**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.

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.

**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 neighbour 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 analyse 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 summarise 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 origindestination pairs and the forecast horizon. Sruti Oza, Somya Sharma [9] used multiple linear regression to predict weatherinduced 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.

**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.

**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. Fig. 1 shows some of the fields of the original dataset. There were many lines with missing and null values. The data must be pre-processed for later use. The methodology here uses the supervised learning technique to gather the advantages of having the schedule and real arrival time. Initially, some specific monitoring algorithms with a light computation cost were considered candidates and therefore the best candidate was perfected for the final model. We develop a system that predicts for a delay in flight departure based on certain parameters. We train our model for forecasting using various attributes of a particular flight, such as arrival performances, flight summaries, origin/destination, etc.

**Software &Hardware requirements**

**HARDWARE REQUIREMENTS:**

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

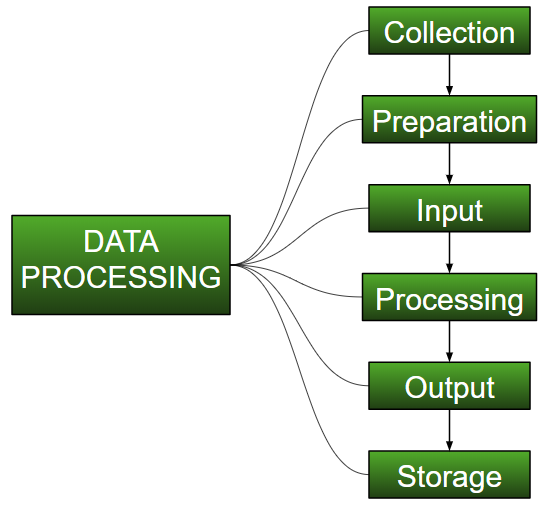
**SOFTWARE REQUIREMENTS:**

* Programming Language : Python
* IDE : pycharm/jupyter

**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.

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.



**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](https://data.gov.in/), [Kaggle](https://www.kaggle.com/) or [UCI dataset repository](https://archive.ics.uci.edu/ml/datasets.html). 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.

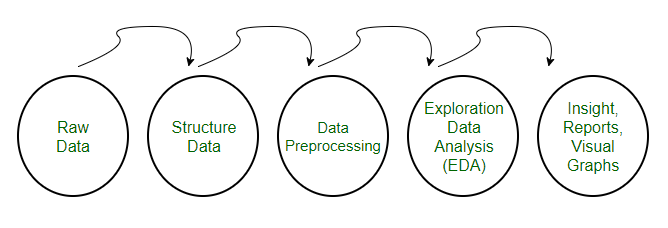
**Output:**  
In this stage, results are procured by the machine in a meaningful manner which can be inferred easily by the user. Output can be in the form of reports, graphs, videos, etc

**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.

**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.



**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.

Data Cleansing

Introduction:

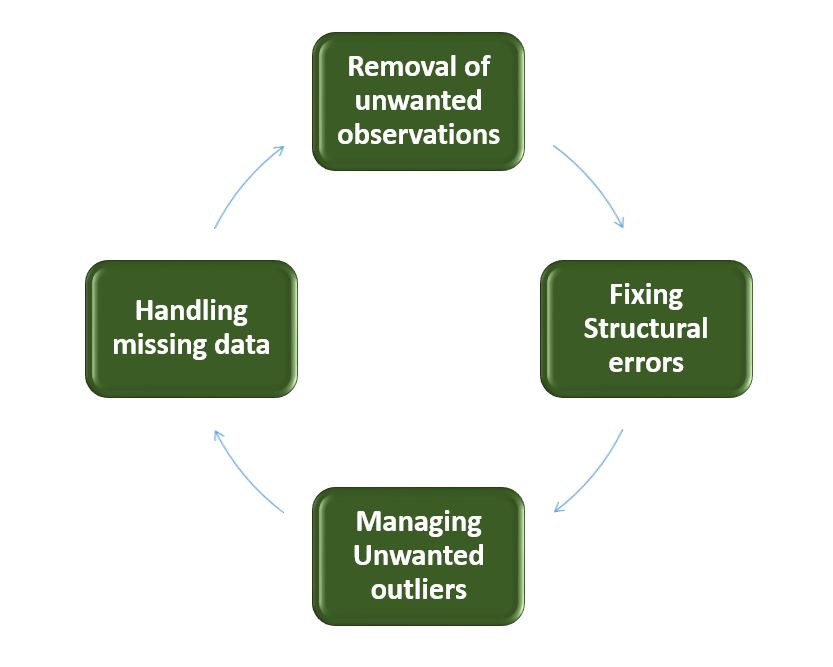
Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. Data Cleaning is one of those things that everyone does but no one really talks about. It surely isn’t the fanciest part of machine learning and at the same time, there aren’t any hidden 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.

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

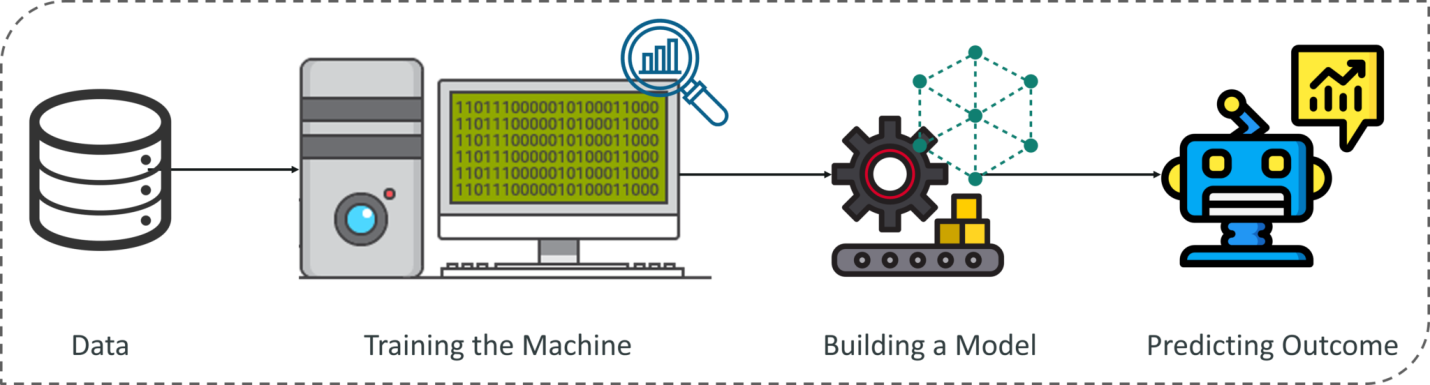
• IBM Infosphere Quality Stage

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.

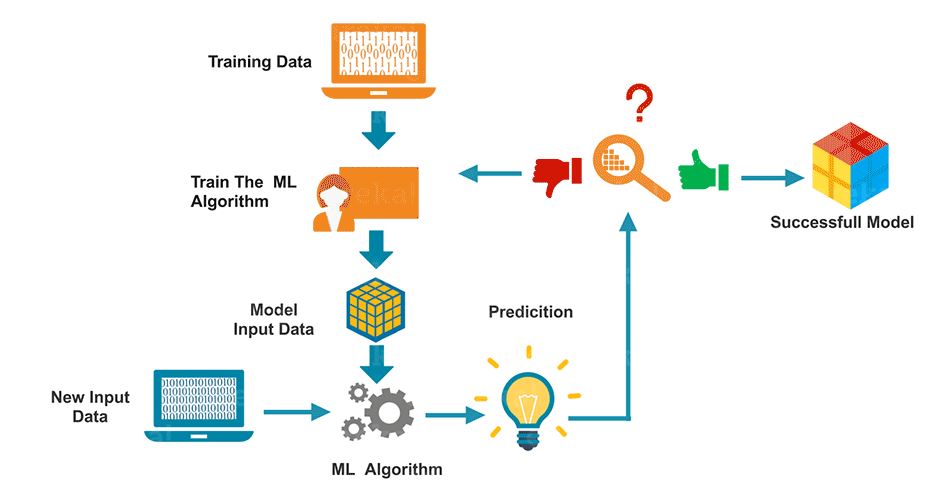
**Machine Learning Process**

****

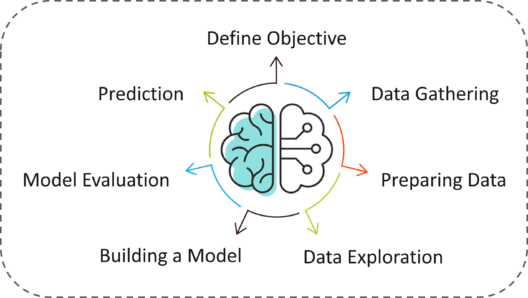
How does Machine Learning Work?

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.



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.



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 (eg. 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.

**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

**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.

**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.

**Nonfunctional requirement**

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 (.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
* Scalabilit

**INPUT DESIGN 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.

**SYSTEM STUDY**

**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.

Three key considerations involved in the feasibility analysis are

* 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.

**SOFTWARE REQUIREMENT SPECIFICATION**

**What is SRS?**

Software Requirements 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).

The SRS phase consists of two basic activities:

**Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

**Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, Specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

**Role of SRS:**

The purpose of the SRS is to reduce the communication gap between the clients and the developers. SRS is the medium though which the client and user needs are accurately specified.

It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

**Purpose:**

The purpose of this document is to describe all external requirements for the E-learning System. It also describes the interfaces for the system.

* Scope**:**

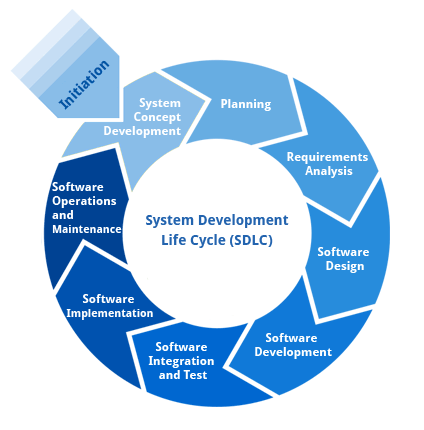
This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also by the basis for validating the final deliver system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alternations without the permission of the client.

* Overview**:**

The SRS begins the translation process that converts the software Requirements into the language the developers will use. The SRS draws on the Use Cases from the user Requirement Document and analyses the situations from a number of perspectives to discover and eliminate inconsistencies, ambiguities and omissions before development progresses significantly under mistaken assumptions.

**Software Development Life Cycle:**

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies use to develop these systems.



**Requirement Analysis and Design**

Analysis gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization. Options for changing the business process may be considered. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

**Implementation**

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like PYTHON 3.6, Anaconda Cloud are used for coding. With respect to the type of application, the right programming language is chosen.

**Testing**

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).

**Maintenance**

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

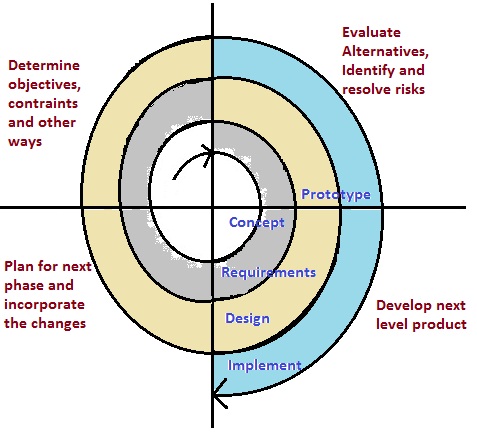
SDLC METHDOLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The following diagram shows how a spiral model acts like:



The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible.
* This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning a designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**Software Environment**

What is Python? Executive Summary

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**DJANGO:**

What is Django?

Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you normally will find with other frameworks.

When you’re building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use.

Why Django?

It’s very easy to switch database in Django framework.

It has built-in admin interface which makes easy to work with it.

Django is fully functional framework that requires nothing else.

It has thousands of additional packages available.

It is very scalable.

Popularity of Django

Django is used in many popular sites like as: Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic etc. There are more than 5k online sites based on the Django framework. ( Source )

Sites like Hot Frameworks assess the popularity of a framework by counting the number of GitHub projects and StackOverflow questions for each platform, here Django is in 6th position. Web frameworks often refer to themselves as “opinionated” or “un-opinionated” based on opinions about the right way to handle any particular task. Django is somewhat opinionated, hence delivers the in both worlds( opinionated & un-opinionated).

Features of Django

Versatility of Django

Django can build almost any type of website. It can also work with any client-side framework and can deliver content in any format such as HTML, JSON, XML etc. Some sites which can be built using Django are wikis, social networks, new sites etc

Security

Since Django framework is made for making web development easy, it has been engineered in such a way that it automatically do the right things to protect the website. For example, In the Django framework instead of putting a password in cookies, the hashed password is stored in it so that it can’t be fetched easily by hackers.

Scalability

Django web nodes have no stored state, they scale horizontally – just fire up more of then when you need them. Being able to do this is the essence of good scalability. Instagram and Disqus are two Django based products that have millions of active users, this is taken as an example of the scalability of Django.

Portability

All the codes of the Django framework are written in Python, which runs on many platforms. Which leads to run Django too in many platforms such as Linux, Windows and Mac OS.

**Tomcat Server:**

Introduction

TomCat, in its simplest concept, is a web server. Well, it's more than that. While it can serve static files, its primary purpose is to act as a 'servlet container' that serves Java web applications. It can process [.jsp] files, which are like PHP scripts but for Java, and it can also run [Java Servlets], which are classes that process the GET, POST, and other HTTP requests. TomCat will listen on TCP ports for HTTP requests, and route requests to your Java classes, JSP files, or static files. It is also possible to embed TomCat in to a standalone application, but that is out of the scope of this document.

Beginner level Java knowledge is expected, although there is virtually no Java code in this tutorial specfically. This will cover basic installation, configuration, and admin aspects. This is aimed at beginner to intermediate developers who want to learn about basic Java web app hosting. You should either be a system administrator setting up TomCat or a developer learning how to set up TomCat.

I would always recommend using the official documentation at http://tomcat.apache.org/ for the latest and most accurate information. This is meant as a quick start guide that includes the things I think are important to know to get started. There are a number of features that are not covered here. For example, TomCat can be embedded inside your application.One popular alternative to TomCat is Eclipse Jetty. It is often used when you need a lightweight embedded servlet container.

Deploying Django with Apache and mod\_wsgi is a tried and tested way to get Django into production.

mod\_wsgi is an Apache module which can host any Python WSGI application, including Django. Django will work with any version of Apache which supports mod\_wsgi.

The official mod\_wsgi documentation is your source for all the details about how to use mod\_wsgi. You’ll probably want to start with the installation and configuration documentation.

**About SQLite**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. SQLite database files are a recommended storage format by the US Library of Congress. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()

SQLite is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.

SQLite is very carefully tested prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage. SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are ACID even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides bugs lists and minute-by-minute chronologies of code changes.

The SQLite code base is supported by an international team of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the published interface spec, SQL syntax, and database file format. The source code is absolutely free to anybody who wants it, but professional support is also available.

The SQLite project was started on 2000-05-09. The future is always hard to predict, but the intent of the developers is to support SQLite through the year 2050. Design decisions are made with that objective in mind.

We the developers hope that you find SQLite useful and we entreat you to use it well: to make good and beautiful products that are fast, reliable, and simple to use. Seek forgiveness for yourself as you forgive others. And just as you have received SQLite for free, so also freely give, paying the debt forward.

**Introduction To PyCharm**

Many programmers nowadays opt for Python to build software applications with the concise, clean, and readable code base. They can even accelerate custom software application development by taking advantage of a number of integrated development environments (IDEs) for Python.

JetBrains has developed PyCharm as a cross-platform IDE for Python. In addition to supporting versions 2.x and 3.x of Python, PyCharm is also compatible with Windows, Linux, and macOS. At the same time, the tools and features provided by PyCharm help programmers to write a variety of software applications in Python quickly and efficiently. The developers can even customize the PyCharm UI according to their specific needs and preferences. Also, they can extend the IDE by choosing from over 50 plug-ins to meet complex project requirements. PyCharm is one of the most widely used IDEs for Python programming language. At present, the Python IDE is being used by large enterprises like Twitter, Pinterest, HP, Symantec, and Groupon.

**Libraries:**

**Pandas:**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Key Features of Pandas**

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**Numpy:**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionality. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

**Operations using NumPy**

* Using NumPy, a developer can perform the following operations −
* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**Keras:**

Deep learning is one of the major subfield of machine learning framework. Machine learning is the study of design of algorithms, inspired from the model of human brain. Deep learning is becoming more popular in data science fields like robotics, artificial intelligence (AI), audio & video recognition and image recognition. Artificial neural network is the core of deep learning methodologies. Deep learning is supported by various libraries such as Theano, TensorFlow, Caffe, Mxnet etc., Keras is one of the most powerful and easy to use python library, which is built on top of popular deep learning libraries like TensorFlow, Theano, etc., for creating deep learning models.

**Overview of Keras**

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

**Features**

* Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features −
* Consistent, simple and extensible API.
* Minimal structure - easy to achieve the result without any frills.
* It supports multiple platforms and backends.
* It is user friendly framework which runs on both CPU and GPU.
* Highly scalability of computation.

**Benefits**

* Keras is highly powerful and dynamic framework and comes up with the following advantages −
* Larger community support.
* Easy to test.
* Keras neural networks are written in Python which makes things simpler.
* Keras supports both convolution and recurrent networks.
* Deep learning models are discrete components, so that, you can combine into many ways.

**TensorFlow:**

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

Let us now consider the following important features of TensorFlow −

* It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
* It includes a programming support of deep neural networks and machine learning techniques.
* It includes a high scalable feature of computation with various data sets.
* TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.

**OpenCV**

OPEN CV IS a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as [Numpy](https://www.geeksforgeeks.org/python-numpy/" \t "_blank) which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

**Why is TensorFlow So Popular?**

TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same.

TensorFlow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

**Pypoly:**

Plotly is a Montreal based technical computing company involved in development of data analytics and visualisation tools such as Dash and Chart Studio. It has also developed open source graphing Application Programming Interface (API) libraries for Python, R, MATLAB, Javascript and other computer programming languages.

Some of the important features of Plotly are as follows −

* It produces interactive graphs.
* The graphs are stored in JavaScript Object Notation (JSON) data format so that they can be read using scripts of other programming languages such as R, Julia, MATLAB etc.
* Graphs can be exported in various raster as well as vector image formaats

**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.

**SOFTWARE DESIGN**

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consists 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.

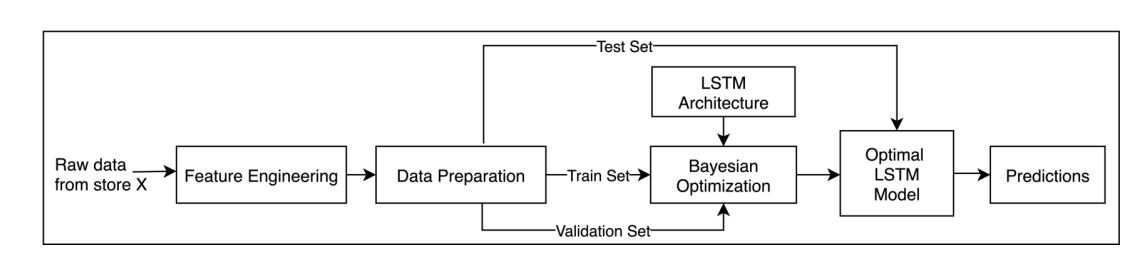
4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provide

**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.



**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.

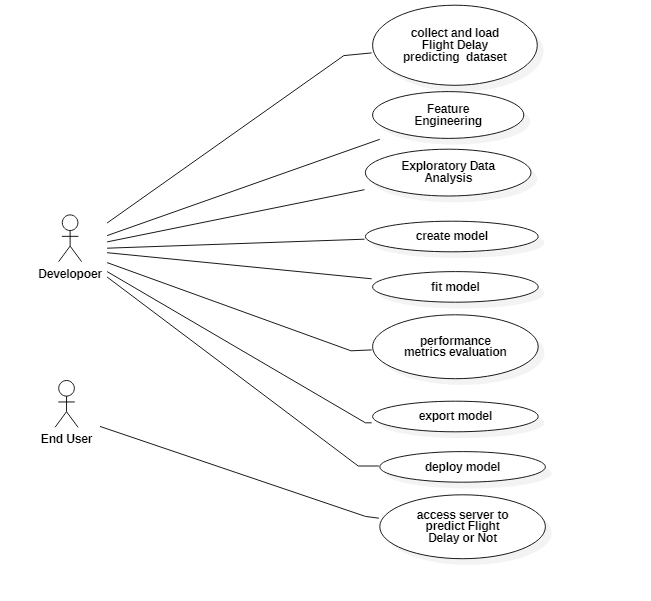
Encourage the growth of OO tools market.

Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.

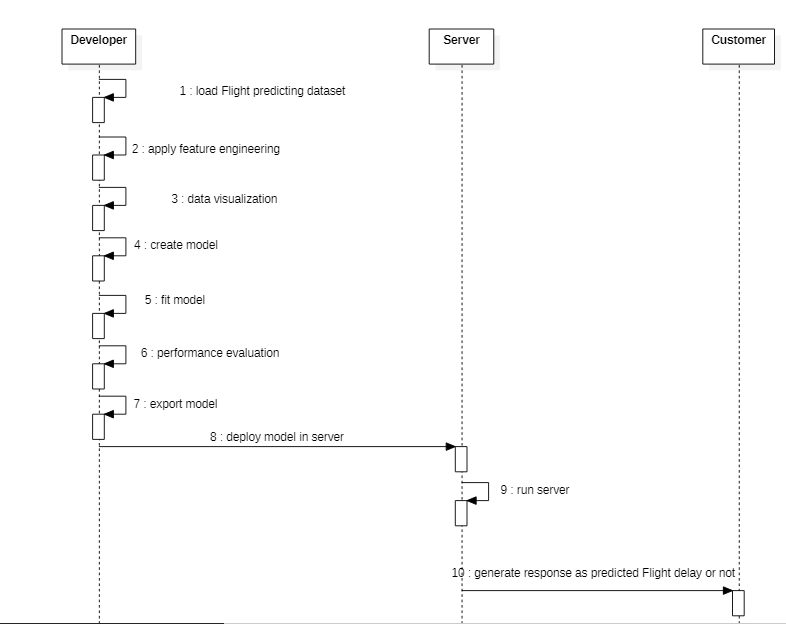
**Use Case Diagram:**

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.

****

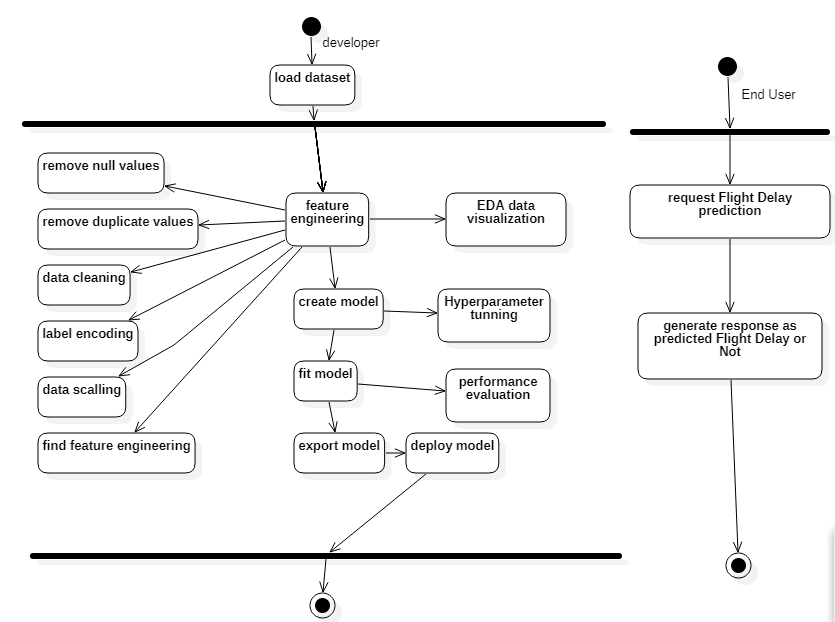
**Sequence diagram:**

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.

****

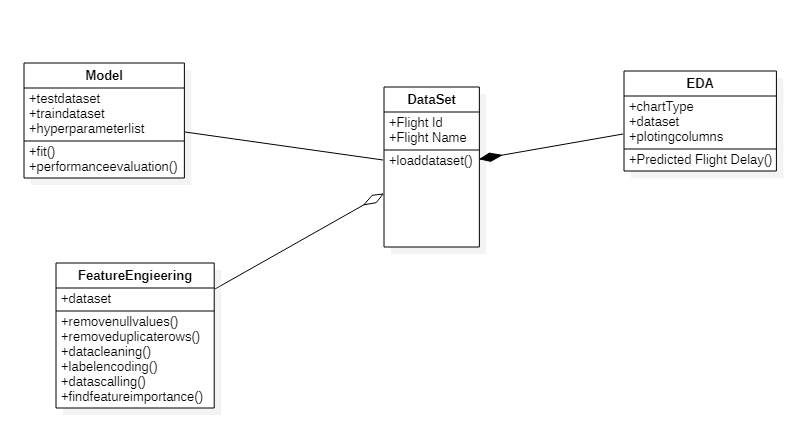
**Activity diagram:**

* 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.

****

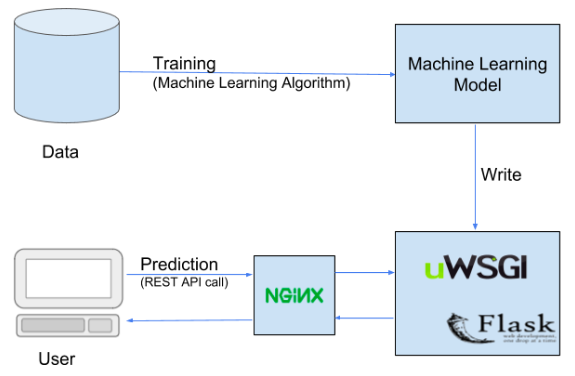
**CLASS DIAGRAM:**

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.

****

**Deployment diagram**

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



Packaging your ML model

Securing your packaged ML model

Planning of how to serve/expose your ML model to the consumers (as a REST service, etc.)

If you have planned for a REST service, then, creating a REST API for your ML model

Securing your REST API

Deploying your REST service in production (using Docker and Kubernetes)

Packaging your ML model

Instead of saving the ML model as it is, you can package your ML model (say mnist), and create a .pkl file for it, using Python’s joblib library.

For creating and restoring the .pkl file, you can either use joblib library or pickle library of Python. You normally use joblib to save an object having large data, else, you use pickle library. Here, in this case, we have used joblib library.

.pkl file is nothing but a serialized pickle file, which if you want, you can compress it further, to save storage space, using say Python’s gzip library. After you apply compression, your ML model file name will look like – mnist.pkl.gz

Securing your packaged ML model

The content of your pickle file (in this case your ML model) could be intercepted and modified by anyone over the network if your network is not secured. Hence, it is advisable to use secured (encrypted) network connection while exchanging the pickle file.

In addition to this, the pickle file could be signed (using ‘cryptographic signature’) before storing or transmitting, and this signature can be verified before it is restored at the receiver’s end (say your REST API). Cryptographic signature helps in detecting any alterations to your pickle file data.

Cryptographic signature uses a cryptographic algorithm which generates a cryptographic hash of your pickle file data along with shared secret key. SHA-1 cryptographic algorithm is considered to be the best algorithm to create a stronger hash, hence, it is highly advisable to use it.

**Test Cases**

**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 pag

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. 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 test**

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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 Test**

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

**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.

CONCLUSION AND FUTURE WORKS

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

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

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

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

**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 manoeuvring 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-- ---f10ba6e38234----------------------#sklearn-metrics-metrics.