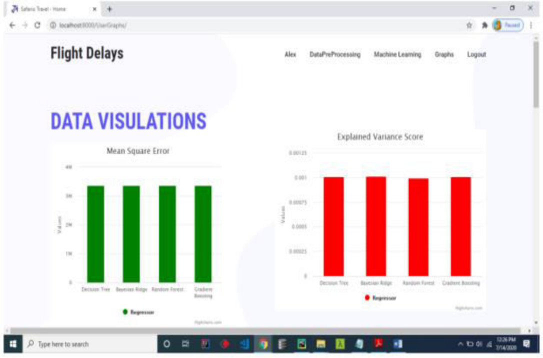
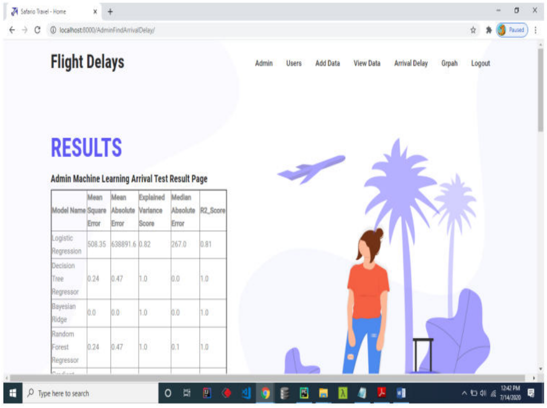


Fig12: User Side Graph

Admin View Results



Admin Machine Learning Arrival Test Result Page

Model Name	Mean Square Error	Mean Absolute Error	Explained Variance Score	Median Absolute Error	R2_Score
Logistic Regression	508.35	638891.6	0.82	267.0	0.81
Decision Tree Regressor	0.24	0.47	1.0	0.0	1.0
Bayesian Ridge	0.0	0.0	1.0	0.0	1.0
Random Forest Regressor	0.24	0.47	1.0	0.1	1.0

Fig13: Admin View Results

Arrival Graph

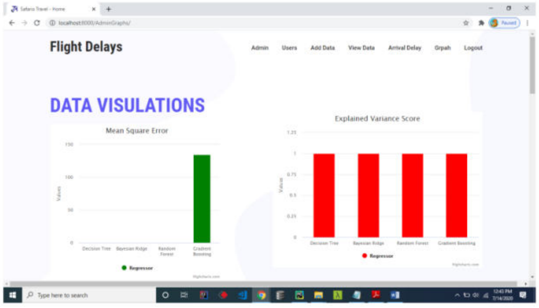


Fig14: Arrival Graph

Arrival Graph:

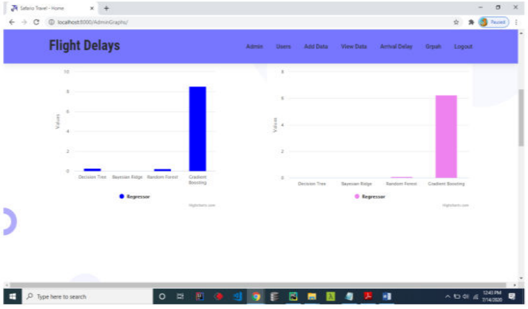
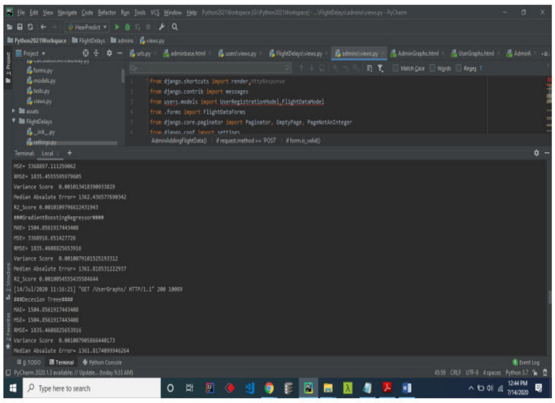


Fig15: Arrival Graph

Server Side results


File Edit View Navigate Code Refactor Run Tools VCS Window Help Python2021Workspace [G:\Python2021Workspace] - ...\FlightDelays\admins\views.py - PyCharm
HearPredict
Python2021Workspace FlightDelays admins views.py
Project
urls.py adminbase.html users\views.py FlightDelays\views.py admins\views.py AdminGraphs.html UserGraphs.html AdminA
Match Case Words Regex
forms.py
models.py
tests.py
views.py
assets
FlightDelays
__init__.py
settings.py
AdminAddingFlightData() if request.method == 'POST' if form.is_valid()
Terminal: Local
6: TODO Terminal Python Console
PyCharm 2020.1.3 available // Update... (today 9:33 AM)
Type here to search

```
from django.shortcuts import render,HttpResponse
from django.contrib import messages
from users.models import UserRegistrationModel,FlightDataModel
from .forms import FlightDataForms
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
from django.conf import settings
```

```
MSE= 3368897.111259062
RMSE= 1835.4555595979605
Variance Score  0.0010134183900933829
Median Absalute Error= 1362.4365776890342
R2_Score 0.0010109796612431943
###GradientBoostingRegressor####
MAE= 1504.8561917443408
MSE= 3368916.651427726
RMSE= 1835.4608825653916
Variance Score  0.0010079101525193312
Median Absalute Error= 1361.818531222937
R2_Score 0.0010054555435584644
[14/Jul/2020 11:16:21] "GET /UserGraphs/ HTTP/1.1" 200 10089
###Decesion Treee####
MAE= 1504.8561917443408
MSE= 1504.8561917443408
RMSE= 1835.4608825653916
Variance Score  0.0010079058866440173
Median Absalute Error= 1361.8174099946264
```

Fig16: Server Side Result

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.

REFERENCES:

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

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

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

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

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

[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.

[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.

[8] A. M. Kalliguddi, Area K., Leboulluec, "Predictive Modelling of Aircraft Flight

Delay," Universal Journal of Management, pp. 485 - 491, 2017.

[9] 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.

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

[11] 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.

[12] [Online]. Available: http://scikitlearn.org/stable/modules/classes.html?source=post_page- - --- f10ba6e38234----------------------#sklearn-metrics-metrics.