

Real Estate Web Application using Flask

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

Mohit Anil Kumar (38110323)
Piyush Kumar Singh (38110408)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY – 2022

BONAFIDE CERTIFICATE

Th



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that this Project Report is the bonafide work of **Mohit Anil Kumar(38110323)** , **PIYSUH KUMAR SINGH(38110408)** who carried out the project entitled "**Real Estate Web Application using Flask**" under my supervision from NOV 2021 to MAY 2022.

Internal Guide

Dr.D.USHA NANDINI

Head of Department

Dr. S.VIGNESHWARI, M.E., Ph.D.,

Dr. Lakshmanan L, M.E, Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I MOHIT ANIL KUMAR (38110323) hereby declare that the Project Report entitled **"Real Estate Web Application using Flask "** done by me and **PIYUSH KUMAR SINGH(38110408)** under the guidance of **Dr.D. USHA NANDINI** is submitted in partial fulfillment of the requirements for the award of **Bachelor of Engineering degree in Computer Science and Engineering.**

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing** **Dr.L.Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.D.USHA NANDINI** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Machine Learning is a study of computer algorithm to improve automatically through experience and by use of data. It has been as a part of artificial Intelligence .Machine learning algorithms build a model based on sample data, known as “Training data”, in order to make predictions or decisions without being explicitly programmed to do so .Machine learning algorithm are used in wide variety of applications, such as medicine, email filtering and computer vision where it is difficult or unfeasible to develop a conventional algorithms to perform needed task.A subset of machine learning is closely related to computational statistics , which is focus on making predictions.

The Housing Price Prediction Using Concept of Machine Learning Has Become one of the most interesting application of Machine Learning Concept. To predict The Price of the House using Linear Regression Algorithm of Machine Learning . Regression is A Machine Learning tool That helps to make prediction by learning from the existing statistical data and this is done by finding relationship between Target Parameter and the set of Other Parameters. According to this definition , a House’s price depends on parameters such as Number of Bedrooms, Living Area , Location etc . If we apply Artificial Learning to these Parameters we can Calculate House Valuations In a given Geographical area.

Jupyter python programming software is used to design the code for Predicting the Housing Prices. It is an open source Software and it provides services for interactive computing across dozen of programming Languages .The software provides various functions and tools and In-Build Python Library so that operations can be performed accurately with maximum efficiency. with loops, functions variables, operators to perform various operations and obtain data. These software are very precise thus used in various research and analytical and educational related works.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	i
	LIST OF FIGURES	v
	LIST OF TABLES	vii
	ABBREVRATIONS	viii
1.	INTRODUCTION	01
	1.1 TYPES OF ALGORITHM USED	06
	1.1.1 K-Nearest Neighbor classification	06
	1.1.2 Support Vector Machine algorithm	10
	1.1.3 Naïve Bayes algorithm	12
	1.1.4 Decision Tree algorithm	15
	1.1.5 Random Forest classification	15
	1.2 ORGANIZATION OF THE PROJECT	
	WORK	17
2.	LITERATURE SURVEY	23
3.	AIM AND SCOPE OF PRESENT	
	INVESTIGATION	24
	3.1 AIM	24
	3.2 SCOPE	24
	3.3 PROBLEM DEFINITION	25

3.4 RELATED WORKS	25
3.5 EXISTING SYSTEM	26
3.5.1 Disadvantages of existing system	26
3.6 OVERVIEW OF PROPOSED SYSTEM	27
3.6.1 Advantages of proposed system	27
4. METHODOLOGY	28
4.1 HARDWARE REQUIREMENTS	28
4.2 SOFTWARE REQUIREMENTS	28
4.2.1 Overview of jupyter notebook	28
4.3 SYSTEM DESIGN	31
4.4 SYSTEM ARCHITECTURE	31
4.5 MODULE	31
4.5.1 Data set collection	32
4.5.2 Pre-processing of data set	32
4.5.3 Extraction of data set	33
4.5.4 Prediction of results	33
4.6 ALGORITHM DESCRIPTION OF METHODS	33
4.7 PERFORMANCE MEASURES	40
4.7.1 HRFLM ALGORITHMS	40
4.7.2 Algorithm1 Decision tree-based partition	41
4.8 Algorithm2 Apply ml to find less error rate	41

5.	RESULTS AND DISCUSSION	42
	5.1 RESULTS AND DISCUSSIONS	42
	5.2 PERFORMANCE ANALYSIS	42
	5.3 BENCHMARKING OF THE PROPOSAL MODEL	43
6.	SUMMARY AND CONCLUSION	44
	6.1 CONCLUSION	44
	6.2 FUTURE ENHANCEMENT	44
	REFERENCES	45
	APPENDIX	46
	A. SAMPLE CODE	46
	B. SCREENSHOTS	51
	C. PUBLICATION WITH PLAGIARISM REPORT	53

LIST OF FIGURES

Fig. No.	Fig. Name	Page No.
Fig. no. 1.1	Linear Regression with two variables	2
Fig. no. 1.2	Linear Regression with Multiple variable	4
Fig. no. 1.3	Supervised Learning Architecture	5
Fig. no. 1.4	Different Clusters of data On graph	7
Fig. no. 1.5	Clustering of Red Circles using Blue Star	7
Fig. no. 1.6	Clustering Using Different values of K	8
Fig. no. 1.7	Graph of Error	9
Fig. no. 1.8	Graph of Validation Error	9
Fig. no. 1.9	Graph of Different Classes	10
Fig. no. 1.10	Graph of Segregating the classes	11
Fig. no. 1.11	Classification of different clusters	11
Fig. no. 1.12	Graph Of Different Classes	12
Fig. no. 1.13	Jupyter Open Source Software	18
Fig. no. 1.14	Browser View of Jupyter Notebook	20
Fig. no. 1.15	Sci – Kit Learn	21
Fig. no. 1.16	Supervised Learning Classification	21
Fig. no. 1.17	Supervised Learning Regression	21

Fig. no. 4.1	Browser View of Jupyter Notebook	29
Fig. no. 4.2	Architecture of Data Prediction	31
Fig. no. 4.3	Graph Of Classifier	37
Fig. no. 4.4	Overall error rate of Dataset	41
Fig. no. 5.1	Comparison Between Proposed and Existing Model	42
Fig. no. B.1	Scatter Matrix Of few Attributes	51
Fig. no. B.2	Positive Correlation Scatter plot	52
Fig. no. B.3	Negative Correlation Scatter plot	52

Table N0.	Table NAME	Page No.
Table 1.1	Linear Regression with two Variables	2
Table 1.2	Linear Regression with Multiple Variables	3
Table 1.3	Comparing Different ML Models	6
Table 1.4	Classification Of Fruits on Different Attributes	13
Table 3.1	Regression Analysis	39

ABBREVIATIONS

ABBREVIATION	EXPANSION
ML	-Machine Learning
SDK	-Software Development Kit
KNN	-K Nearest Neighbour
SVM	-Support Vector Machine
CAS	-Carotid Artery Stenting
HRFLM	-Hybrid Random Forest with Linear Model
IHDPS	-Intelligent Heart Disease Prediction System
MLP	-Multi Layer Perception
CNN	-Convolutional Neural Networks
CPU	-Central Processing Unit
RAM	-Random Access Memory
RF	-Random Forest
LM	-Linear Method

CHAPTER 1:

INTRODUCTION

Housing Price Prediction is commonly used to estimate the change in the Housing Price. Since Housing Price is Strongly Correlated to other Factors such as location , area , population ,it requires other information apart from HPI to predict Individual Prediction Prices. However to Predict the Housing Price our Machine Learning model requires data About more Number of Features so that prices can be Predicted more accurately , To do this we use the Machine Learning Algorithm called as Linear Regression . And using the Algorithm we will see effect of different features eg location , area, Furnishing which acts as Independent variables on The price of our house which is a Dependent Variable .

Linear Regression is one of the most Easiest algorithm of Machine learning . It is basically a statistical model that attempts to show the relationship between two variables with a linear equation. Linear Regression is a Supervised Learning Algorithm where the Predicted Output is Continous and has a constant slope . It's used to predict value within a continuous range (eg sales,price) rather than trying to classify them into catogeries (eg Black,Blue) .

Simple regression

Simple linear regression uses traditional slope-intercept form, where mm and bb are the variables our algorithm will try to “learn” to produce the most accurate predictions. xx represents our input data and yy represents our prediction.

$$y=mx+b$$

Let's say we are given a datasett with the following columns (features): how much a company spends on Radio advertising each year and its annual Sales in terms of units sold. We are trying to develop an equation that will let us to predict units sold based on how much a company spends on radio advertising. The rows (observations) represent companies.

TABLE 1.1 : Company Table for Regression with two variables

Company	Radio (\$)	Sales
Amazon	37.8	22.1
Google	39.3	10.4
Facebook	45.9	18.3
Apple	41.3	18.5

Our prediction function outputs an estimate of sales given a company's radio advertising spend and our current values for *Weight* and *Bias*.

$$\text{Sales} = \text{Weight} \cdot \text{Radio} + \text{Bias}$$

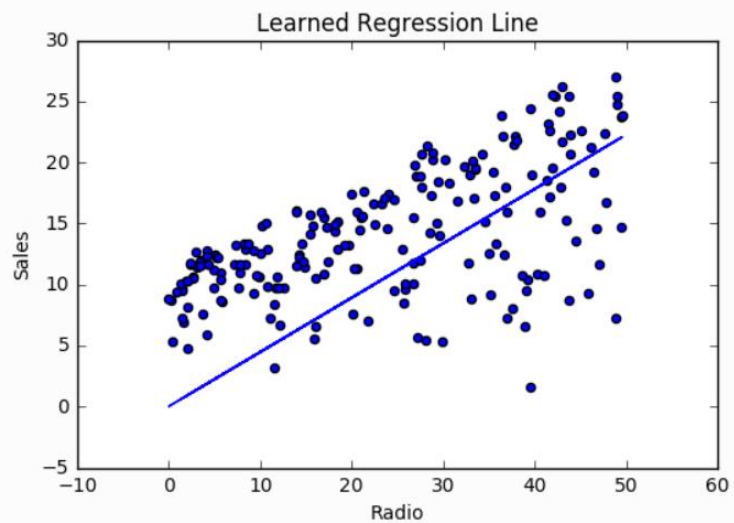


FIGURE 1.1: Linear Regression with Two Variables

Multivariable regression

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

$$f(x,y,z)=w_1x+w_2y+w_3z \quad f(x,y,z)=w_1x+w_2y+w_3z$$

The variables x,y,z represent the attributes, or distinct pieces of information, we have about each observation. For sales predictions, these attributes might include a company's advertising spend on radio, TV, and newspapers.

$$\text{Sales}=w_1\text{Radio}+w_2\text{TV}+w_3\text{News} \quad \text{Sales}=w_1\text{Radio}+w_2\text{TV}+w_3\text{N}$$

Let's say we are given data on TV, radio, and newspaper advertising spend for a list of companies, and our goal is to predict sales in terms of units sold.

TABLE 1.2 : Company Table for Regression with Multiple variables

Company	TV	Radio	News	Units
Amazon	230.1	37.8	69.1	22.1
Google	44.5	39.3	23.1	10.4
Facebook	17.2	45.9	34.7	18.3
Apple	151.5	41.3	13.2	18.5

Growing Complexity : As the Number of Features grows , the Complexity of our model increases and it becomes increasingly difficult to visualize , or even comprehend our data. One solution 1-2 features at a time. In this Example we explore how Radio and TV investment impact sales.

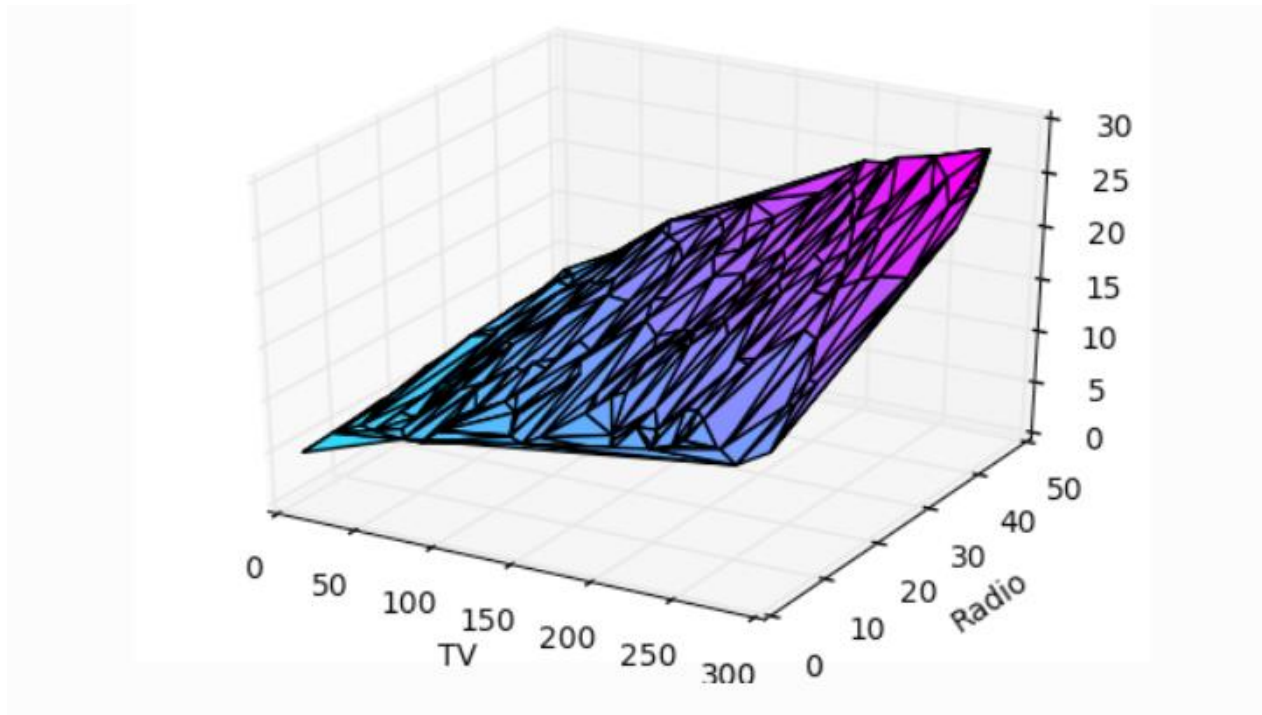


FIGURE 1.2: Linear Regression with Multiple Variables

Our predict function outputs an estimate of sales given our current weights (coefficients) and a company's TV, radio, and newspaper spend. Our model will try to identify weight values that most reduce our cost function.

$$\text{Sales} = W_1 \text{TV} + W_2 \text{Radio} + W_3 \text{Newspaper}$$

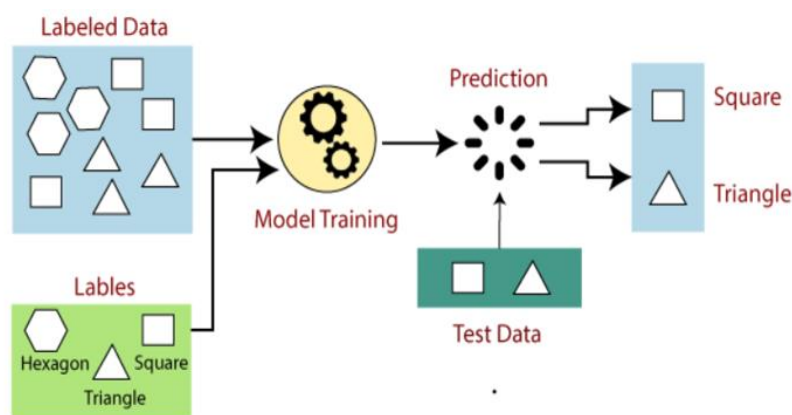
Normalization

As the number of features grows, calculating gradient takes longer to compute. We can speed this up by “normalizing” our input data to ensure all values are within the same range. This is especially important for datasets with high standard deviations or differences in the ranges of the attributes. Our goal now will be to normalize our features so they are all in the range -1 to 1.

HOW SUPERVISED LEARNING WORKS ?

In the supervised learning, models are trained using labelled dataset , where the model learns about each type of data. Once the Learning process is completed , the model is tested on the basis of test data (a subset of the training data) , and then it predicts the output.

FIGURE 1.3:Supervised Learning Architecture



The working of supervised learning can be easily understood by the above example diagram. Suppose we have a dataset of different types of shapes which includes squares, rectangle, triangle and polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four side , and all the sides are equal , then it will be labelled as a square .
- If the given shape has three sides, then it will be labelled as a triangle.
- If the given shape has all six equal sides then it will labelled as an hexagon. Now after training , we test our model using the test set , and the task of the model is to identify the shape. The Machine is already trained on all types of shapes, and when it finds a new shape , it classifies the shape on the bases of a number of sides, and predicts the output.

1.1 TYPES OF ALGORITHM USED

For determination of disease and for prediction purposes, there are five important types of algorithms are used. These algorithms have all the type of prediction accuracy and have all the precision control. These are:

- K-Nearest Neighbour (KNN) classification.
- Support Vector Machine (SVM) algorithm.
- Naïve Bayes algorithm.
- Decision Tree algorithm.
- Random Forest classification.

1.1.1 K-Nearest Neighbour (KNN) classification

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 2 important aspects:

1. Ease to interpret output
2. Calculation time

Let us take a few examples to place KNN in the scale:

	Logistic Regression	CART	Random Forest	KNN
1. Ease to interpret output	2	3	1	3
2. Calculation time	3	2	1	3
3. Predictive Power	2	2	3	2

Table 1.3: Comparing Different ML Models

KNN algorithm fares across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS):

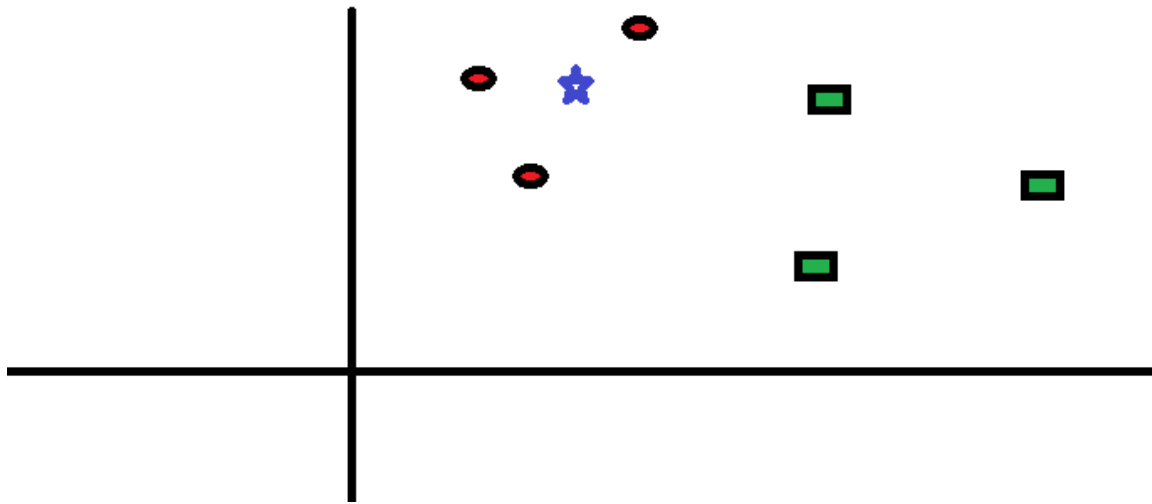


Fig1.4 Different clusters of data on graph

You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” in KNN algorithm is the nearest neighbors we wish to take vote from. Let’s say $K = 3$. Hence, we will now make a circle with BS as centre just as big as to enclose only three datapoints on the plane. Refer to following diagram for more details:

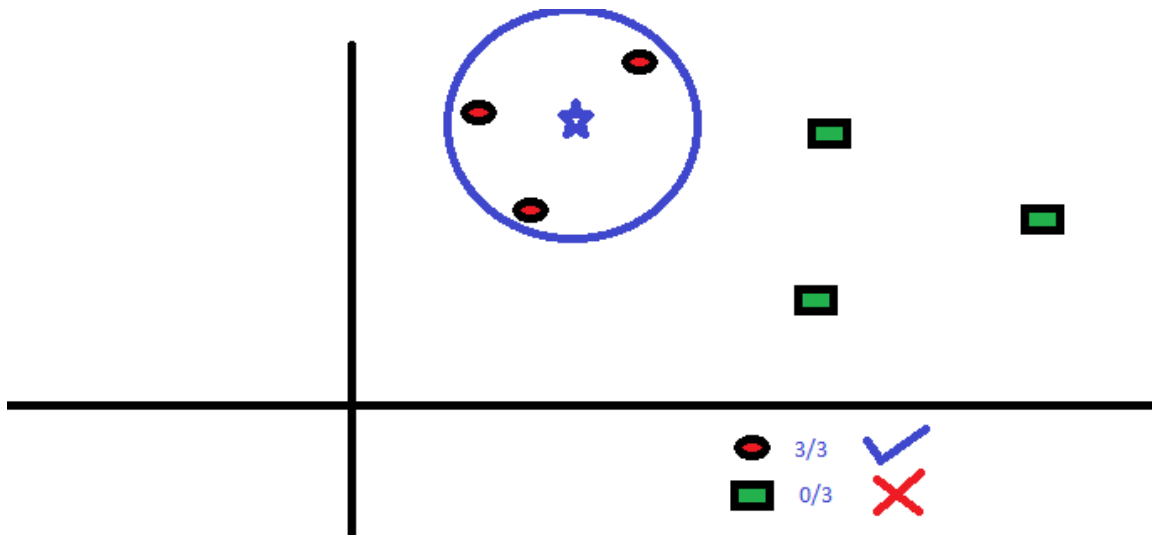


Fig1.5 Cluster of red circles using blue star

The three closest points to BS are all RC. Hence, with good confidence level we can say that the BS should belong to the class RC. Here, the choice became obvious as all three votes from the closest neighbour went to RC. The choice of the parameter K is

very crucial in this algorithm. Next we will understand what are the factors to be considered to conclude the best K .

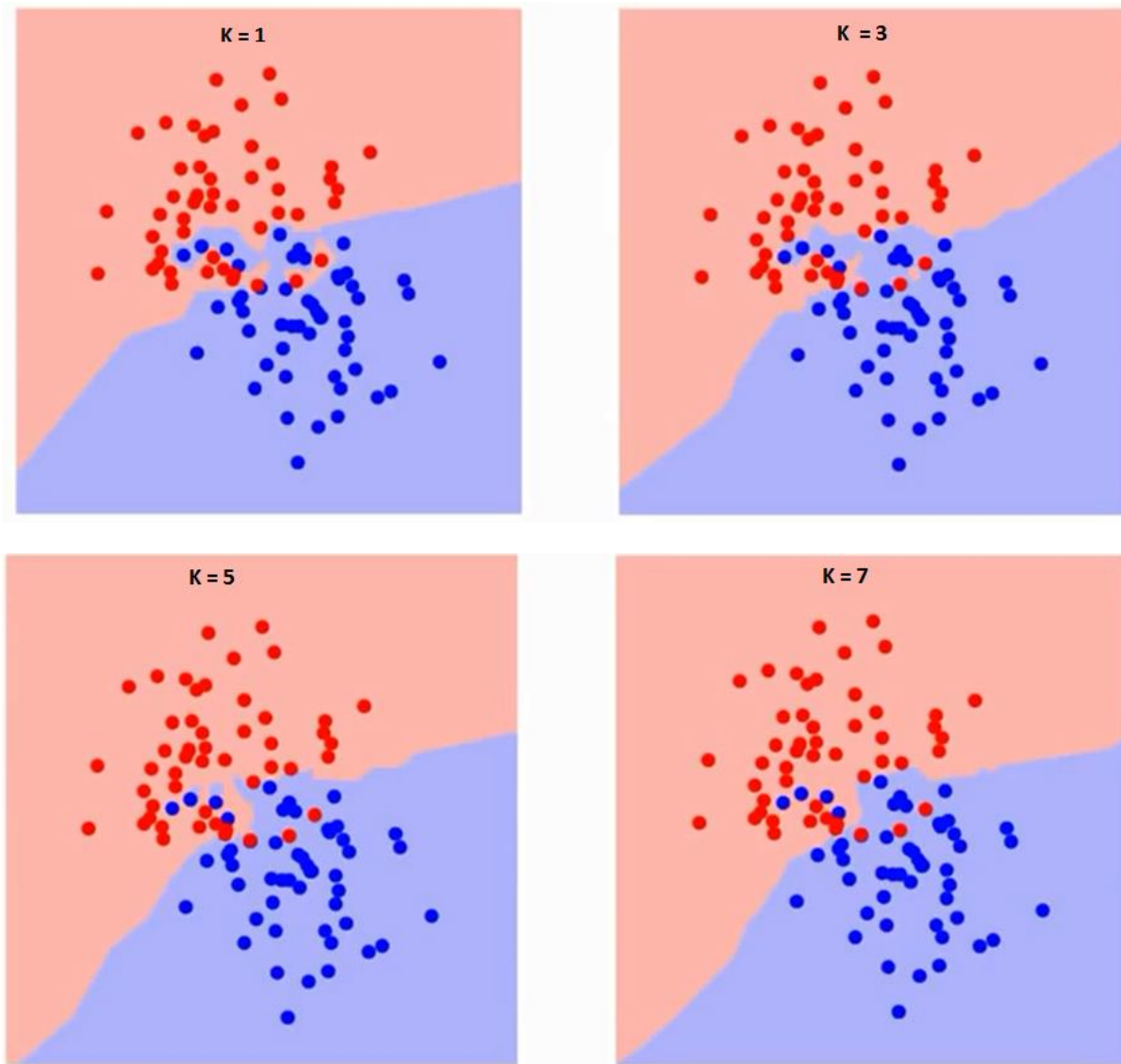


Fig1.6 Clustering using different values of K .

If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access on different K-value. Following is the curve for the training error rate with varying value of K :

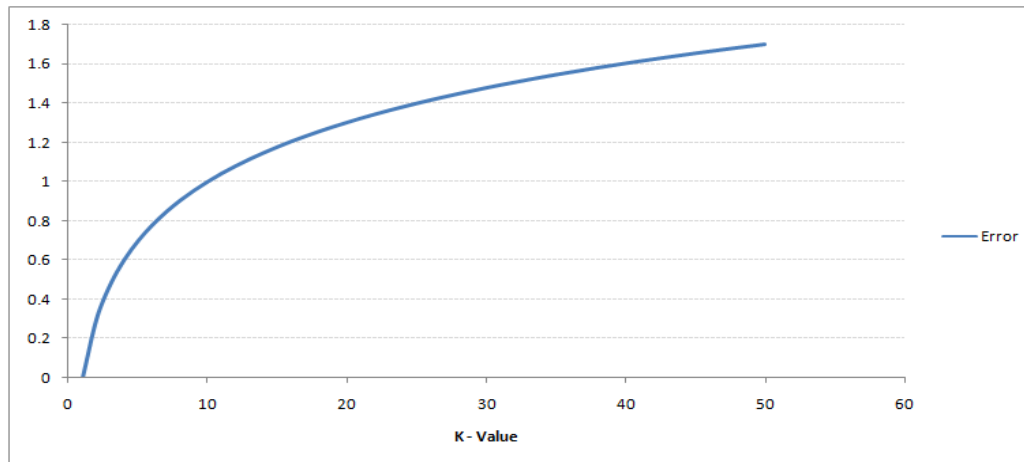


Fig1.7 Graph of error

As you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1. Following is the validation error curve with varying value of K:

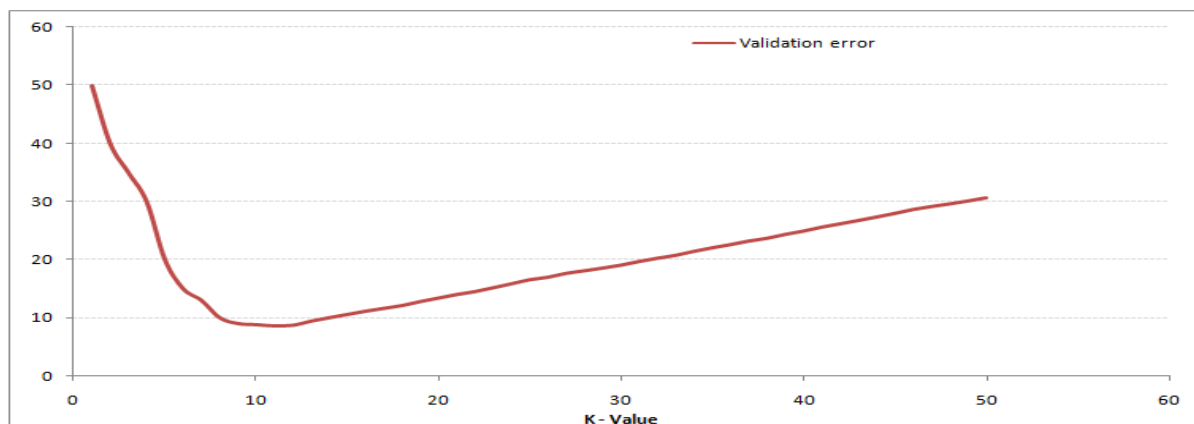


Fig1.8 Graph of validation error

This makes the story more clear. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training

and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

1.1.2 Support Vector Machine (SVM) algorithm

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

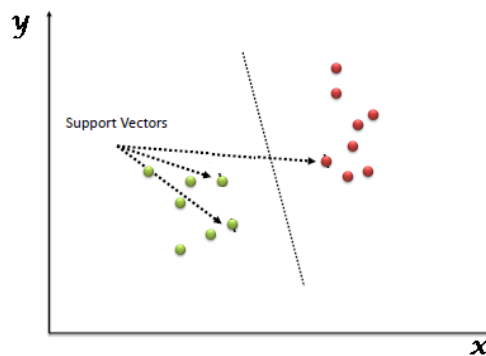


Fig1.9 Graph of different classes

There are certain points which must be followed and must be studied for understanding of support vector machine algorithm.

Identify the right hyper-plane (Scenario-1) : Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify stars and circles.

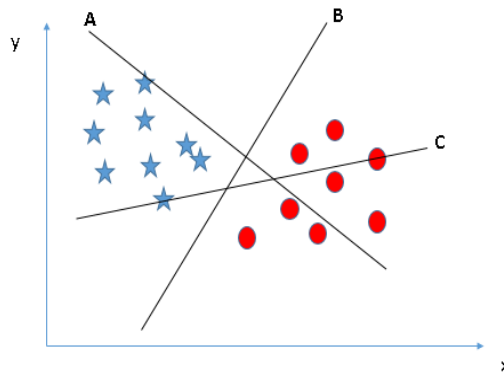


Fig1.10 Graph of segregating the classes.

- One need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.
- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

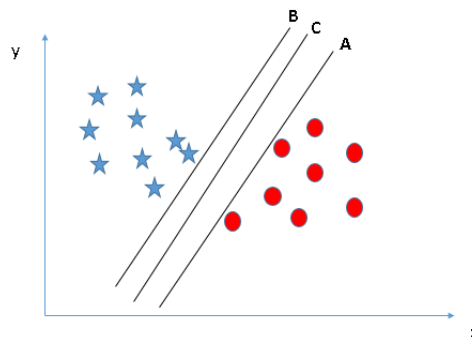


Fig1.11 Classification of different clusters

Here, maximizing the distances between nearest data point (either class) or hyper-plane will help us to decide the right hyper-plane. This distance is called as margin. Let’s look at the below snapshot:

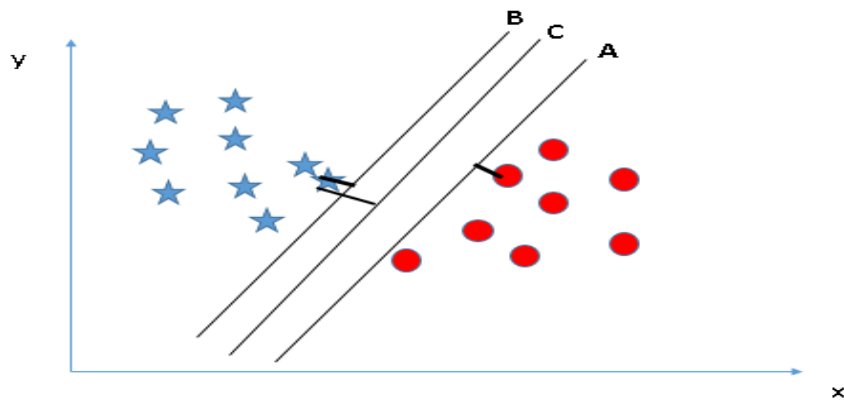


Fig1.12 Graph of different classes

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

1.1.3 Naïve Bayes algorithm

Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks. Typical applications include filtering spam, classifying documents, sentiment prediction etc. It is based on the works of Rev. Thomas Bayes (1702-61) and hence the name. Let us take an example and then study this algorithm. Say you have 1000 fruits which could be either 'banana', 'orange' or 'other'. These are the 3 possible classes of the Y variable. We have data for the following X variables, all of which are binary (1 or 0).

- Long
- Sweet
- Yellow

The first few rows of the training dataset look like this:

Table1.3 Classification of fruits

Fruit	Long (x1)	Sweet (x2)	Yellow (x3)
Orange	0	1	0
Banana	1	0	1
Other	1	1	1

For the sake of computing the probabilities, let's aggregate the training data to form a counts table like this.

Table1.4 Classification of fruits on different attributes

Type	Long	Not Long	Sweet	Not Sweet	Yellow	Not Yellow	Total
Banana	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

So the objective of the classifier is to predict if a given fruit is a 'Banana' or 'Orange' or 'Other' when only the 3 features (long, sweet and yellow) are known. Let's say you are given a fruit that is: Long, Sweet and Yellow, can you predict what fruit it is? This is the same of predicting the Y when only the X variables in testing data are known. Let's solve it by hand using Naive Bayes. The idea is to compute the 3 probabilities, that is the probability of the fruit being a banana, orange or other. Whichever fruit type gets the highest probability wins. All the information to calculate these probabilities is present in the above tabulation.

- Compute the 'Prior' probabilities for each of the class of fruits.

That is, the proportion of each fruit class out of all the fruits from the population. You can provide the 'Priors' from prior information about the population. Otherwise, it can be computed from the training data. For this case, let's compute from the training data. Out of 1000 records in training data, you have 500 Bananas, 300 Oranges and 200 Others. So, the respective priors are 0.5, 0.3 and 0.2.

$$P(Y=\text{Banana}) = 500 / 1000 = 0.50$$

$$P(Y=\text{Orange}) = 300 / 1000 = 0.30$$

$$P(Y=\text{Other}) = 200 / 1000 = 0.20$$

- Compute the probability of evidence that goes in the denominator.

This is nothing but the product of P of X s for all X . This is an optional step because the denominator is the same for all the classes and so will not affect the probabilities.

$$P(x_1=\text{Long}) = 500 / 1000 = 0.50$$

$$P(x_2=\text{Sweet}) = 650 / 1000 = 0.65$$

$$P(x_3=\text{Yellow}) = 800 / 1000 = 0.80$$

- Compute the probability of likelihood of evidences that goes in the numerator.

It is the product of conditional probabilities of the 3 features. If you refer back to the formula, it says $P(X_1 | Y=k)$. Here X_1 is 'Long' and k is 'Banana'. That means the probability the fruit is 'Long' given that it is a Banana. In the above table, you have 500 Bananas. Out of that 400 is long. So, $P(\text{Long} | \text{Banana}) = 400/500 = 0.8$.

Probability of Likelihood for Banana

$$P(x_1=\text{Long} | Y=\text{Banana}) = 400 / 500 = 0.80$$

$$P(x_2=\text{Sweet} | Y=\text{Banana}) = 350 / 500 = 0.70$$

So, the overall probability of Likelihood of evidence for Banana = $0.8 * 0.7 * 0.9 = 0.504$

1.1.4 Decision Tree algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithm. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too. The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data (training data). The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record's attribute. Based on comparison, we follow the branch

corresponding to that value and jump to the next node. We continue comparing our record's attribute values with other internal nodes of the tree until we reach a leaf node with predicted class value. As we know how the modelled decision tree can be used to predict the target class or the value. Now let's understanding how we can create the decision tree model.

Random forest algorithm is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. If you know the decision tree algorithm. You might be thinking are we creating more number of decision trees and how can we create more number of decision trees. As all the calculation of nodes selection will be same for the same dataset. Yes. You are true. To model more number of decision trees to create the forest you are not going to use the same apache of constructing the decision with information gain or gain index approach.

The random algorithm used in wide varieties applications. In this article, we are going address few of them. Below are some the application where random forest algorithm is widely used.

1. Banking
2. Medicine
3. Stock Market
4. E-commerce

1. Banking:

In the banking sector, random forest algorithm widely used in two main application. These are for finding the loyal customer and finding the fraud customers.

The loyal customer means not the customer who pays well, but also the customer who can take the huge amount as loan and pays the loan interest properly to the bank. As the growth of the bank purely depends on the loyal customers. The bank customers data highly analysed to find the pattern for the loyal customer based the customer details.

In the same way, there is need to identify the customer who are not profitable for the bank, like taking the loan and paying the loan interest properly or find the outlier customers. If the bank can identify theses kind of customer before giving the loan the customer. Bank will get a chance to not approve the loan to these kinds of customers.

In this case, also random forest algorithm is used to identify the customers who are not profitable for the bank.

2.Medicine

In medicine field, random forest algorithm is used identify the correct combination of the components to validate the medicine. Random forest algorithm also helpful for identifying the disease by analysing the patient's medical approach.

3.Stock Market

In the stock market, random forest algorithm used to identify the stock behaviour as well as the expected loss or profit by purchasing the particular stock.

4.E-commerce

In e-commerce, the random forest used only in the small segment of the recommendation engine for identifying the likely hood of customer liking the recommend products based on the similar kinds of customers.

Running random forest algorithm on very large dataset requires high-end GPU systems. If you are not having any GPU system.

1.2 ORGANIZATION OF THE PROJECT WORK

This section describes about various chapters included in this report.

Chapter 1 deals with an introduction to the project where the existing system is been discussed. It also gives an overview of algorithms' accuracy, precision control, data clustering and the data clustering techniques.

In Chapter 2, a detailed description of the literature survey of the papers which are referred during the project was summarized.

Chapter 3 and 4 gives a brief explanation on the aim and scope of the project and the methodology of the project. Here proposed system has been compared with the existing system. The issues in the existing system and the advantages of the proposed system are also discussed in this chapter.

In chapter 5, the Results and Discussion along with the screen layouts of each module has been depicted. It also includes the performance analysis of the system.

Chapter 6 deals with the summary and conclusion of the project. It also includes the future scope of the project.

JUPYTER NOTEBOOK

Project Jupyter is a project and community whose goal is to “Develop open source software, open-standards, and services for interactive computing across dozens of programming Languages”. It was spun off from IPython.



FIGURE 1.13: Jupyter Open Source Software

Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Now we have Jupyter installed , lets learn it. To get started , all we need to do is open our terminal application and to go to a folder of your choice . The recommendation is to use document folder to start out with and subfolder there called Notebooks or Something else that is easy to remember

The *ipython kernel*, referenced in this guide, executes python code. Kernels for many other languages exist (official case). When you open a Notebook document, the associated *kernel* is automatically launched. When the notebook is *executed* (either cell-by-cell or with menu *Cell -> Run All*), the *kernel* performs the computation and produces the results

The Jupyter Notebook is an open source web application that we can use to create and share document that contain live code, equations,visulisation, and text . Jupyter Notebook is maintained by people at Project Jupyter. Jupyter Notebook are a spin-off project from the Ipython Notebook project itself .

The name Jupyter comes from the core supported programming language that it supports: Julia, Python, and R.Jupyter ships with Ipython Kernel , which allows you to write your programs in Python , but they are Currently over 100 other kernels that we can also use .

GETTING UP AND RUNNING WITH JUPYTER NOTEBOOK

The Jupyter Notebook is not included with Python , so if we to try it out we will need to install Jupyter . There are many distribution of python Language . The most popular Cpython ,which is reference version of python is called as Python 3.

INSTALLATION: If we want to install Jupyter Then we can use handy tool that comes with Python called Pip to install Jupyter Notebbok like this:

```
$ pip install jupyter
```

The next most popular distribution of python is Anaconda . Anaconda has its own installer tool called Conda that we could use for installing a third- party package. However, Anaconda comes with many scientific preinstalled , including the Jupyter Notebook , so we don't actually do anything other than Installing Anaconda itself.

STARTING JUPYTER NOTEBOOK SERVER:

Now we have Jupyter installed , lets learn it. To get started , all we need to do is open our terminal application and to go to a folder of your choice . The recommendation is to use document folder to start out with and subfolder there called Notebooks or Something else that is easy to remember . Then just go to that Location in our command and run the following command:

```
$ jupyter notebook
```

. Depending on the type of computations, the *kernel* may consume significant CPU and RAM. Note that the RAM is not released until the *kernel* is shutdown. The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

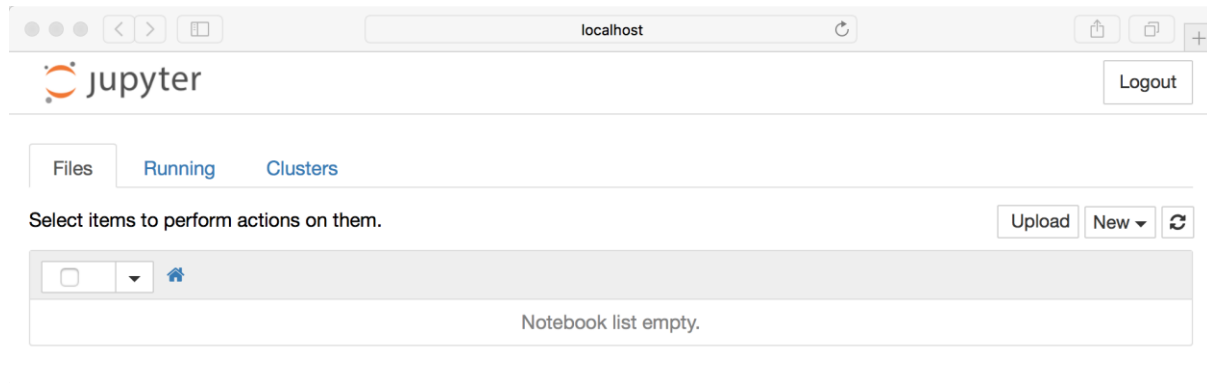


Fig1.14 Browser view of Jupyter Notebook

The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

- File
- Edit
- View
- Insert

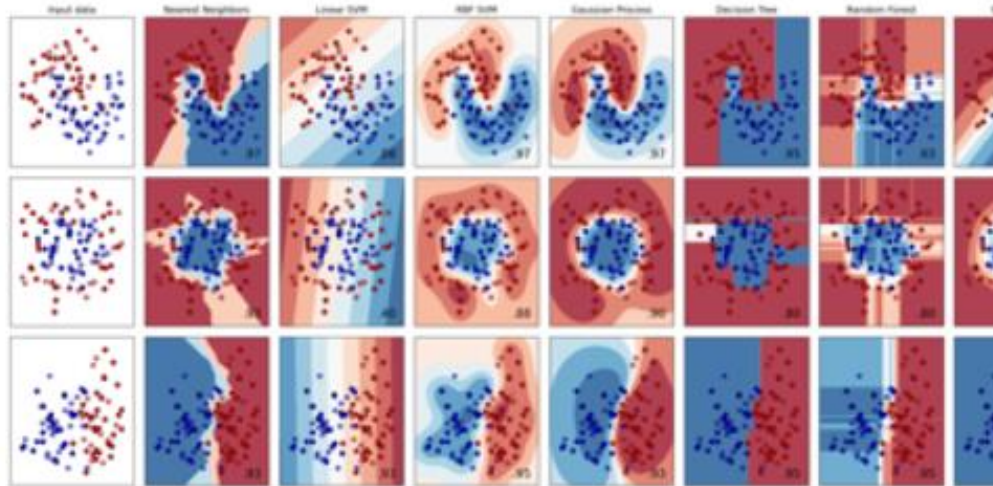
Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

FIGURE 1.16: Supervised Learning Classification



Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Housing Prices, Cancer Detection, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...

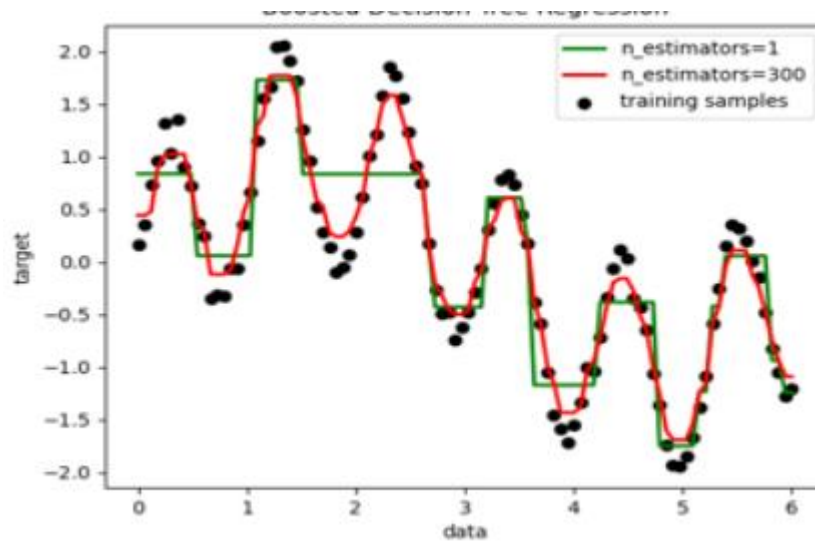


FIGURE 1.17: Supervised Learning Regression

CHAPTER 2

LITERATURE SURVEY

Tony C.T. Kuo and Arbee L.P. Chen in March 2000, proposed “Prediction of House Price” using hybrid machine learning technique. The paper predicted the result by using data and the data set by regression method. The method is then implemented on the basic data set and then predicted using Regression technique. The data are basically summarized using Machine Learning and accuracy of prediction is 92%.

TD Phan in 2018 in International Conference of Machine Learning proposed that House price Forecasting is an important Topic of Real Estate . The Literature attempts to derive useful knowledge from Historical data of Property Markets . Machine Learning Techniques are applied to analyze historical property transaction in Australia to discover useful models for house buyers and sellers. Revealed is the high discrepancy between house price in the most expensive and most affordable suburbs in the Melbourne . Moreover the Experiment Demonstrate that the combination of stepwise and Support Vector Machine that is based on mean squared error measurement is a competitive approach.

B Park and JK Bae in 2015 proposed that House sales are determined based on the standard & Poor’s case shiller Home Price indices and housing price Index of the Office of Federal Housing Enterprise Oversight . These Reflect the Trends of the US Housing Market.

This study uses machine learning algorithms as a research methodology to develop a housing price prediction model. To improve the accuracy of housing price prediction, this paper analyzes the housing data of 5359 townhouses in Fairfax County, Virginia, gathered by the Multiple Listing Service (MLS) of the Metropolitan Regional Information Systems (MRIS). We develop a housing price prediction model based on machine learning algorithms such as C4.5, RIPPER, Naïve Bayesian, and AdaBoost and compare their classification accuracy performance. We then propose an improved housing price prediction model to assist a house seller or a real estate agent make better informed decisions based on house price valuation. The experiments demonstrate that the RIPPER algorithm, based on accuracy, consistently outperforms the other models in the performance of housing price prediction.

J GU , M Zhu , L Jiang in 2011 proposed Forecasting of future housing price using SVM can be Very significant .There have been many previous research works that used various regression techniques to address the question of the changes house price. This work considers the issue of changing house price as a classification problem and applies machine learning techniques to predict whether house prices will rise or fall. This work applies various feature selection techniques such as variance influence factor, Information value, principle component analysis and data transformation techniques such as outlier and missing value treatment as well as box-cox transformation techniques.

Thus from the literature survey, the issues in the existing system has been studied which helps in improvising the proposed system. The aim and scope of the project is been discussed in the next chapter.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM

The Main Aim of the Project is to design a code which can Predict the Price of the House based upon the features provided using the Concept and Algorithms of Machine Learning.

3.2 SCOPE

The Prediction of House Price Using Machine Learning Algorithms are been used successfully from past two decades by numerous real Estate companies . A customer can approach an Agent to Enquire about price of house in a particular locality and even give their preferences about the size,number of rooms, Taxes , Windows etc and based on it the agent can give a predicted price of the house , so that the customer can avoid “ Overpaying “ if they are buying the house , or “ Underpaying” if they are trying to sell the house. This price Estimation method is highly accurate and can even give edge to the Real Estate company against its peers .

In cancer detection process using Machine Learning we use the concept of Supervised Learning Algorithm , and we will use the regression ,which predicts the discrete value output . And to implement the code for the same we will the open-source software i.e “Jupyter Notebook”. In Notebook we can use python data science tools such as some famous popular packages like Numpy,Pandas,Matplotlib and Scikit learn.

Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which have both input and output parameters. In this type of learning both training and validation datasets are labelled as shown in the figures below.

3.3 PROBLEM DEFINITION

All the algorithms that are used in the different papers and used in the data prediction are well defined but mathematical calculation is not so perfect in all the papers. The mathematical calculation is nothing but a scientific approach to solve any problem and it is one of the basic rules to solve the problem. The data mining technique don't use the basic attributes and also don't uses the data accuracy control. The data mining technique in some papers have not the hybrid approach. The hybrid approach of the machine learning is best way of prediction.

3.4 RELATED WORKS

There is ample related work in the fields directly related to this paper. ANN has been introduced to produce the highest accuracy prediction in the Real Estate field. The back propagation multilayer perception (MLP) of ANN is used to predict heart disease. The obtained results are compared with the results of existing models within the same domain and found to be improved. The data of heart disease patients collected from the UCI laboratory is used to discover patterns with NN, DT, Support Vector machines SVM, and Naive Bayes. The results are compared for performance and accuracy with these algorithms. The proposed hybrid method returns results of 86.8% for F-measure, competing with the other existing methods. The classification without segmentation of Convolutional Neural Networks (CNN) is introduced. This method considers the heart cycles with various start positions from the Electrocardiogram (ECG) signals in the training phase. CNN is able to generate features with various positions in the testing phase of the patient. A large amount of data generated by the medical industry has not been used effectively previously. The new approaches presented here decrease the cost and improve the prediction of heart disease in an easy and effective way. The various research techniques considered in this work for prediction and classification of heart disease using ML and deep learning (DL) techniques are highly accurate in establishing the efficacy of these methods.

into two classes in SVM as well as ANN for further analysis as shown in. The Back Propagation Neural Network (BPNN) with classification method is introduced, where the hypertension gene sequence is generated and then, thereafter the exact gene sequence. The performance of the BPNN techniques has been measured in the training phase as well as the testing phase with the various numbers of samples. The accuracy of this technique has improved in correspondence to the number of records.

3.5 EXISTING SYSTEM

The Existing Method to Give an Estimated House Price is old and Not Efficient . Even there is a high chance that the customer may end-up overpaying if He/She is buying a House or Underpaying If they are willing to sell it , because the only way to know the approx price of the house was through Real Estate Agents , Who can Even Charge a Fare Share of Amount to Make their Profit. And there is a high chance of Encountering Fraudulent deals in case of dubious real Estate Agents.

3.5.1 Disadvantages of existing system

1. The data set don't have mathematical approach for prediction.
2. The data attributes have basic data transaction which do not have accuracy.
3. The data prediction is also not good on note of different algorithms.

3.6 OVERVIEW OF THE PROPOSED SYSTEM

In this study, to Build our system we have imported the dataset from UCI machine Learning Repository and then split the dataset into proper Rows for "Data Values" and Columns for "Attributes", using Microsoft Excel . The code is designed Using Python Language as it consists of Large Number of In-Built Libraries like " Numpy, Pandas, Matplotlib , sci-kit Learn " which are used for data analysis , and we will Execute our Code in Jupyter Notebook. The Algorithm which is used in this Prediction Task is called as Supervised Machine Learning Regression Algorithm.

Supervised we already have Training data which will train our regressor , So that when a new set of features for a data is given , the regressor can give its Prediction. And if the Prediction is closed to the True value of Price our model is good or we need to look for a better ML Model. The Models which are Primarily used in This Analysis are:

1. Linear Regression Algorithm
2. Decision Tree Algorithm
3. Random Forest Algorithm

3.6.1 Advantages of the proposed system

1. The system contains five predictive algorithms that are used for the measurement of the data accuracy and precision.
2. The data prediction and the data accuracy raised up to 88.93% by random forest algorithm.
3. The mathematical approach is very clear, and the data prediction is done based on the measurement of the accuracy control.
4. The data calculation is very simple by using sigma function and sigmoid function.

Thus, the aim and scope of the project has been discussed and the issues with the existing system are also studied. The advantages in the proposed work are also mentioned. The methods and the algorithms implemented in the project are discussed in the next chapter.

CHAPTER 4

METHODOLOGY

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

4.1 HARDWARE REQUIREMENTS

Processor	: i4 or Later versions
Hard Disk	: 500 GB
RAM	: 1 GB

4.2 SOFTWARE REQUIREMENTS

Operating System	: Windows XP/7/8
Tool used	: Jupyter Notebook.

4.2.1 Overview of Jupyter Notebook

The Jupyter *Notebook App* is a server-client application that allows editing and running notebook documents via a web browser. The *Notebook App* can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter *Notebook App* has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels. A notebook kernel is a “computational engine” that executes the code contained in a Notebook document. The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official case). When you open a Notebook document, the associated *kernel* is automatically launched. When the notebook is *executed* (either cell-by-cell or with menu *Cell -> Run All*), the *kernel* performs the computation and

produces the results. Depending on the type of computations, the *kernel* may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shutdown. The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

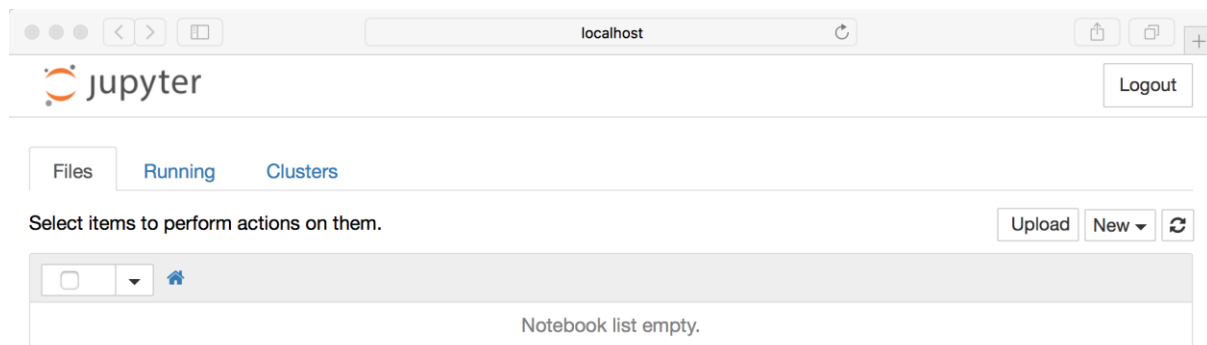


Fig4.1 Browser view of Jupyter Notebook

The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

- File
- Edit
- View
- Insert
- Cell
- Kernel
- Widgets
- Help

Let's go over the menus one by one. This article won't go into detail for every single option in every menu, but it will focus on the items that are unique to the Notebook application.

The first menu is the File menu. In it, you can create a new Notebook or open a pre-existing one. This is also where you would go to rename a Notebook. I think the most interesting menu item is the *Save and Checkpoint* option. This allows you to create checkpoints that you can roll back to if you need to.

Next is the Edit menu. Here you can cut, copy, and paste cells. This is also where you would go if you wanted to delete, split, or merge a cell. You can reorder cells here too. Note that some of the items in this menu are greyed out. The reason for this is that they do not apply to the currently selected cell. For example, a code cell cannot have an image inserted into it, but a Markdown cell can. If you see a greyed out menu item, try changing the cell's type and see if the item becomes available to use. The *View* menu is useful for toggling the visibility of the header and toolbar. You can also toggle *Line Numbers* within cells on or off. This is also where you would go if you want to mess about with the cell's toolbar. The *Insert* menu is just for inserting cells above or below the currently selected cell.

The *Cell* menu allows you to run one cell, a group of cells, or all the cells. You can also go here to change a cell's type, although I personally find the toolbar to be more intuitive for that. The other handy feature in this menu is the ability to clear a cell's output. If you are planning to share your Notebook with others, you will probably want to clear the

4.3 SYSTEM DESIGN

System design is a process by which the software requirements are translated into a representation of software components, interfaces and data necessary for the implementation phase. Implementation is the final phase of the project which reveals us the real outcome of the previous steps. Here the 25 software-based tasks directly dealing with the projects target is carried out. The implementation explains about the way the project work. This includes the software support, modules and their implementation.

Design is a meaningful engineering of something that is to be built. Software Design sites at the technical kernel of software engineering. Software design is a process through which the requirements are translated into a representation of the software i.e. the blueprint for constructing software. Design provides us with representation of software that can be assesses for quality.

4.4 SYSTEM ARCHITECTURE

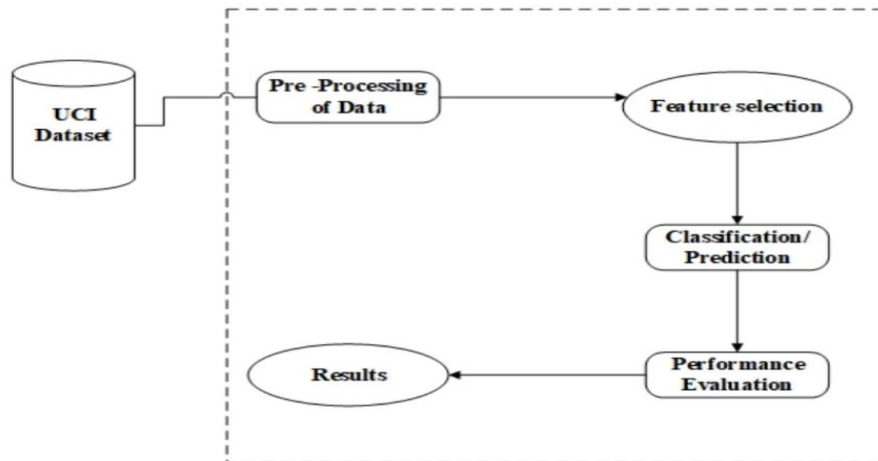


Fig4.2 System architecture of data prediction

4.5 MODULE DESIGN

The proposed system to be implemented has four modules namely

- Data set collection
- Pre-processing of data set
- Extraction of data set
- Prediction of results

4.5.1 Data set collection

Boston Housing Data is pre-processed after collection of various records. The dataset contains a total of 506 House records, out of which 404 is used for “Training Set” and rest “102” for testing. The data is obtained from UCI ML Repository in HTML Format. And then this raw data is pre-processed into Understandable format using “Microsoft Excel”, The dataset consists of 14 Attributes e.g. “CRIM, CHAS, RM, AGE, LSTAT, MEDV”, out of 14 Attributes there are 13 Continuous Attribute 1 Binary- Valued Attribute. The Excel file Formatted is termed as “data1.csv”, in this file Columns consist of 14 Attributes and row consist of its Corresponding data-value.

4.5.2 Pre-processing of data set

From 14 Attributes acquire columns of the Dataset and 506 Data Elements acquire Rows of the Dataset. The MEDV is the main attribute which is the “Price” which

has strong Positive Correlation with few Attributes(RM,ZN,DIS etc) and also have Strong Negative Correlation with Few Attributes(CRIM, LSTAT,NOX,TAX) , However if we want to make our analysis more simpler we can remove few attributes like 'CHAS' which doesn't have strong Correlation with the Final output "Price" From our Dataset. We have used mainly three ML Models in our Analysis Here " Linear Regression , Decision Tree , Random Forest ", and we have analyzed all the attributes using these 3 Models , To Compare the Results.

4.5.3 Extraction of data set

The Dataset we Obtained From UCI ML Repository was Completely Unstructured and was Present In HTML Formats onto two files , One file was having Information about data Elements and All Attributes , and other file consisted of Data – values of all Attributes. Then this Un-Organized data has been Separated into rows and Columns using Excel file ,Where 14 Attributes occupy all Columns of the file , while 506 Instances of data occupy Rows of the file, and this is how data is Extracted into a Structured Format.

4.5.4 Prediction of results

For The Prediction Part first we need to do TRAIN-TEST-SPLITTING , in which we use 402 instances for Training Our Model and 104 for Checking the Accuracy of the Predicted value , and decide which Model is best and the one with least Error , As per our Analysis for the Prediction using 3 Models , The Model with less Mean & Standard Deviation(std) , is the one having the Highest Accuracy in Predicting the Price . After Doing the Data Analysis we find out that "Random Forest Regression Model" is the best Model with more accuracy as compared to "Linear Regression & Decision Tress Model". And the Mean and Standard Deviation we Obtained are :[Mean : 3.302041119610937 & Standard Deviation : 0.6303744882489559] , which is very less compared to other Models.

4.6 ALGORITHM DESCRIPTION OF METHODS

There are certain methods that are used for the prediction of the algorithms. Let us study each algorithm one by one.

1) Decision Trees

For training samples of data D, the trees are constructed based on high entropy inputs. These trees are simple and fast constructed in a top down recursive divide and conquer (DAC) approach. Tree pruning is performed to remove the irrelevant samples on D.

$$Entropy = - \sum_{j=1}^m p_{ij} \log_2 p_{ij} \quad (1)$$

2) Language Model

For given input features x_i , y_i with input vector x_i of data D the linear form of solution $f(x) = mx + b$ is solved by subsequent parameters:

$$m = \frac{(\sum_i x_i y_i) - n \bar{x} \bar{y}}{(\sum_i x_i^2) - n \bar{x}^2} \quad (2)$$

$b = \bar{y} - m \bar{x}$ where $\bar{x}, \bar{y}$ are the means.

3) Support Vector Machine

Let the training samples having dataset $Data = \{y_i, x_i\}; i = 1, 2, \dots, n$ where $x_i \in R^n$ represent the i th vector and $y_i \in R^n$ represent the target item. The linear SVM finds the optimal hyperplane of the form $f(x) = w^T x + b$ where w is a dimensional coefficient vector and b is a offset. This is done by solving the subsequent optimization problem:

$$Min_{w,b,\xi_i} \frac{1}{2} w^2 + C \sum_{i=1}^n \xi_i \quad (3)$$

4) Random Forest

This ensemble classifier builds several decision trees and incorporates them to get the best result. For tree learning, it mainly applies bootstrap aggregating or bagging. For a given data, $X = \{x_1, x_2, x_3, \dots, x_n\}$ with responses $Y = \{y_1, y_2, y_3, \dots, y_n\}$ which repeats the bagging from $b = 1$ to B . The unseen samples x_0 is made by averaging the predictions $P_{b=1}^B f_b(x_0)$ from every individual trees on x_0 :

$$j = \frac{1}{B} \sum_{b=1}^B fb(x') \quad (4)$$

The uncertainty of prediction on these trees is made through its standard deviation,

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (fb(x') - \hat{f})^2}{B - 1}} \quad (5)$$

6

5) Naive Bayes

This learning model applies Bayes rules through independent features. Every instance of data D is allotted to the class of highest subsequent probability. The model is trained through the Gaussian function with prior probability $P(X_f) = \text{priority} \in (0 : 1)$

$$P(X_{f1}, X_{f2}, \dots, X_{fn} | c) = \prod_{i=1}^n P(X_{fi} | c) \quad (6)$$

$$P(X_f | c_i) = \frac{P(c_i | X_f) P(X_f)}{P(c_i)} \quad c \in \{ \text{benign}, \text{malignant} \}$$

At last, the testing data is categorized based on the probability of association:

$$c_{nb} = \arg \max_k P(c_k) \prod_{i=1}^n P(X_{fi} | c_k), \text{ for } k = 1, 2$$

Types of Supervised Learning

As we know, the Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

1. Classification Algorithm

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog**, etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

In classification algorithm, a discrete output function(y) is mapped to input variable(x)

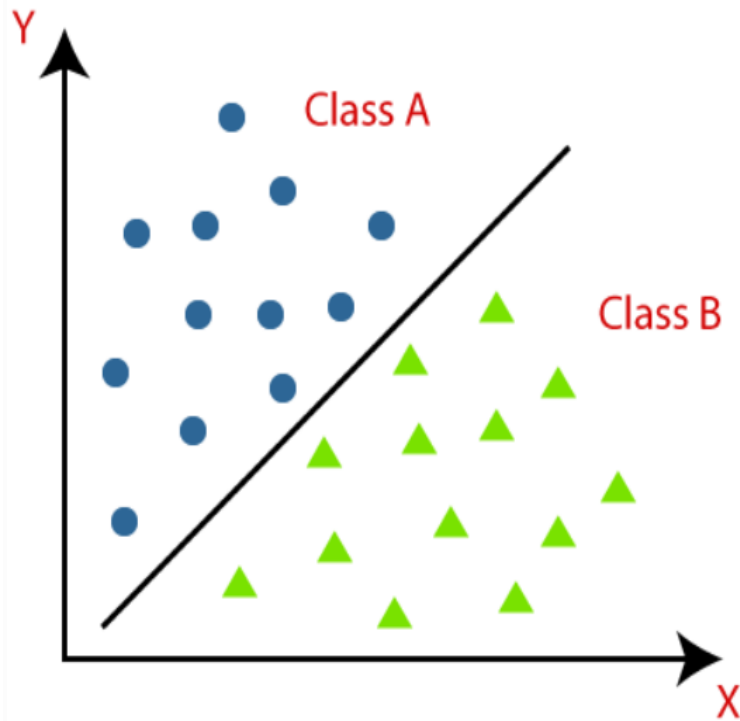
$$Y = f(x) , \text{ where } y = \text{categorical output}$$

- The best example of an ML classification algorithm is **Email Spam Detector**.
- The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Classification algorithms can be better understood using the below diagram. In the below diagram, there are two classes, class A and Class B. These classes have features that are similar to each other and dissimilar to other classes.



FIGURE 4.3: Graph of Classifier



The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

- **Binary Classifier:** If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
- **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
Example: Classifications of types of crops, Classification of types of music.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the Mainly two category:

- **Linear Models**
 - Logistic Regression
 - Support Vector Machines
- **Non-linear Models**
 - K-Nearest Neighbours
 - Kernel SVM
 - Naïve Bayes
 - Decision Tree Classification
 - Random Forest Classification

2. *Regression Algorithm*

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as **temperature, age, salary, price**, etc.

We can understand the concept of regression analysis using the below example:

Example: Suppose there is a marketing company A, who does various advertisement every year and get sales on that. The below list shows the advertisement made by the company in the last 5 years and the corresponding sales:

TABLE 4.1: Regression Analysis

Advertisement	Sales
\$90	\$1000
\$120	\$1300
\$150	\$1800
\$100	\$1200
\$130	\$1380
\$200	??

Now, the company wants to do the advertisement of \$200 in the year 2019 and wants to know the prediction about the sales for this year. So to solve such type of prediction problems in machine learning, we need regression analysis.

Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables. It is mainly used for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

In Regression, we plot a graph between the variables which best fits the given datapoints, using this plot, the machine learning model can make predictions about the data. In simple words, "Regression shows a line or curve that passes through all the datapoints on target-predictor graph in such a way that the vertical distance between the datapoints and the regression line is minimum." The distance between datapoints and line tells whether a model has captured a strong relationship or not.

Some examples of regression can be as:

- Prediction of rain using temperature and other factors
- Determining Market trends
- Prediction of road accidents due to rash driving.

Types of Regression

There are various types of regressions which are used in data science and machine learning. Each type has its own importance on different scenarios, but at the core, all the regression methods analyze the effect of the independent variable on dependent variables. Here we are discussing some important types of regression which are given below:

- **Linear Regression**
- **Logistic Regression**
- **Polynomial Regression**
- **Support Vector Regression**
- **Decision Tree Regression**
- **Random Forest Regression**

4.7 EXPERIMENTATIONAL ENVIRONMENT

4.7.1 Datasets

The Dataset for the analysis is taken from UCI Machine Learning Repository which was Initially in HTML Format , with two files one consisting of All Attributes and their names and other consisting of data-values of attributes. The data is the Extracted in Microsoft Excel in Rows and Columns , and also label the columns with their corresponding Attribute. The title of dataset taken from UCI ML Repository is “ Boston Housing Data”,

The Dataset consist of 506 Instances(Data-elements) , and there are 13 Attributes like CRIM(per capita Crime rate by town), RM(average number of rooms per dwelling) , RAD(Index of accessibility to radial hiways , ZN(Proportional of residential land zooned),Age(Proportion of owner-occupied units bulid prior to 1940) etc , all the Allributes values are Numeric values and no Null attribute is present.

4.7.2 Experimental setup for evaluation

We have used an Jupyter Notebook to perform the Prediction of House Price from Boston Housing Data from UCI repository. Figure 1 depicts the evaluation of the experiment by step-by-step stages. In the first step, the UCI dataset is loaded and the data becomes ready for pre-processing. The subset of 13 attributes (CRIM , ZN,INDUS ,CHAS NOX,RM,AGE,DIS,RAD,TAX) is selected from the pre-processed data set of heart disease. The three existing models for Housing price prediction we used three Models:

1. Linear Regression Model
2. Decision Tree Model
3. Random Forest Model

4.8 EVALUATION RESULTS

The prediction models are developed using 13 features and the accuracy is calculated for modeling techniques. The best classification methods are given below in Table 3. This table compares the accuracy, Regression error, precision, F-measure, sensitivity and specificity. The highest accuracy is achieved by Random Forest Regressor method in comparison with existing methods.

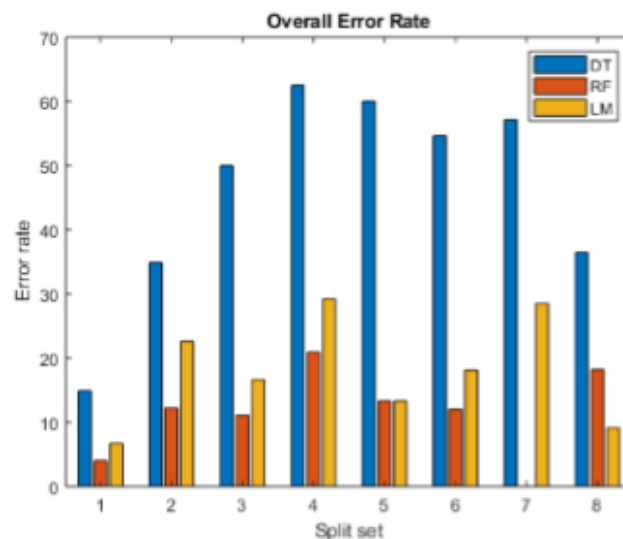


Fig4.4 Overall error rate of data set

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS AND DISCUSSION

The code for the Analysis is written in Python Language and executed in Jupyter Notebook , then Attributes which we obtain from Dataset , we used train-test-splitting method to predict the price of the house, to do that we used around 250 plus instances to train our model I.e we basically from model with values of differen attributes and along with price , and after our model is trained with sufficient amount of data we can move further to test its accuracy by test-data set , in which we provide the model with values of 13 attributes and the our model tries to predict the price , If the value predicted by model is closer to the actual value then our model is having high accuracy , or in former case we need to change our model , here we used three models for analysis i.e Linear Regression , Decision Tree , Random Forest & we found out that Random forest Regression model is Predicting Price of the House with Highest accuracy. An thus we use Random Forest Regressor to Improve our Result.

5.2 PERFORMANCE ANALYSIS

The proposed system when compared to that of the existing system is efficient in terms of accuracy, speed and reliability of matching

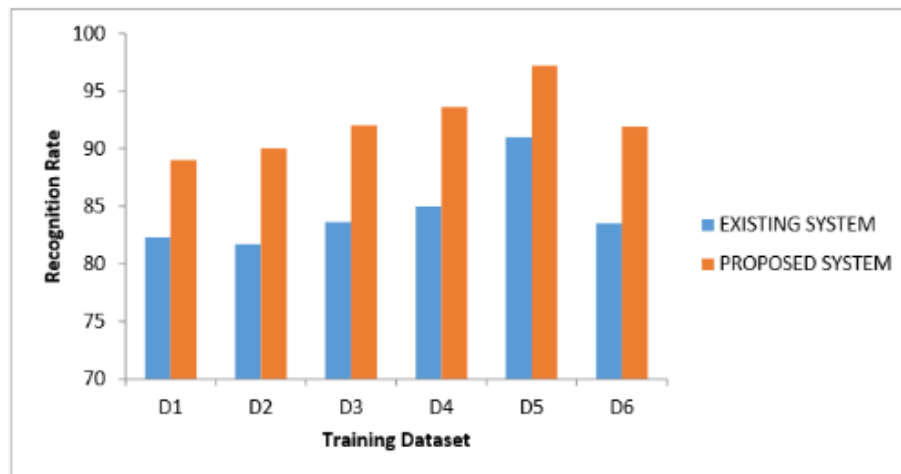


Fig:5.1 :Comparison between Proposed and Existing System

In the existing system Linear regression algorithm is used which is slow in processing and computation and is affected by variation to noisy data and fussy data

whereas Random forest algorithm is not variant to noisy, unorthodox and fussy and fast in predicting results compared to other prediction matching algorithms.

5.3 BENCHMARKING OF THE PROPOSED MODEL

Benchmarking is needed to compare the performance of the existing models compared with the proposed model. This method is used to identify whether the proposed method is the best and improves accuracy or not. The accuracy is calculated with the number of feature selection and the model generated results. The Random forest Regression Model is more accurate when compared to Linear Regression & Decision Tree Regression Model , because the Price Predicted By Random Forest Regressor is pretty much close to the actual value based upon the 13 attributes , which makes our model highly efficient to use.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION

Predicting the Price of the house , can help the customer to avoid Overpaying if he/she is trying to buy the house , or getting Underpaid if they are willing to sell the house . Machine Learning Algorithm are highly accurate and trustworthy , if they are designed well and can get an accuracy of 90 to 95% . Housing Price Prediction Model is interesting thing , because if an Real Estate company uses these kind of model , it can get an Edge over its competitors and can also help to increase its client base . Because if Customer are getting Fair and accurate price of their house they can even give a pie of money to Real Estate company for their Help . Further extension of this study is highly desirable to direct the investigations to real-world datasets instead of just theoretical approaches and simulations. The proposed Random Forest, Decision Tree & Linear Regression Methods are Highly accurate in these kinds of practices to predict price. The future course of this research can be performed with diverse mixtures of machine learning techniques to better prediction techniques. Furthermore, new feature selection methods can be developed to get a broader perception of the significant features to increase the performance of House Price Prediction.

6.2 FUTURE ENHANCEMENT

The model is supposed to be enhanced with accuracy rate and precision over different algorithm which are used to predict the algorithm. The algorithms that are used have 94% accuracy by random forest algorithm, but all different algorithms have precision up to 85 to 90% on an average. The data prediction is basically have all the data set present that are well defined and well placed according to prediction average.

REFERENCES

- B Park , JK Bae – 2015 , House sales are determined based on Standard & Poor's case chiller home Price indices Housing price Index of the Office of Federal Housing Enterprise(OFHEO).
- TD Phan , 2018 on International Conference on Machine Learning , that House

Price Forecasting is an Important topic of real estate to derive useful Knowledge from Historical Data of Property Markets.

- O Truong , M Nguyen , H Dang , B Mei in 2020, Abstract House Price Index(HPI) is commonly used to estimate changes in housing Price . Since price is strongly correlated to other factors such as location, area , Population, it requires other information apart from HPI to predict Individual Housing Price.
- D Banerjee , S Dutta , 2017 IEEE International Conference The phenomenon of the falling or rising of the house prices has attracted interest from the researcher as well as many other interested parties.
- A Varma , A Sarma , S Doshi , 2018 IEEE International Conference Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price.
- J Guo , S Chiang , M liu , CC Yang , of strategic Property in 2020 Housing frenzies in China have attracted widespread global attention over the past few years, but the key is how to more accurately forecast housing prices in order to establish an effective real estate policy. Based on the ubiquitousness and immediacy of Internet data, this research adopts a broader version of text mining to search for keywords in relation to housing prices and then evaluates the predictive abilities using machine learning algorithms. Our findings indicate that this new method, especially random forest, not only detects turning points, but also offers prediction ability that clearly outperforms traditional regression analysis.

APPENDIX

A. SAMPLE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import os
print(os.listdir("../input"))
sns.countplot(x="target", data=df, palette="bwr")
plt.show()
countNoDisease = len(df[df.target == 0])
```

```

countHaveDisease = len(df[df.target == 1])
print("Percentage of Patients Haven't Heart Disease: {:.2f}%".format((countNoDisease /
(len(df.target))*100)))
print("Percentage of Patients Have Heart Disease: {:.2f}%".format((countHaveDisease /
(len(df.target))*100)))
sns.countplot(x='sex', data=df, palette="mako_r")
plt.xlabel("Sex (0 = female, 1= male)")
plt.show()
sns.countplot(x='sex', data=df, palette="mako_r")
plt.xlabel("Sex (0 = female, 1= male)")
plt.show()
def update(weight,bias,x_train,y_train,learningRate,iteration) :
    costList = []
    index = []

    #for each iteration, update weight and bias values
    for i in range(iteration):
        cost,gradients = forwardBackward(weight,bias,x_train,y_train)
        weight = weight - learningRate * gradients["Derivative Weight"]
        bias = bias - learningRate * gradients["Derivative Bias"]

        costList.append(cost)
        index.append(i)

    parameters = {"weight": weight,"bias": bias}

    print("iteration:",iteration)
    print("cost:",cost)

    plt.plot(index,costList)
    plt.xlabel("Number of Iteration")
    plt.ylabel("Cost")
    plt.show()
for i in range(1,20):
    knn2 = KNeighborsClassifier(n_neighbors = i) # n_neighbors means k
    knn2.fit(x_train.T, y_train.T)
    scoreList.append(knn2.score(x_test.T, y_test.T))

plt.plot(range(1,20), scoreList)
plt.xticks(np.arange(1,20,1))

```

```

plt.xlabel("K value")
plt.ylabel("Score")
plt.show()

acc = max(scoreList)*100
accuracies['KNN'] = acc
print("Maximum KNN Score is {:.2f}%".format(acc))

from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train.T, y_train.T)

acc = nb.score(x_test.T, y_test.T)*100
accuracies['Naive Bayes'] = acc
print("Accuracy of Naive Bayes: {:.2f}%".format(acc))

```

```

from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train.T, y_train.T)

acc = dtc.score(x_test.T, y_test.T)*100
accuracies['Decision Tree'] = acc
print("Decision Tree Test Accuracy {:.2f}%".format(acc))

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
rf.fit(x_train.T, y_train.T)

acc = rf.score(x_test.T, y_test.T)*100
accuracies['Random Forest'] = acc
print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))

# Predicted values
y_head_lr = lr.predict(x_test.T)
knn3 = KNeighborsClassifier(n_neighbors = 3)
knn3.fit(x_train.T, y_train.T)
y_head_knn = knn3.predict(x_test.T)
y_head_svm = svm.predict(x_test.T)
y_head_nb = nb.predict(x_test.T)
y_head_dtc = dtc.predict(x_test.T)
y_head_rf = rf.predict(x_test.T)

In [40]:
from sklearn.metrics import confusion_matrix

```

```
cm_lr = confusion_matrix(y_test,y_head_lr)
cm_knn = confusion_matrix(y_test,y_head_knn)
cm_svm = confusion_matrix(y_test,y_head_svm)
cm_nb = confusion_matrix(y_test,y_head_nb)
cm_dtc = confusion_matrix(y_test,y_head_dtc)
cm_rf = confusion_matrix(y_test,y_head_rf)
```

```
plt.suptitle("Confusion Matrixes",fontsize=24)
plt.subplots_adjust(wspace = 0.4, hspace= 0.4)
```

```
plt.subplot(2,3,1)
plt.title("Logistic Regression Confusion Matrix")
sns.heatmap(cm_lr,annot=True,cmap="Blues",fmt="d",cbar=False, annot_kws={"size":
24})
```

```
plt.subplot(2,3,2)
plt.title("K Nearest Neighbors Confusion Matrix")
sns.heatmap(cm_knn,annot=True,cmap="Blues",fmt="d",cbar=False, annot_kws={"size
": 24})
```

```
plt.subplot(2,3,3)
plt.title("Support Vector Machine Confusion Matrix")
sns.heatmap(cm_svm,annot=True,cmap="Blues",fmt="d",cbar=False, annot_kws={"size
": 24})
```

```
plt.subplot(2,3,4)
plt.title("Naive Bayes Confusion Matrix")
sns.heatmap(cm_nb,annot=True,cmap="Blues",fmt="d",cbar=False, annot_kws={"size":
24})
```

```
plt.subplot(2,3,5)
plt.title("Decision Tree Classifier Confusion Matrix")
sns.heatmap(cm_dtc,annot=True,cmap="Blues",fmt="d",cbar=False, annot_kws={"size"
: 24})
```

```
plt.subplot(2,3,6)
plt.title("Random Forest Confusion Matrix")
```


B. SCREENSHOTS

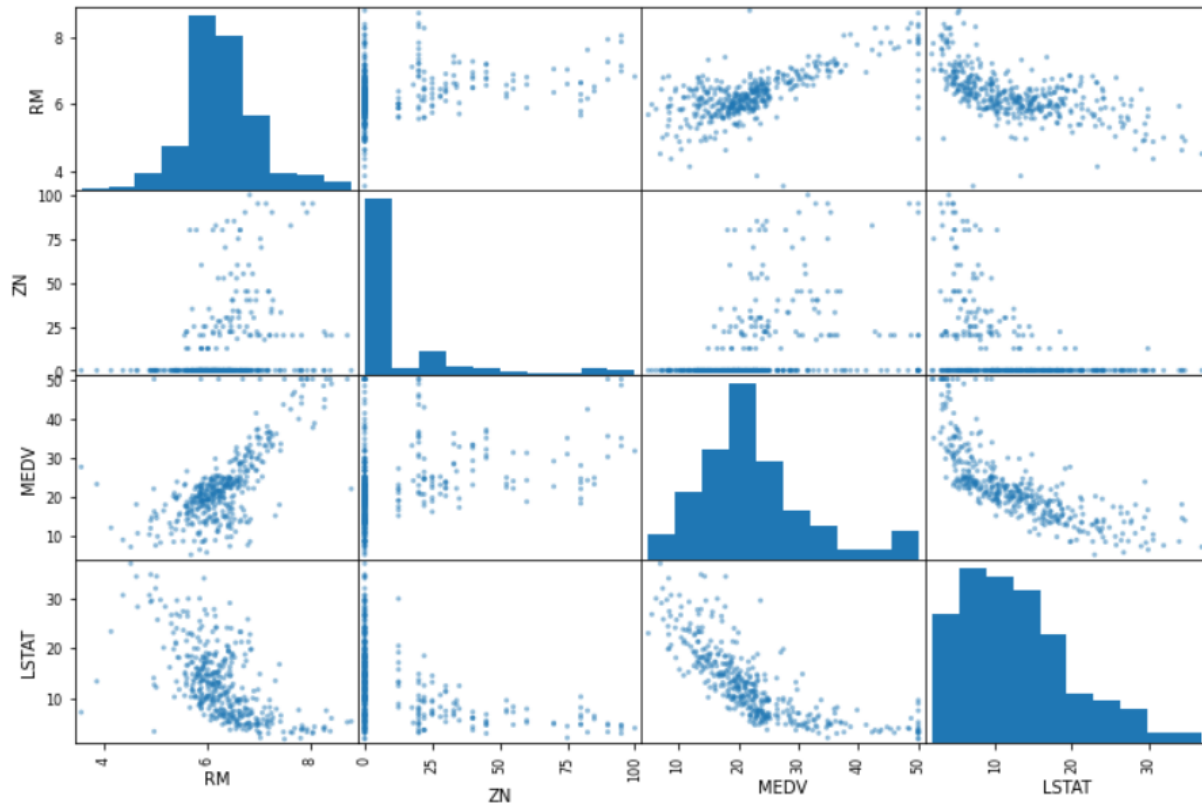


Fig B.1 : Scatter Matrix of few Attributes

```
housing.plot(kind="scatter" , x="RM" , y="MEDV" , alpha=0.8) # VERY HIGH POSITIVE CORRELATION
```

```
<AxesSubplot:xlabel='RM', ylabel='MEDV'>
```

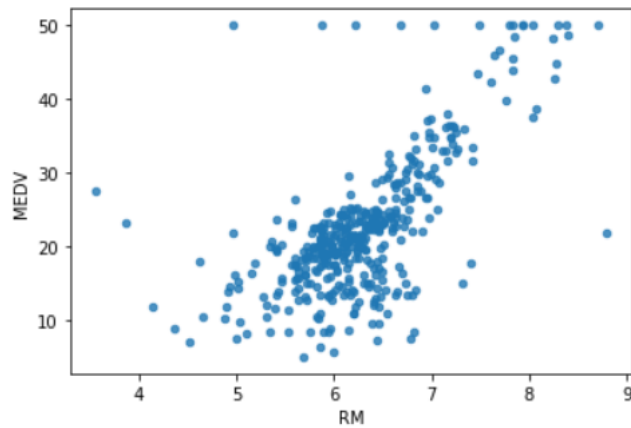


Fig B.2 Positive Correlation Scatter Plot

```
housing.plot(kind="scatter" , x="TAXRM" , y="MEDV" , alpha=0.8) #VERY HIGH NEGATIVE CORRELATION
```

```
<AxesSubplot:xlabel='TAXRM', ylabel='MEDV'>
```

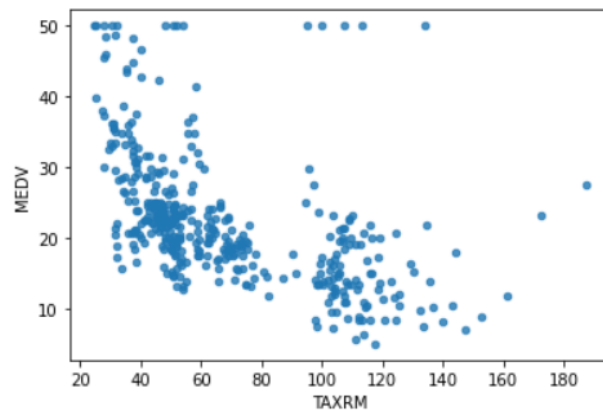


Fig B.3 Negative Correlation Scatter Plot