**A**

**MAJOR PROJECT REPORT ON**

**PREDICTION OF MEDICAL COSTS USING REGRESSION ALGORITHMS**

***Submitted partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

***In***

**COMPUTER SCIENCE & ENGINEERING**

***By***

**K. ASHWINI**

**2020 - 2021**

**ABSTRACT**

Heath care costs increases day by day. As there are a greater number of new viruses entering into people, there is a need to predict health charges. This type of prediction helps the governments to make a decision regarding health issues. People also knows the importance of health care costs. Machine Learning is a filed which has its impact on every filed. Health care system also uses machine learning models for several health related applications. In this paper, we have done predicate analysis on medical health insurance charges. We build a model to predict the medical insurance cost of a person based on gender. We collect the dataset from Kaggle, which contains 1338 rows of data with the features age, gender, smoker ,BMI, children, region, insurance charges. The data contains medical information and costs billed by health insurance companies. We applied various regression algorithms on this dataset to predict medical costs.

**Key Phrases**: [Medical insurance costs, [Kaggle,](https://easychair.org/publications/keyword/vNtx) Machine Learning.,](https://easychair.org/publications/keyword/7HjR)

Table of Contents

[CHAPTER 1 1](#_Toc75200628)

[INTRODUCTION 1](#_Toc75200629)

[1.1 INTRODUCTION 2](#_Toc75200630)

[CHAPTER 2 3](#_Toc75200631)

[LITERATURE SURVEY 3](#_Toc75200632)

[2.1 LITERATURE REVIEW 4](#_Toc75200633)

[2.2 EXSTING SYSTEM 4](#_Toc75200634)

[2.3 PROPOSED SYSTEM: 5](#_Toc75200635)

[CHAPTER 3 6](#_Toc75200636)

[SYSTEM ANALYSIS 6](#_Toc75200637)

[3.1 SOFTWARE REQUIREMENTS: 7](#_Toc75200638)

[3.2 HARDWARE REQUIREMENTS: 7](#_Toc75200639)

[CHAPTER 4 8](#_Toc75200640)

[SYSTEM DESIGN 8](#_Toc75200641)

[4.1 ARCHITECTURE OF PROPOSED SYSTEM 9](#_Toc75200642)

[4.2 MODULES 11](#_Toc75200643)

[4.6 SUPPORT VECTOR MACHINE 15](#_Toc75200644)

[CHAPTER 5 18](#_Toc75200645)

[5.1 UML DIAGRAMS 19](#_Toc75200646)

[5.2 SEQUENCE DIAGRAM: 19](#_Toc75200647)

[5.3 ACTIVITY DIAGRAMS: 20](#_Toc75200648)

[5.4 USECASE DIAGRAM: 21](#_Toc75200649)

[5.5 CLASS DIAGRAM: 21](#_Toc75200650)

[CHAPTER 6 23](#_Toc75200651)

[SYSTEM IMPLEMENTATION 23](#_Toc75200652)

[6.1 MACHINE LEARNING 24](#_Toc75200653)

[6.2 DEEP LEARNING 31](#_Toc75200654)

[6.3 TENSOR FLOW 35](#_Toc75200655)

[6.4 PYTHON OVERVIEW 36](#_Toc75200656)

[6.6 SOURCE CODE 50](#_Toc75200657)

[CHAPTER 7 55](#_Toc75200658)

[TESTING 55](#_Toc75200659)

[7.1 TESTING METHODS 56](#_Toc75200660)

[TYPES OF TESTING 56](#_Toc75200661)

[7.1.1 UNIT TESTING 56](#_Toc75200662)

[7.1.2 INTEGRATION TESTING 56](#_Toc75200663)

[7.1.3 FUNCTIONAL TESTING 56](#_Toc75200664)

[CHAPTER 8 58](#_Toc75200665)

[CONCLUSION 58](#_Toc75200666)

[CONCLUSION 59](#_Toc75200667)

[CHAPTER 9 60](#_Toc75200668)

[REFERENCES 60](#_Toc75200669)

[9.1 REFERENCES 61](#_Toc75200670)

**FIG NO NAME PAGE NO**

Fig 4.1 BLOCK DIAGRAM 9

#### Fig 4.1.2 ARCHITECUTE SYSTEM 9

Fig 4.2 DATA INGESTION 11

Fig 4.3 A SIMPLE LINEAR REGRESSION RANDOM FOREST 13

Fig 4.4 SIMPLE RANDOM FOREST CLASSIFIER 14

Fig 4.5 SIMPLE DECISION TREE 15

Fig 4.6 MARGIN 16

Fig 5.2 SEQUENCE DIAGRAM 20

Fig 5.3 ACTIVITY DIAGRAM 20

Fig 5.4 USECASE DIAGRAM 21

Fig 5.5 CLASSDIAGRAM 21

Fig 6.1 MACHINE LEARNING 23

Fig 6.1.1 LEARNING PHASE 24

Fig 6.1.2 INFERENCE FROM MODEL 24

Fig 6.1.3 Machine learning Algorithms and there uses 25

Fig 6.2 ARTIFICIAL INTELLIGENCE31

Fig 6.2.1 GRAPH 33

Fig 7.1 SMOKER 41

Fig 7.1.1 SEX 42

# 

# CHAPTER 1

# INTRODUCTION

### 1.1 INTRODUCTION

As indicated by the World Bank, the absolute use on medicinal services as an extent of GDP in 2015 was 3.89%. Out of 3.89%, the legislative well-being consumption as an extent of GDP is simply 1%, and the cash-based use as an extent of the present well-being use was 65.06% in 2015. Throughout the most recent couple of decades, the progression in clinical innovation has made it conceivable to fix illnesses that were once viewed as serious. In any case, the expense of their treatment is so high, it is practically incomprehensible for a white collar class individual to manage the cost of them. As indicated by insights, Rs 5 lakh family floater strategy will cover self, mate and one kid will cost any place between Rs 10,000 and Rs 17,000 on a yearly premise though Rs. 5 lakh singular well-being plan will cost a multi year old Rs. 4,000-7,000 per year.

# 

# CHAPTER 2

# LITERATURE SURVEY

### 2.1 LITERATURE REVIEW

Machine Learning is a technology where machines can learn from the previous data and predict new samples. Machine Learning models are applicable in all fields. Medical files also not having any exclusion to machine learning. Medical field using ML models in different situation from last several years. Many of the researchers applied machine learning techniques to medical related cos-prediction. B. Nithya [1] et.al applied machine learning models in predictive Analytics in Health Care. They applied various supervised and unsupervised models for predictive analysis. They also suggested machine learning tools and techniques are decisive in health care province and exclusively used in the diagnosis and predictions of various types of cancers. Anuja Tike[2] et.al applied hierarchical decision tress for medical price prediction system. Their experiments shown that the price prediction system achieves high accuracy. Moran et al. [3] utilized linear regression techniques to anticipate Intensive Care Unit (ICU) expenses and utilize understanding socioeconomics, DRG (Diagnostic Related Group), length of stay in the clinic and a couple of others as highlights. Gregor [4] et.al applied various regression models for analyzing medical costs in health care system. They mainly concentrated on reduce the bias in the cost estimates to achieve good results. Dimitris Bertsimas[5] et.al applied different data mining techniques which provided an accurate predictions of medical costs and represent a powerful tool for prediction of health-care cos

### 2.2 EXSTING SYSTEM

Previously they applied various supervised and unsupervised models for predictive analysis. They also suggested machine learning tools and techniques are decisive in health care province and exclusively used in the diagnosis and predictions of various types of cancers. One approach applied hierarchical decision tress for medical price prediction system. Their experiments shown that the price prediction system achieves 70% accuracy. Machine Learning is a technology where machines can learn from the previous data and predict new samples. Machine Learning models are applicable in all fileds. Medical files also not having any exclusion to machine learning. Medical field using ML models in different situation from last several years. Many of the researchers applied machine learning techniques to medical related cost prediction. B. Nithya [1] et.al applied machine learning models in predictive Analytics in Health Care. They applied various supervised and unsupervised models for predictive analysis. They also suggested machine learning tools and techniques are decisive in health care province and exclusively used in the diagnosis and predictions of various types of cancers. Anuja Tike[2] et.al applied hierarchical decision tress for medical price prediction system. Their experiments shown that the price prediction system achieves high accuracy. Moran et al. [3] utilized linear regression techniques to anticipate Intensive Care Unit (ICU) expenses and utilize understanding socioeconomics, DRG (Diagnostic Related Group), length of stay in the clinic and a couple of others as highlights. Gregori [4] et.al applied various regression models for analyzing medical costs in health care system. They mainly concentrated on reduce the bias in the cost estimates to achieve good results. Dimitris Bertsimas[5] et.al applied different data mining techniques which provided an accurate predictions of medical costs and represent a powerful tool for prediction of health-care costs.

### 2.3 PROPOSED SYSTEM:

The dataset used for experiments is collected from Kaggle[6] machine learning repository. This dataset was inspired by the book Machine Learning with R by Brett Lantz. The data contains medical information and costs billed by health insurance companies. It contains 1338 rows of data and the following columns: age, gender, BMI, children, smoker, region, insurance charges . In these features insurance charges is a dependent variable and the remaining features are called as independent variables. In regression analysis, we need to predict the value of dependent variable using independent variables. First, we collected dataset and applied various data preprocessing methods. Data preprocessing is a technique in which we can remove missing values in the data. Because of these missing values, it is not possible to apply machine learning algorithms. After removal of missing values, we need to apply label encoding, one hot encoding data to the categorical features. Categorical features are the features whose values are labels instead of values. After that, apply standardization or normalization techniques to our data. This method is used when all the attribute values are not in the same scale. In regression analysis, we need to predict the value of dependent variable using independent variables. First, we collected dataset and applied various data preprocessing methods. Then we applied following four regression models on the dataset. i) Linear Regression ii) Support Vector Regression iii) Decision Tree Regression iv) Random Forest Regression.

# 

# CHAPTER 3

# SYSTEM ANALYSIS

### 3.1 SOFTWARE REQUIREMENTS:

* + - System : INTEL Processor
    - Hard Disk : 200 GB.
    - Monitor : 15’’ LED
    - Input Devices : Keyboard, Mouse
    - RAM : 4GB
    - NETWORK: LAN

## 3.2 SOFTWARE REQUIREMENTS:

* + - Operating system : Windows 7,8 AND 10(32 AND 64 BIT)/Windows XP
    - Coding Language : Python
    - GUI : Python GUI or Anaconda Navigator

**3.3 SYSTEM REQUIREMENT:**

**Operating System:** Windows 7 Ultimate 32 bit / Windows XP

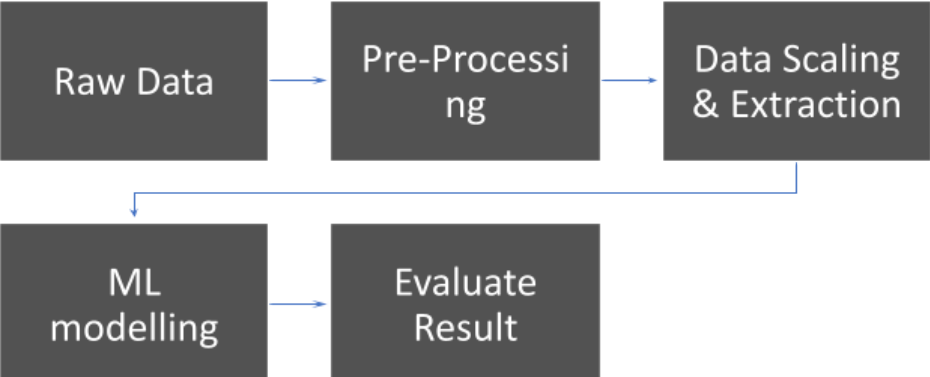
# 

# CHAPTER 4

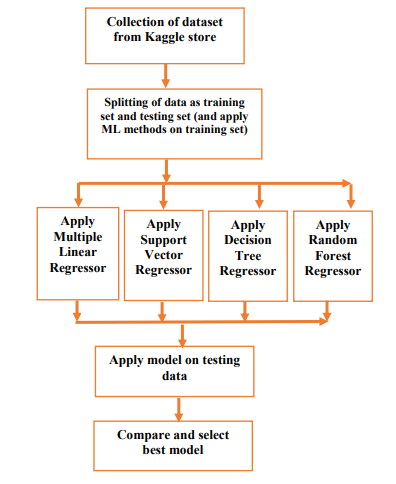
# SYSTEM DESIGN

### 4.1 ARCHITECTURE OF PROPOSED SYSTEM

**Block Diagram :**



#### FIG-4.1 BLOCK DIAGRAM



#### FIG-4.1.2 ARCHITECUTE SYSTEM

**STEPS for Proposed Approach**

Step 1:-Initialize the dataset containing training data wholesale price index

Step 2:-Select all the rows and column 1from dataset to “x” Which is independent variable

Step 3:-Select all of the rows and column 2 from dataset to “y” Which is dependent variable

Step 4:- Fit DTR/SVR/LR/RF to the dataset

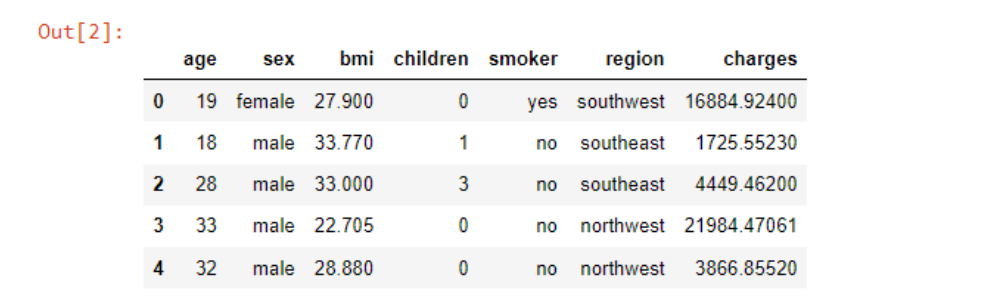
Step 5:-Predict the new value

Step 6:-Visualize the result and check the accuracy

### 4.2 MODULES

**1. Data Ingestion:**

Data ingestion is the transportation of data from assorted sources to a storage medium where it can be accessed, used, and analyzed by an organization. The destination is typically a data warehouse, data mart, database, or a document store. Sources may be almost anything – including SaaS data, in-house apps, databases, spreadsheets, or even information scraped from the internet. The data ingestion layer is the backbone of any analytics architecture. Downstream reporting and analytics systems rely on consistent and accessible data. There are different ways of ingesting data, and the design of a particular data ingestion layer can be based on various models or architectures.



**FIG-4.2 DATA INGESTION**

**2. Data Preprocessing:**

Data Preprocessing is a data mining technique used to transform the raw data into useful and efficient format. The data here goes through 2 stages:

1. Data Cleaning: It is very important for data to be error free and free of unwanted data. So, the data is cleansed before performing the next steps. Cleansing of data includes checking for missing values, duplicate records and invalid formatting and removing them.

2. Data Transformation: Data Transformation is transformation of the datasets mathematically; data is transformed into appropriate forms suitable for data mining process. This allows us to understand the data more keenly by arranging the 100‟s of records in an orderly way. Transformation includes Normalization, Standardization, Attribute Selection.

**3. Exploratory data analysis:**

Exploratory data analysis(EDA) is an approach to understand the datasets more keenly by the means of visual elements like scatter plots, bar plots, etc. This allows us to identify the trends in the data more accurately and to perform analysis accordingly. From the yearly trends graphs it is observed that, US Exports depend on and follows the areas planted and harvested annually. A sudden drop in China ‘s Exports in the year 2009 is observed and in the mean time its imports kept increasing in the last 12 years regardless of the global yield, which implies China has a huge and lasting demand of soybean crop but now it relies on the global supply to meet the needs.

4. **Data Splitting and Modelling**

Modelling of data involves creating a data model for the data to be stored in the database. The process of modeling means training a Machine Learning Algorithm to predict the labels from the features, tuning it for business need, and validating it on the hold out data. The output from modeling is a trained model that can be used for inference, making predictions on new data points. Modeling is independent of the previous steps in the Machine Learning process and has standardized inputs which means we can alter the prediction problem without needing to rewrite all our code. If the business requirements change, we can generate new label times, build corresponding features, and input them into the model. Regressors used for prediction purpose :

o Random Forest Regressor- regression method

o Support Vector Regression (SVR) – uses kernel functions

o Linear Regression – regression method

o Decision Tree Regression – regression method

5. **Evaluation Metric**

Models are implemented and later evaluated for their accuracies using root mean square error Since this is multi classification problem, we use the following metrics:

**R2 score -** The r2-score of a regression is the percentage of the test set

tuples that are correctly classified by the regressor.

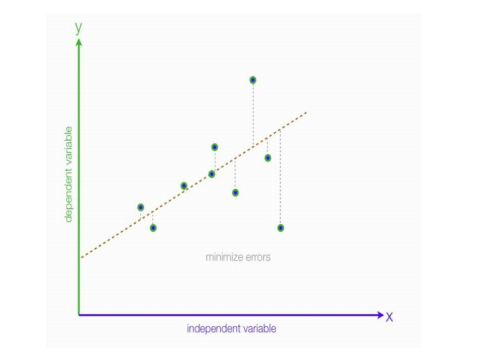
**4.3 LINEAR REGRESSION**

In statistics, a linear regression is a method to find a relationship between a scalar dependent variable y and one or more explanatory variables or independent variables called X. When there is only one independent variable it is called a simple linear regression. On the other hand, when there are more than one independent variables, it is called as a multiple linear regression. In machine learning, a linear regression model tries to model the relationship between two variables by fitting a linear equation to observed data. Observed data is the training data. For example, a modeler might want to relate the area of a rectangles to their heights using linear regression. A linear regression line has an equation of the form,

Y = c + mx

where X is the independent variable and Y is the dependent variable. The slope of the line is m, and c is the intercept that is the value of y when x = 0. In machine learning, we pass the training data to a model and train it so as when the new data come a model predicts the outcome for it. For example, (x1, y1), (x2, y2), (x3, y3) till (xm, ym) can be training data. Then the model uses these set of points to find the coefficient m and the constant c such that they fit in the equation mentioned above. Once it is done, when the new point (xn) is given as a test record the model predicts the value of yn for this value for x. The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line. If a point lies on the fitted line exactly, then its vertical deviation is 0.

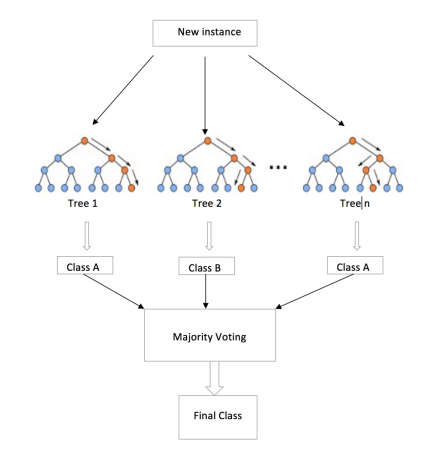
Figure below shows simple linear regression .



**FIG-4.3 A SIMPLE LINEAR REGRESSION RANDOM FOREST**

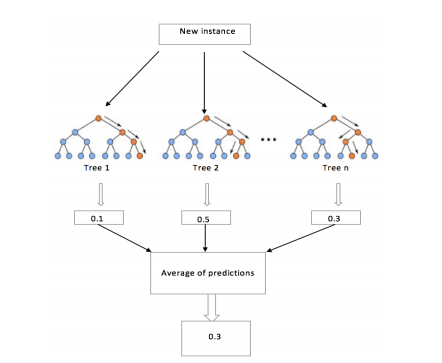
**4.4 RANDOM FORESTS**

Ensemble Methods The Random Forests is called ensemble methods, for constructing multiple decision trees. Ensemble methods take multiple weak learners, such as decision trees, and construct a strong learner from them such as random forest. The Random Forests and gradient-boosted trees both can be used for classification and regression task. In ensemble methods a classification task for a new instance will be done by taking the majority of votes from each tree. The class getting the highest votes will be chosen as the final target 12 value for the new instance. Below Fig. shows a simple illustration of random forest classifier.



**FIG-4.4 SIMPLE RANDOM FOREST CLASSIFIER4.5 DECISION TREE**

On the other hand, in a regression task, the new instance is passed through all the trees and the outcomes from all the individual trees are aggregated to produce an overall outcome. Below Fig. shows a simple illustration of random forest regressor. Like decision trees, we can calculate the importance of each feature in the ensemble method also. They are calculated by computing the importance of features of individual trees and then averaging them across the trees. Ensemble methods are more robust because they decrease the tendency of a single decision tree to overfit the training data.



**FIG-4.5 SIMPLE DECISION TREE**

### 

### 4.6 SUPPORT VECTOR MACHINE

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line

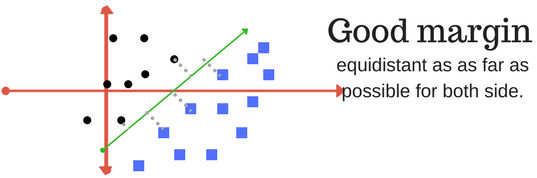
• **Margin**

And finally last but very important characteristic of SVM classifier. SVM to core tries to achieve a good margin.

• **A margin is a separation of line to the closest class points.**

• A **good margin**is one where this separation is larger for both the classes.

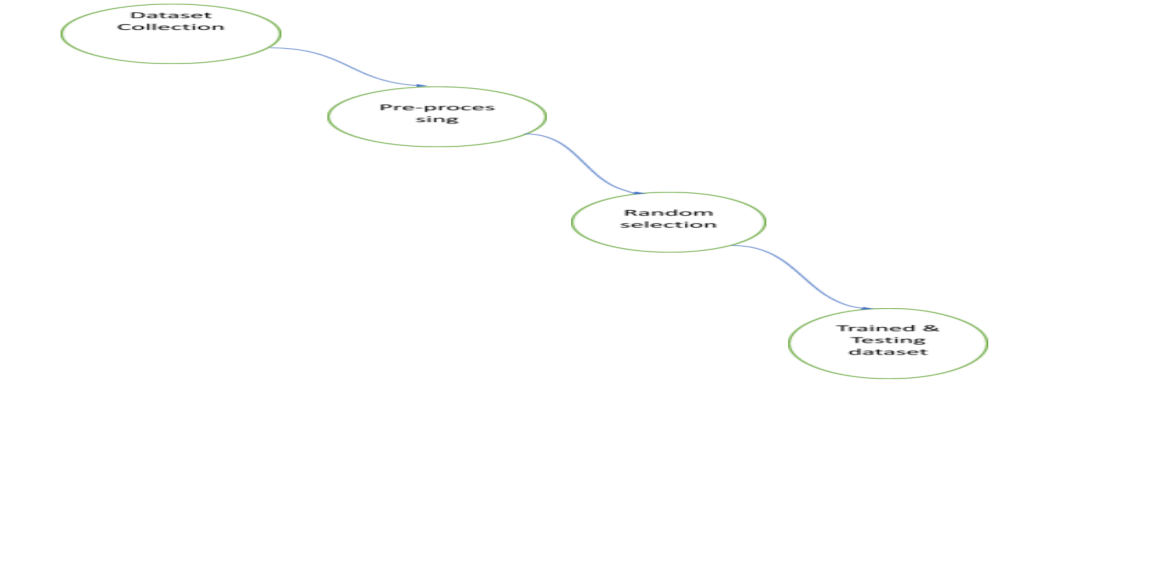
Images below gives to visual example of good and bad margin. A good margin allows the points to be in their respective classes without crossing to other class.



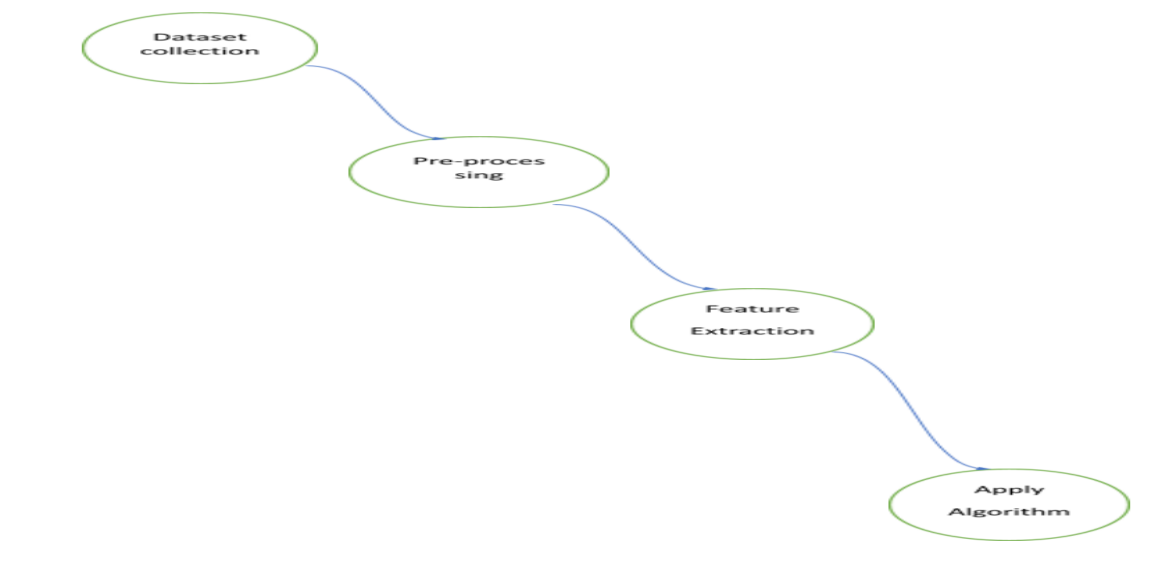
**FIG-4.6 MARGIN**

**4.7** **DATA FLOW DIAGRAM**

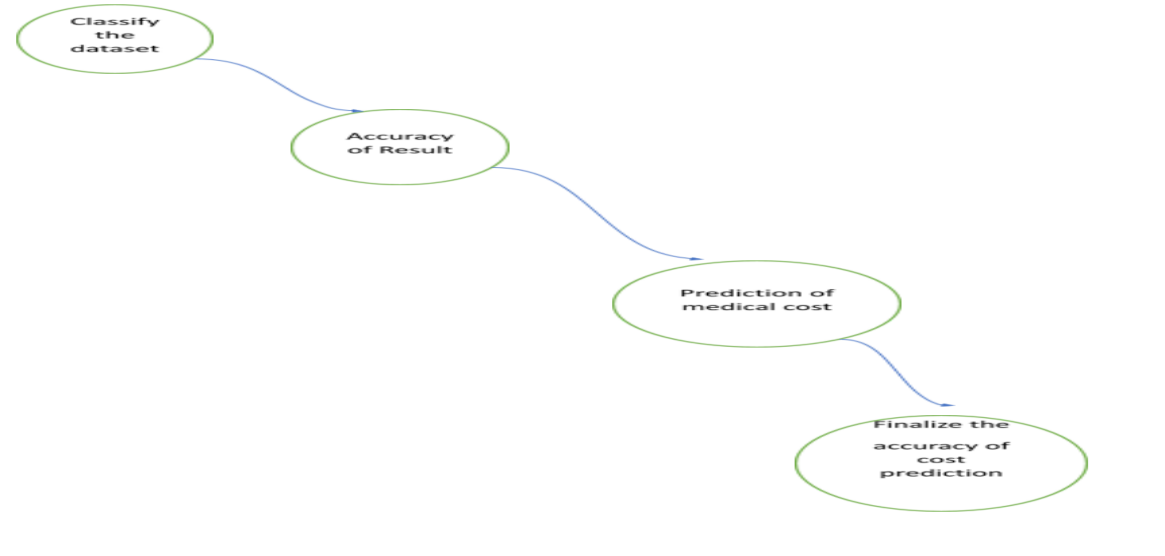
**LEVEL 0**



**LEVEL 1**



**LEVEL 2**



# CHAPTER 5

**UML DIAGRAMS**

## 5.1 UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

● actors

● business processes

● (logical) components

● activities

● programming language statements

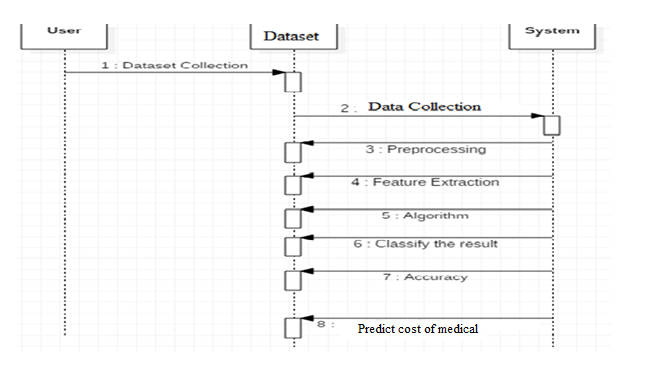
● database schemas, and

● Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

## 5.2 SEQUENCE DIAGRAM:

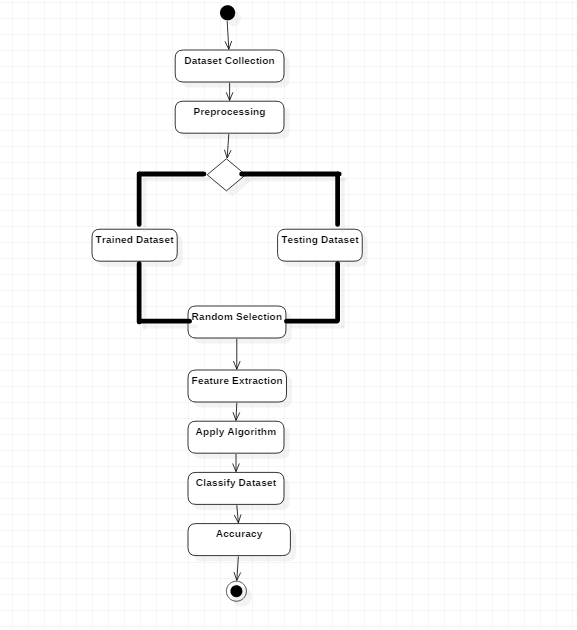
Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in 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. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.



**FIG-5.2 SEQUENCE DIAGRAM**

## 5.3 ACTIVITY DIAGRAMS:

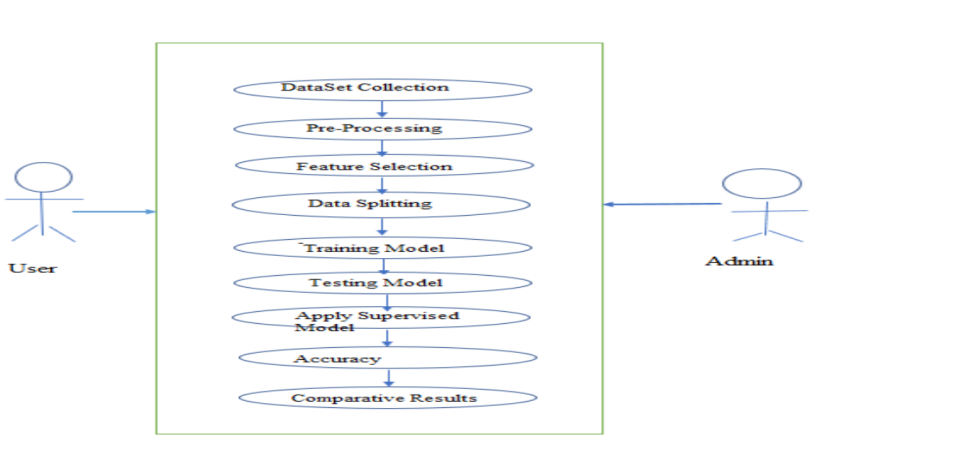
Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.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



**FIG-5.3 ACTIVITY DIAGRAM**

## 5.4 USECASE DIAGRAM:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. OMG is continuously putting effort to make a truly industry standard. UML stands for **U**nified **M**odeling **L**anguage. UML is a pictorial language used to make software blue prints.



**FIG-5.4 USECASE DIAGRAM**

## 5.5 CLASS DIAGRAM:

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. In the diagram, classes are represented with boxes that contain three compartments: The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.



**FIG-5.5 CLASS DIAGRAM**

# 

# CHAPTER 6

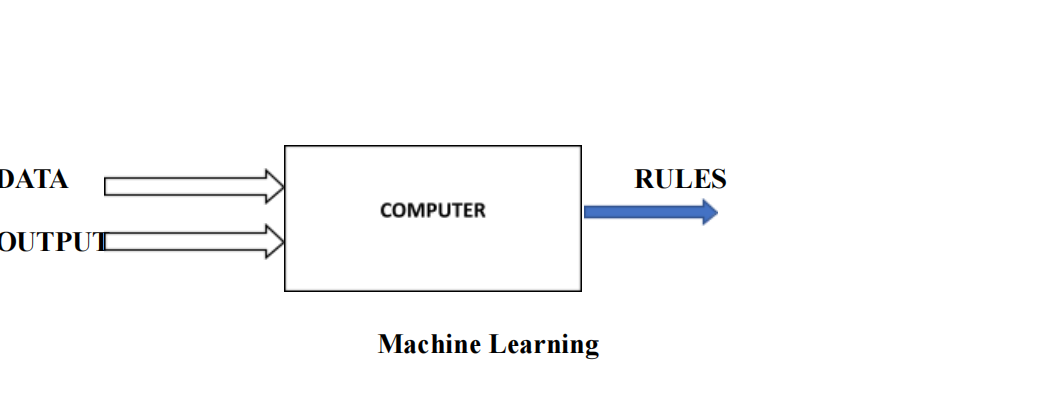
## SYSTEM IMPLEMENTATION

## 6.1 MACHINE LEARNING

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. receives data as input, use an algorithm to formulate answers. A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation. Machine learning is also used for a variety task to like fraud detection, predictive maintenance, portfolio optimization, automatize task.

**Machine Learning vs. Traditional Programming :**

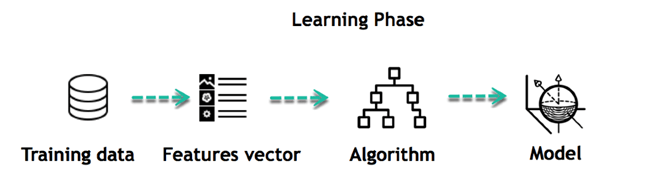
Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.



**FIG-6.1 MACHINE LEARNING**

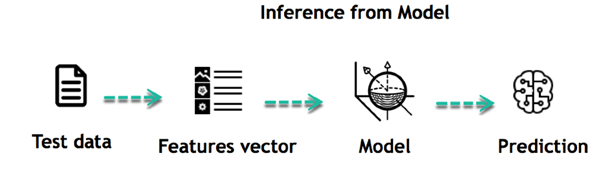
**How does Machine learning work?**

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if it’s feed a previously unseen example, the machine has difficulties to predict. The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used describe the data and summarize it into a model.



**FIG-6.1.1 LEARNING PHASE**

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model Inferring When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.



**FIG-6.1.2 INFERENCE FROM MODEL**

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question

2. Collect data

3. Visualize data

4. Train algorithm

5. Test the Algorithm

6. Collect feedback

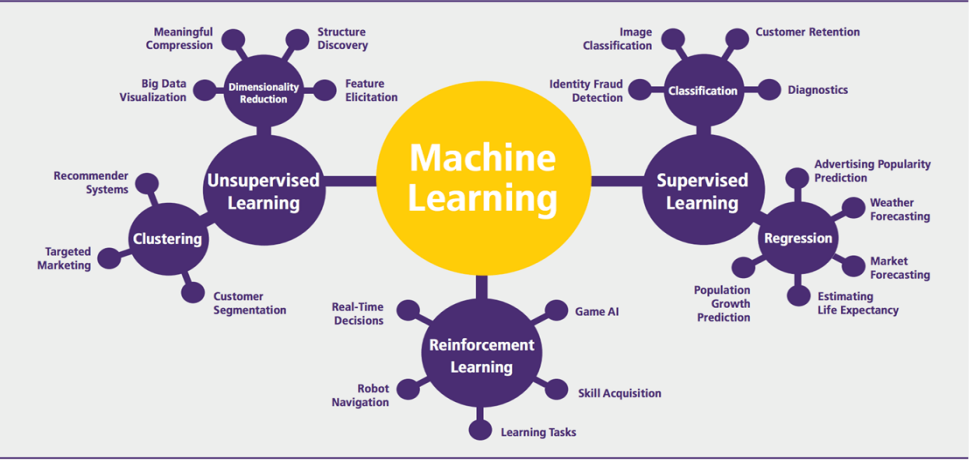
7. Refine the algorithm

8. Loop 4-7 until the results are satisfying

9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data. Machine learning can be grouped into two broad learning tasks: Supervised and

Unsupervised. There are many other algorithms



**FIG-6.1.3 Machine learning Algorithms and there uses**

**Supervised learning** :

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans. You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

1.Classification task

2. Regression task

1. **Classification**

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

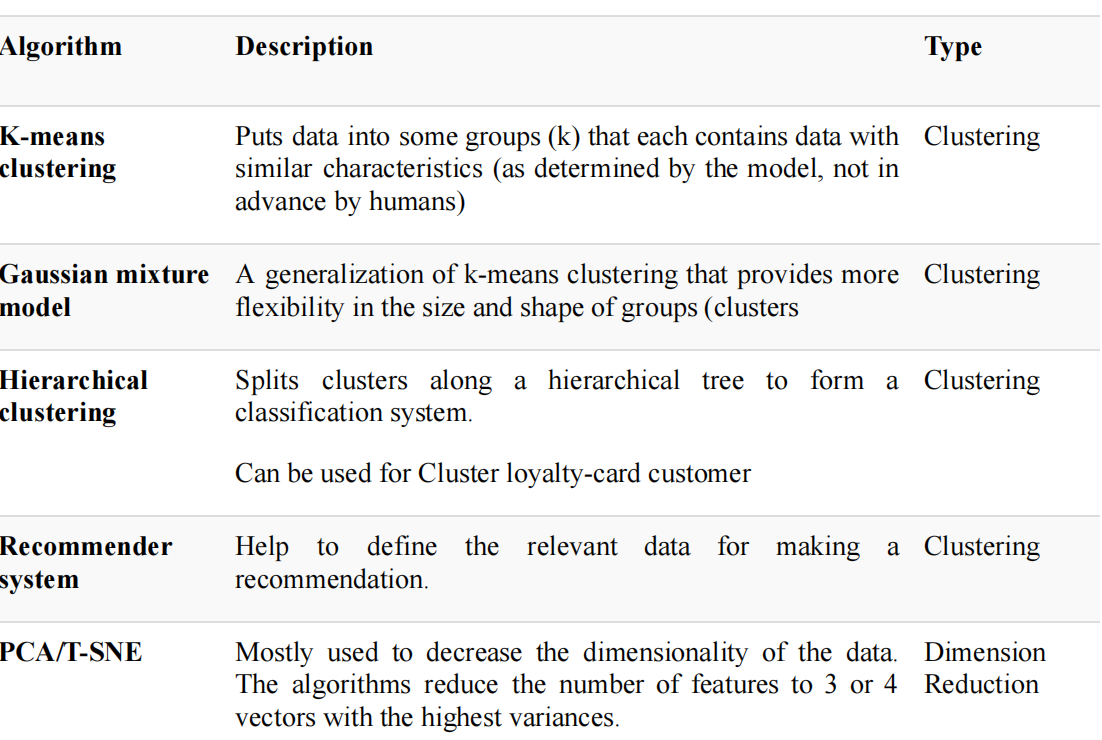
The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

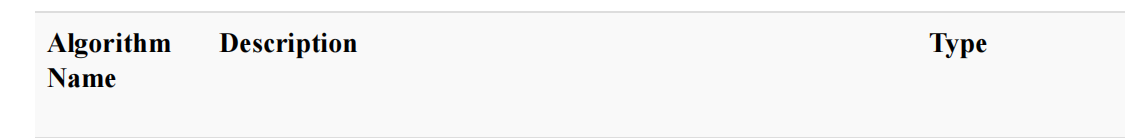
1. **Regression**

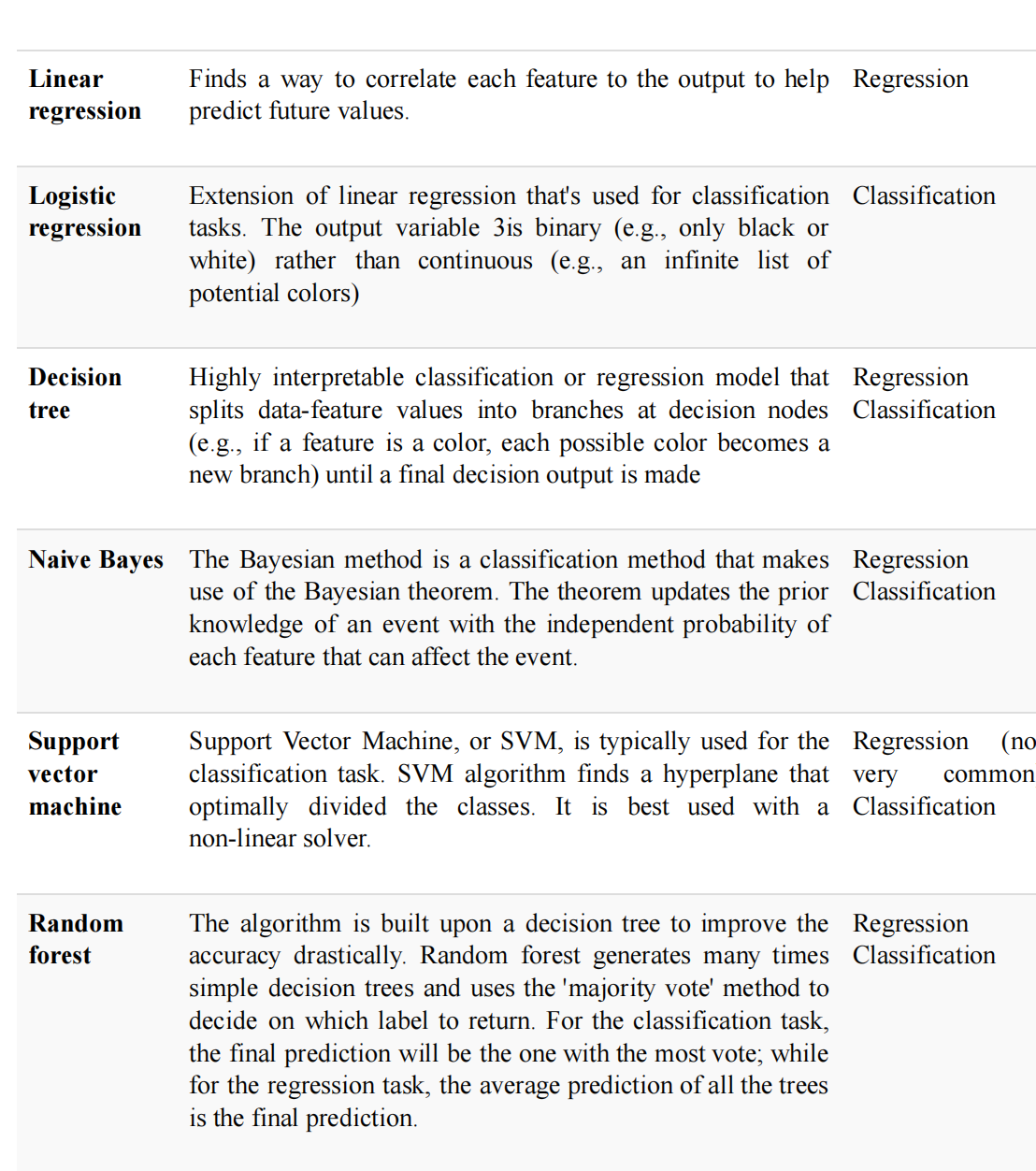
When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

**Unsupervised learning**

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you







**APPLICATION OF MACHINE LEARNING**

**Augmentation**:

● Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

● Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

● Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

● The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

● Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

● Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network. Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear. For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time. In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

## 6.2 DEEP LEARNING

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot. One of the most famous algorithms are:

● Q-learning

● Deep Q network

● State-Action-Reward-State-Action (SARSA)

● Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:** The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants. Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

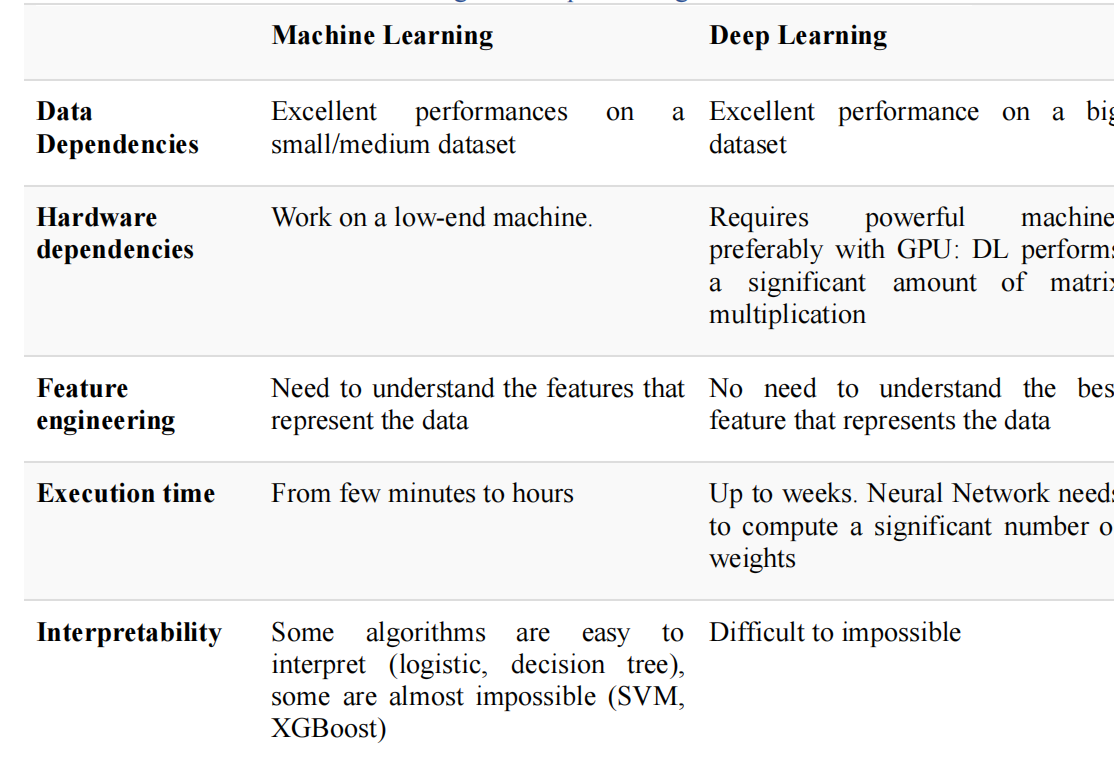
**AI in HR:** Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour 's ability to have their retail stores fully staffed, ramped and ready to operate. At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:** AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers. For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue. Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

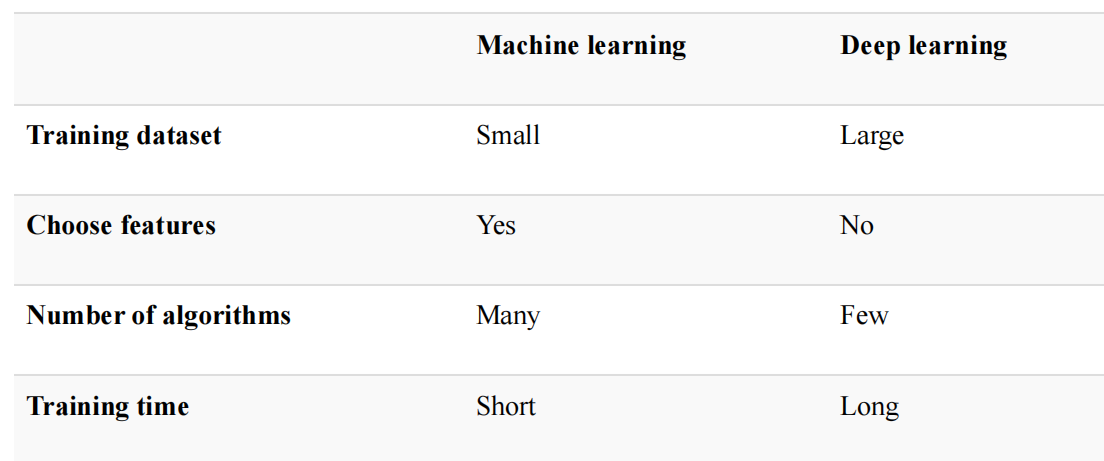
**Machine Learning**

Difference between Machine Learning and Deep Learning

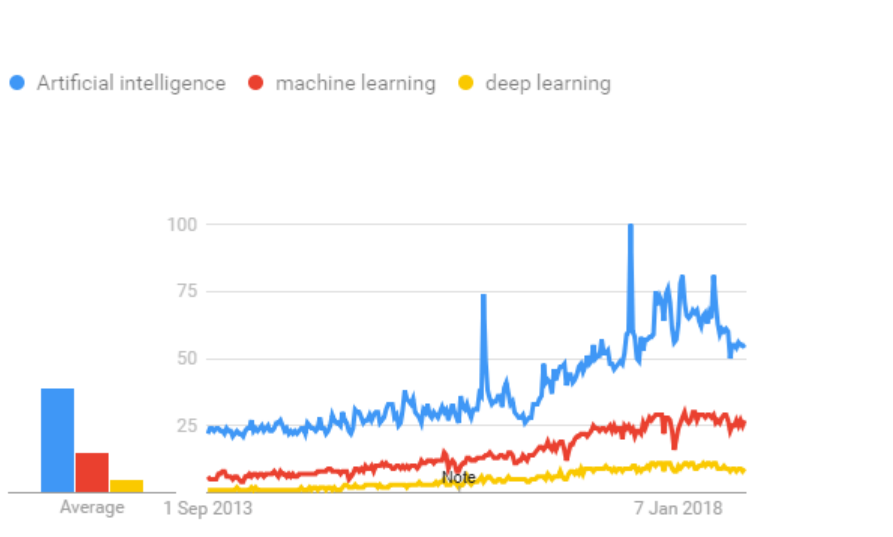


When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning



With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.



**FIG:6.2.1 GRAPH**

## 6.3 TENSOR FLOW

The most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word. Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

● Researchers

● Data scientists

● Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency. Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research. It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java. In this tutorial, you will learn

**TensorFlow Architecture**

Tensor flow architecture works in three parts:

● Pre processing the data

● Build the model

● Train and estimate the model

It is called Tensor flow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can Tensor flow run?**

TensorFlow can hardware, and software requirements can be classified into Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop. Run Phase or Inference Phase: Once training is done Tensor flow can be run on many different platforms. You can run it on

● Desktop running Windows, macOS or Linux

● Cloud as a web service

● Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model. The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python. Finally, a significant feature of Tensor Flow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

● Linear regression: tf. estimator .Linear Regressor

● Classification :tf. Estimator .Linear Classifier

● Deep learning classification: tf. estimator. DNN Classifier

● Booster tree regression: tf.estimator.BoostedTreesRegressor

● Boosted tree classification: tf.estimator.BoostedTreesClassifier.

## 6.4 PYTHON OVERVIEW

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, Unix shell, and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Features**

Python's features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:** Python's source code is fairly easy-to-maintain
* **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases:** Python provides interfaces to all major commercial databases.
* **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable :** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

**Python’s standard library**

* Pandas
* Numpy
* Sklearn
* Seaborn
* Matplotlib
* Importing Datasets

**PANDAS :**

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of

the most preferred and widely used tools in data munging/wrangling if not THE most used

one. Pandas is an open source What’s cool about Pandas is that it takes data (like a CSV or

TSV file, or a SQL database) and creates a Python object with rows and columns called data

frame that looks very similar to table in a statistical software (think Excel or SPSS for

example. People who are familiar with R would see similarities to R too). This is so much

easier to work with in comparison to working with lists and/or dictionaries through for loops

or list comprehension.

**Installation and Getting Started**

In order to “get” Pandas you would need to install it. You would also need to have Python

2.7 and above as a pre-requirement for installation. It is also dependent on other libraries

(like NumPy) and has optional dependancies (like Matplotlib for plotting). Therefore, I think

that the easiest way to get Pandas set up is to install it through a package like the Anaconda

distribution , “a cross platform distribution for data analysis and scientific computing.”

In order to use Pandas in your Python IDE (Integrated Development Environment)

like Jupyter Notebook or Spyder (both of them come with Anaconda by default), you need

to import the Pandas library first. Importing a library means loading it into the memory and

then it’s there for you to work with. In order to import Pandas all you have to do is run the

following code:

● **import pandas as pd**

● **import numpy as np**

Usually you would add the second part (‘as pd’) so you can access Pandas with

‘pd.command’ instead of needing to write ‘pandas.command’ every time you need to use it.

Also, you would import numpy as well, because it is very useful library for scientific

computing with Python. Now Pandas is ready for use! Remember, you would need to do it

every time you start a new Jupyter Notebook, Spyder file etc.

**Working with Pandas :**

Loading and Saving Data with Pandas

When you want to use Pandas for data analysis, you’ll usually use it in one of three different

ways:

* Convert a Python’s list, dictionary or Numpy array to a Pandas data frame.
* Open a local file using Pandas, usually a CSV file, but could also be a delimited text

file (like TSV), Excel, etc.

* Open a remote file or database like a CSV or a JSONon a website through a URL or

read from a SQL table/database.

* There are different commands to each of these options, but when you open a file, they would look like this:
* **pd.read\_filetype()**

As I mentioned before, there are different filetypes Pandas can work with, so you would

replace “filetype” with the actual, well, filetype (like CSV). You would give the path,

filename etc inside the parenthesis. Inside the parenthesis you can also pass different

arguments that relate to how to open the file. There are numerous arguments and in order to

know all you them, you would have to read the documentation (for example,

the documentation for pd.read\_csv() would contain all the arguments you can pass in this

Pandas command). In order to convert a certain Python object (dictionary, lists etc) the basic command is :

* **pd.DataFrame()**

Inside the parenthesis you would specify the object(s) you’re creating the data frame from.

This command also has different arguments.

You can also save a data frame you’re working with/on to different kinds of files (like CSV,

Excel, JSON and SQL tables). The general code for that is:

**● df.to\_filetype(filename)**

**Viewing and Inspecting Data**

Now that you’ve loaded your data, it’s time to take a look. How does the data frame look?

Running the name of the data frame would give you the entire table, but you can also get the

first n rows with df.head(n) or the last n rows with df.tail(n). df.shape would give you the

number of rows and columns. df.info() would give you the index, datatype and memory

information. The command s.value\_counts(dropna=False) would allow you to view unique

values and counts for a series (like a column or a few columns). A very useful command

is df.describe() which inputs summary statistics for numerical columns. It is also possible to

get statistics on the entire data frame or a series (a column etc):

● df.mean() : Returns the mean of all columns

● df.corr() : Returns the correlation between columns in a data frame

● df.count() : Returns the number of non-null values in each data frame column

● df.max() : Returns the highest value in each column

● df.min() : Returns the lowest value in each column

● df.median() : Returns the median of each column

● df.std() : Returns the standard deviation of each column

**Selection of Data**

One of the things that is so much easier in Pandas is selecting the data you want in

comparison to selecting a value from a list or a dictionary. You can select a column (df[col])

and return column with label col as Series or a few columns (df[[col1, col2]]) and returns

columns as a new DataFrame. You can select by position (s.iloc[0]), or by index

(s.loc['index\_one']) . In order to select the first row you can use df.iloc[0,:] and in order to

select the first element of the first column you would run df.iloc[0,0] . These can also be

used in different combinations, so I hope it gives you an idea of the different selection and

indexing you can perform in Pandas.

**Filter, Sort and Group by**

You can use different conditions to filter columns. For example, df[df[year] > 1984] would

give you only the column year is greater than 1984. You can use & (and) or | (or) to add

different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order

using df.sort\_values(col1) ; and also in a descending order

using df.sort\_values(col2,ascending=False). Furthermore, it’s possible to sort values

by col1 in ascending order then col2 in descending order by

using df.sort\_values([col1,col2],ascending=[True,False]).

The last command in this section is groupby. It involves splitting the data into groups based

on some criteria, applying a function to each group independently and combining the results

into a data structure. df.groupby(col) returns a groupby object for values from one column

while df.groupby([col1,col2]) returns a groupby object for values from multiple columns.

**Data Cleaning**

Data cleaning is a very important step in data analysis. For example, we always check for

missing values in the data by running pd.is null() which checks for null Values, and returns

a boolean array (an array of true for missing values and false for non-missing values). In

order to get a sum of null/missing values, run pd. Is null().sum(). Pd .not null() is the

opposite of pd. Is null(). After you get a list of missing values you can get rid of them, or

drop them by using df. Drop na() to drop the rows or df. drop na(axis=1) to drop the

columns. A different approach would be to fill the missing values with other values by

using df. Fill na(x) which fills the missing values with x (you can put there whatever you

want) or s .fill na(s.mean()) to replace all null values with the mean (mean can be replaced

with almost any function from the statistics section).

It is sometimes necessary to replace values with different values. For example, s.

replace(1,'one') would replace all values equal to 1 with 'one'. It’s possible to do it for

multiple values: s. replace([1,3],['one', 'three'])would replace all 1

with 'one' and 3 with 'three'. You can also rename specific columns by running: df.

rename(columns={'old\_name': 'new\_ name'})or use df. set\_ index('column\_one') to change

the index of the data frame.

**Join/Combine**

The last set of basic Pandas commands are for joining or combining data frames or

rows/columns. The three commands are:

● df1.append(df2)— add the rows in df1 to the end of df2 (columns should be identical)

● df. concat([df1, df2],axis=1) — add the columns in df1 to the end of df2 (rows should

be identical)

● df1.join(df2,on=col1,how='inner') — SQL-style join the columns in df1with the

columns on df2 where the rows for col have identical values. how can be equal to one

of: 'left', 'right', 'outer', 'inner'

**NUMPY**

Numpy is one such powerful library for array processing along with a large collection of

high-level mathematical functions to operate on these arrays. These functions fall into

categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc.

Getting NumPy

NumPy’s main object is a homogeneous multidimensional array. Unlike python’s array class

which only handles one-dimensional array, NumPy’s nd array class can handle

multidimensional array and provides more functionality. NumPy’s dimensions are known as

axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns.

Sometimes dimension is also known as a rank of that particular array or matrix.

Importing NumPy

NumPy is imported using the following command. Note here np is the convention followed

for the alias so that we don't need to write numpy every time.

**● import numpy as np**

NumPy is the basic library for scientific computations in Python and this article illustrates

some of its most frequently used functions. Understanding NumPy is the first major step in

the journey of machine learning and deep learning.

**Sk learn**

In python, scikit-learn library has a pre-built functionality under sk learn. Pre processing.

Next thing is to do feature extraction Feature extraction is an attribute reduction process.

Unlike feature selection, which ranks the existing attributes according to their predictive

significance, feature extraction actually transforms the attributes. The transformed attributes,

or features, are linear combinations of the original attributes. Finally our models are trained

using Classifier algorithm.. We use nltk . classify module on Natural Language Toolkit

library on Python. We use the labelled dataset gathered . The rest of our labelled data will

be used to evaluate the models. Some machine learning algorithms were used to classify pre

processed data. The chosen classifiers were Decision tree , Support Vector Machines and

Random forest. These algorithms are very popular in text classification tasks.

**SEABORN**

Data Visualization in Python

Data visualization is the discipline of trying to understand data by placing it in a visual

context, so that patterns, trends and correlations that might not otherwise be detected can

be exposed.

Python offers multiple great graphing libraries that come packed with lots of different

features. No matter if you want to create interactive, live or highly customized plots python

has a excellent library for you.

**To get a little overview here are a few popular plotting libraries:**

* **Matplotlib:** low level, provides lots of freedom
* **Pandas Visualization:** easy to use interface, built on Matplotlib
* **Seaborn:** high-level interface, great default styles.
* **ggplot:** based on R’s ggplot2, uses *Grammar of Graphics*
* **Plotly:** can create interactive plots

In this article, we will learn how to create basic plots using Matplotlib, Pandas visualization

and Seaborn as well as how to use some specific features of each library. This article will

focus on the syntax and not on interpreting the graphs.

Matplotlib

Matplotlib is the most popular python plotting library. It is a low level library with a Matlab

like interface which offers lots of freedom at the cost of having to write more code.

1. To install Matplotlib pip anaconda can be used.

2. pip install matplotlib

3. conda install matplotlib

Matplotlib is specifically good for creating basic graphs like line charts, bar charts,

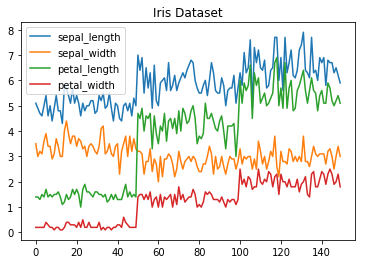
histograms and many more. It can be imported by typing : **import matplotlib.pyplot as plt**

**Line Chart**

In Matplotlib we can create a line chart by calling the plot method. We can also plot multiple

columns in one graph, by looping through the columns we want, and plotting each column

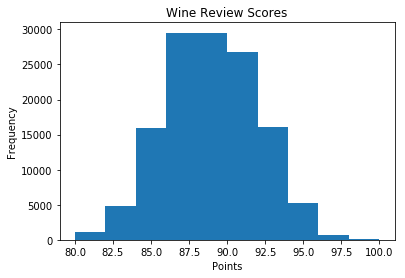
on the same axis.



**6.4.1 LINE CHART**

**Histogram**

50In Matplotlib we can create a Histogram using the hist method. If we pass it categorical data like the points column from the wine-review dataset it will automatically calculate how often each class occurs.



**6.4.2 HISTOGRAM**

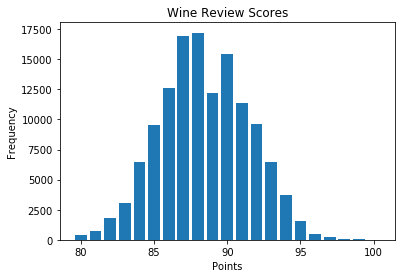
**Bar Chart**

A bar-chart can be created using the bar method. The bar-chart isn’t automatically

calculating the frequency of a category so we are going to use pandas value counts function

to do this. The bar-chart is useful for categorical data that doesn’t have a lot of different

categories (less than 30) because else it can get quite messy.



**6.4.3 BAR CHART**

Pandas Visualization

Pandas is a open source high-performance, easy-to-use library providing data structures,

such as data frames, and data analysis tools like the visualization tools we will use in this

article.

Pandas Visualization makes it really easy to create plots out of a pandas data frame and series.

It also has a higher level API than Matplotlib and therefore we need less code for the same

results.

1. **Pandas can be installed using either pip or conda.**
2. **pip install pandas.**
3. **conda install pandas**

**Heatmap**

A Heatmap is a graphical representation of data where the individual values contained in

a matrix are represented as colors. Heatmaps are perfect for exploring the correlation of

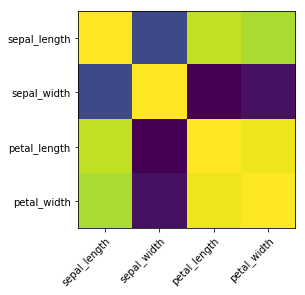
features in a dataset.

To get the correlation of the features inside a dataset we can call <dataset>.corr() , which is

a Pandas dataframe method. This will give use the correlation matrix.

We can now use either Matplotlib or Seaborn to create the heatmap.

**Matplotlib:**



**FIG 6.3.4 Heatmap without annotations**

Data visualization is the discipline of trying to understand data by placing it in a visual

context, so that patterns, trends and correlations that might not otherwise be detected can be

exposed.

Python offers multiple great graphing libraries that come packed with lots of different

features. In this article we looked at Matplotlib, Pandas visualization and Seaborn.

**TEST CASES**

**TEST CASE1:**

INPUT: Anaconda Navigator

OUTPUT: Jupyter Notebook and Browser

**TEST CASE2:**

INPUT: PYTHON PACKAGES IMPORT (Pandas, Numpy, Scikit, Matplot, Seaborn)

OUTPUT: CHECKING OF MODULE

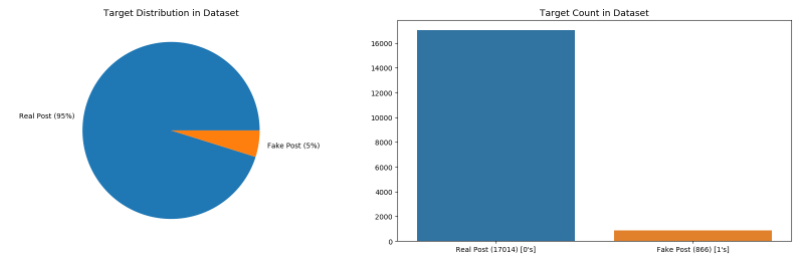
**TEST CASE3:**

INPUT: EXECUTION OF CODE

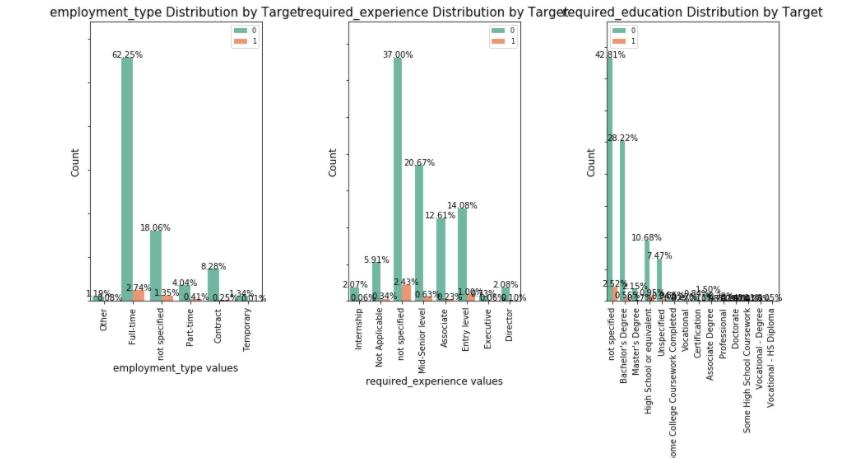
OUTPUT: GRAPH AND ACCURACY

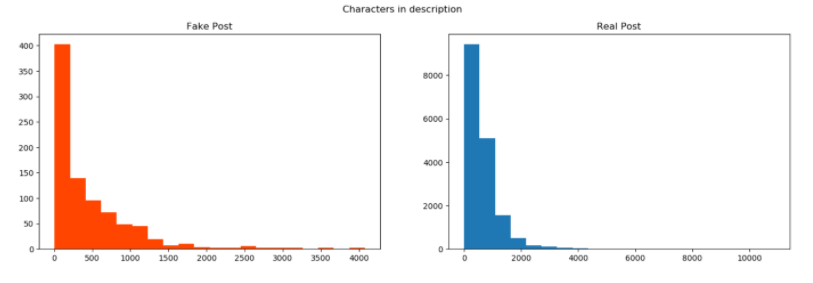
**Exploratory Data Analysis (EDA)**

Target count in dataset

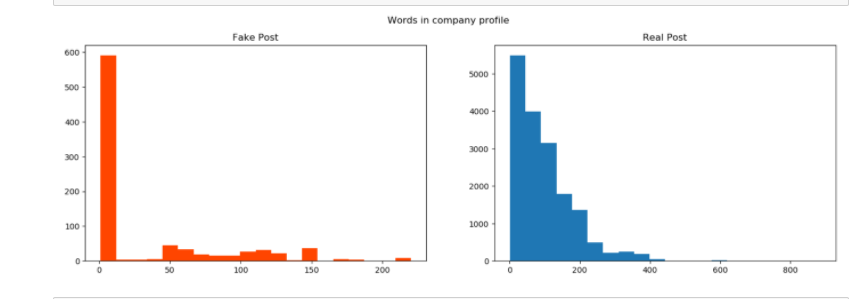


**FIG 6.3.5 EXPLOATARY DATA ANALYSIS**





**FIG 6.3.6 CHARACTERS IN DESCRIPTION**



**FIG 6.3.7 WORDS IN COMPANY PROFILE**

**6.5 ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage anaconda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

**Why use Navigator?**

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions. The command line program anaconda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly. Navigator is an easy, point-and-click way to work with packages and environments without needing to type anaconda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

**WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR** ?

The following applications are available by default in Navigator:

● Jupyter Lab

● Jupyter Notebook

● QT Console

● Spyder

● VS Code

● Glue viz

● Orange 3 App

● Rodeo

● RStudio

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code. You can also use Jupyter Notebooks the same way, Jupyter Notebooks are an interfaces into a single notebook file that is edited, viewed and used in a web browser.

**What’s new in 1.9?**

● Add support for **Offline Mode** for all environment related actions.

● Add support for custom configuration of main windows links.

● Numerous bug fixes and performance enhancements.

## 6.6 SOURCE CODE

import warnings

warnings.filterwarnings('ignore')

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

data=pd.read\_csv('insurance.csv')

data.head()

data.info()

data.shape

data.columns

from sklearn.preprocessing import LabelEncoder

lab=LabelEncoder()

data['sex']=lab.fit\_transform(data['sex'])

data['smoker']=lab.fit\_transform(data['smoker'])

data['region']=lab.fit\_transform(data['region'])

data.head()

sns.countplot(x='smoker',data=data)

sns.countplot(x='sex',data=data)

x=data.iloc[:,data.columns!='charges']

y=data.iloc[:,data.columns=='charges']

#x.shape

#x.head()

x.head()

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score,confusion\_matrix,recall\_score, roc\_curve, auc

xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.3)

xtrain.head()

ytrain.head()

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 300, random\_state = 0) #N-estimators means how many time predict the value like that

regressor.fit(xtrain, ytrain)

y\_pred = regressor.predict(xtest)

#y\_pred

from sklearn import metrics

from sklearn.metrics import r2\_score

print("\n\nr2\_score is " , r2\_score(y\_pred,ytest))

from sklearn.linear\_model import LinearRegression

alg = LinearRegression()

alg.fit(xtrain, ytrain)

y\_predict = alg.predict(xtest)

print("\n\nr2\_score is " , r2\_score(y\_predict,ytest))

from sklearn import tree

dt=tree.DecisionTreeRegressor()

dt.fit(xtrain, ytrain)

x\_predicted=dt.predict(xtest)

print("\n\nr2\_score is " , r2\_score(x\_predicted,ytest))

from sklearn.svm import SVR

regressor = SVR()

regressor.fit(xtrain, ytrain)

predicted=regressor.predict(xtest)

print("\n\nr2\_score is " , r2\_score(predicted,ytest))

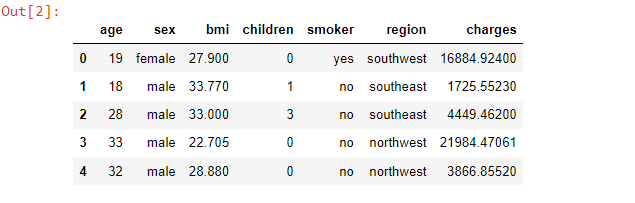
test\_vector = np.reshape(np.asarray([19,0,27.900,0,1,3]),(1,6))

p = int(regressor.predict(test\_vector)[0])

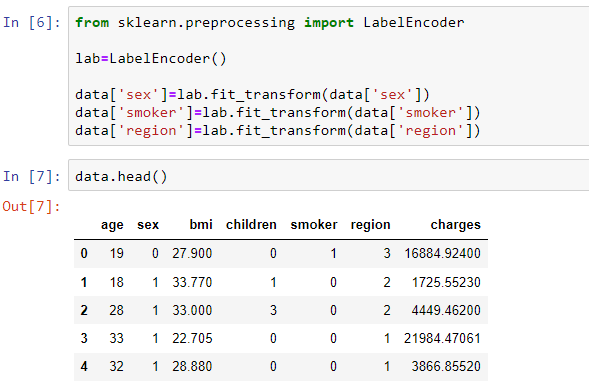
**RESULTS:**

Data mining is a process to extract knowledge from existing data. It is used as a tool in banking and finance, in general, to discover useful information from the operational and historical data to enable better decision-making. It is an interdisciplinary field, the confluence of Statistics, Database technology, Information science, Machine learning, and Visualization. It involves steps that include data selection, data integration, data transformation, data mining, pattern evaluation, knowledge presentation.

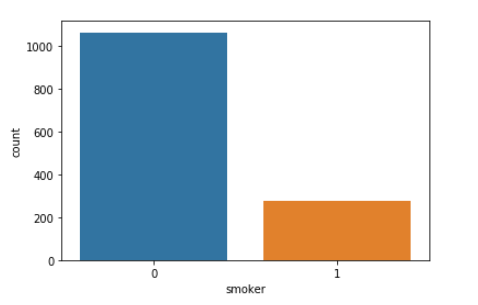
**Dataset**

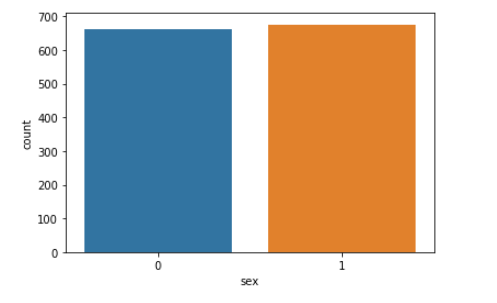


**Data Pre processing**

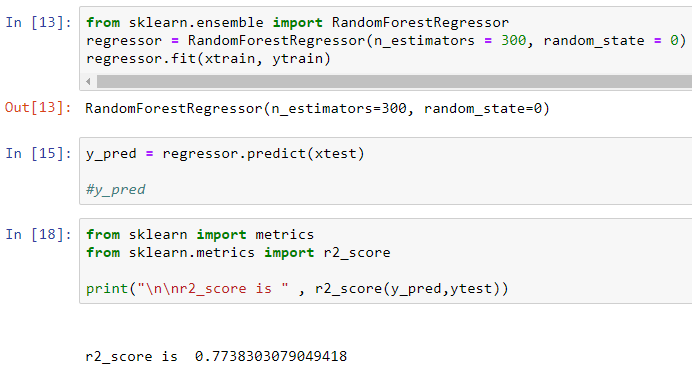


**Data Exploration**

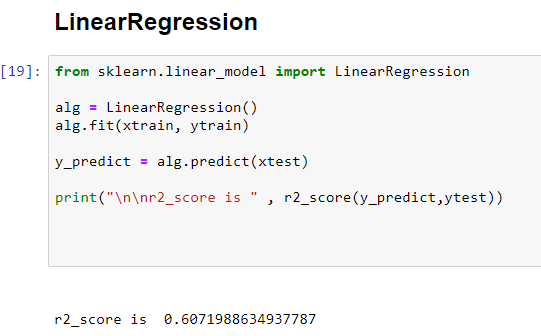




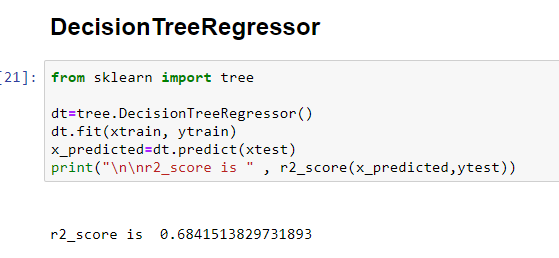
**RF Model Result**



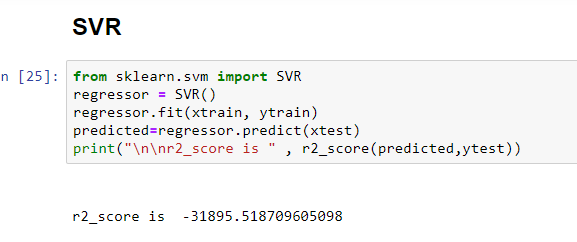
**LR Model Result**



**DTree Model Result**



**SVM Model Result**



# CHAPTER 7

# TESTING

## 7.1 TESTING METHODS

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 TESTING

### 7.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### 7.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 7.1.3 FUNCTIONAL TESTING

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

Functional testing is centered on the following items:

● Functions: Identified functions must be exercised.

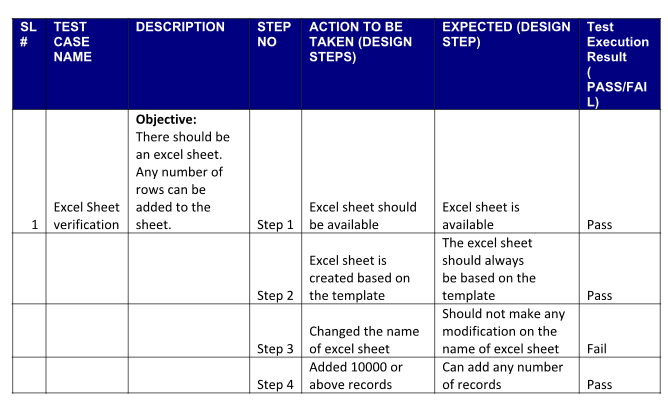
● Output: Identified classes of software outputs must be exercised.

● Systems/Procedures: system should work properly .

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

**Test Case 1 :**



# CHAPTER 8

# CONCLUSION

### CONCLUSION

In this project, we proposed a machine learning model for predicting medical costs.. We applied four regression techniques Linear Regression, Support Vector Regression, Decision Tree Regression, Random Forest Regression. We also applied RF model and observed that age, BMI are features which decides the dependent variable. Out of all experiments, Random Forest model given better result.

# 

# CHAPTER 9

# REFERENCES

### 9.1 REFERENCES

* B. Nithya, Dr. V. Ilango, “Predictive Analytics in Health Care Using Machine

Learning Tools and Techniques”, International Conference on Intelligent

Computing and Control Systems ICICCS 2017, 978-1-5386-2745-7/17/$31.00

©2017 IEEE.

* A. Tike and S. Tavarageri.(2017). A Medical Price Prediction System using

Hierarchical Decision Trees. In: IEEE Big Data Conference 2017. IEEE,

978-1-5386- 2715-0/17/$31.00 ©2017 IEEE.

* Lahiri and N. Agarwal, “Predicting healthcare expenditure increase for an

individualfrom medicare data,” in Proceedings of the ACM SIGKDD Workshop on

Health Informatics, 2014.

* Gregori, M. Petrinco, S. Bo, A. Desideri, F. Merletti, and E. Pagano,

“Regression modelsfor analyzing costs and their determinants in health care: an

introductory review,” International Journal for Quality in Health Care, vol. 23, no.

3, pp. 331–341, 2011.

* Bertsimas, M. V. Bjarnad´ottir, M. A. Kane, J. C. Kryder, R. Pandey, S.

Vempala, and G.Wang, “Algorithmic prediction of health-care costs,” Operations

Research, vol. 56, no. 6, pp. 1382–1392, 2008.

* A. Tike and S. Tavarageri. (2017). A Medical Price Prediction System using

Hierarchical Decision Trees. In: IEEE Big Data Conference 2017. IEEE.

* “National Health Expenditures 2015 Highlights,” CMS.gov, 2015. [Online].

Available:https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Tr

endsand-Reports/NationalHealthExpendData/downloads/highlights.pdf.

* J. Cubanski, C. Swoop and T. Neuman,“How Much Is Enough? Out-of-Pocket

Spending Among Medicare Beneficiaries: A Chartbook,” The Henry J. Kaiser Family

Foundation, Menlo Park, CA, 2014. [Online]. Available: http://files.kff.org/attachment/ [how-much-isenough-out-of-pocket-spending-among-medicare-beneficiaries-a-char tbook- . J. Cubanski and T. Neuman,“The Facts on Medicare Spending and Financing,”

* The Henry J. Kaiser Family Foundation, Menlo Park, CA, 2017. [Online].