

Predictive Analytics for Breast Cancer Detection

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

SHIKHA SHARMA

(EN16CS301243)

Under the Guidance of

Prof. SACHIN SOLANKI



Department of Computer Science & Engineering

Faculty of Engineering

MEDI-CAPS UNIVERSITY, INDORE- 453331

JANUARY 2020 - MAY 2020

Predictive Analytics for Breast Cancer Detection

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

**SHIKHA SHARMA
(EN16CS301243)**

Under the Guidance of

Prof. SACHIN SOLANKI



**Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331**

JANUARY 2020 - MAY 2020

Report Approval

The project work “**Predictive Analytics for Breast Cancer**” is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name: Prof. Sachin Solanki

Designation: Professor

Affiliation: Medi-Caps University

External Examiner

Name:

Designation

Affiliation

Declaration

I hereby declare that the project entitled “**Predictive Analytics For Breast Cancer Detection**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science & Engineering’ completed under the supervision of **Prof. Sachin Solanki (Professor) Computer Science**, Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Shikha Sharma

(EN16CS301243)

Certificate

I/we, Prof. **Sachin Solanki** certify that the project entitled “**Predictive Analytics For Breast Cancer**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Shikha Sharma** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Prof. Sachin Solanki

Computer Science

Medi-Caps University, Indore

Dr Ravi Changle

TCS

Talentsprint

Dr. Suresh Jain

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somani**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda**, Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Suresh Jain** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Mr. Ravi Changle, Talentsprint** as well as to my Internal Guide, **Prof. Sachin Solanki**, Professor, Department of Computer Science, MU, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Shikha Sharma

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

Abstract

The project Predictive Analytics For Breast Cancer is based in Machine Learning. In this project I have implemented machine learning tools on real life scenarios. The objective of this paper is to compare and identify an accurate model to predict the incidence of breast cancer based on various patients' clinical records. Four data mining models are applied in this paper, i.e., support vector machine (SVM), Naive Bayes classifier, AdaBoost tree. Furthermore, feature space is highly discussed in this paper due to its high influence on the efficiency and effectiveness of the learning process. To test the influence of feature space reduction, a hybrid between principal component analysis (PCA) and related data mining models is proposed, which applies a principle component analysis method to reduce the feature space. To evaluate the performance of these models, two widely used test data sets are used, Wisconsin Breast Cancer Database (1991) and Wisconsin Diagnostic Breast Cancer (1995). 10-fold cross-validation method is implemented to estimate the test error of each model. The results performed by this analysis demonstrate a comprehensive trade-off between these strategies and also provides a detailed evaluation on the models. It is expected that in real application, physicians and patients can benefit from the feature recognition outcome to prevent breast cancer I have used various classification techniques that are used for technical analysis of claim prediction like logistic regression, support vector machine, k nearest neighbor, decision forest and random forest algorithm.

A web application for providing interface to the user is built using flask python. Web application synchronizes breast cancer prediction with other files and modules that are present inside the project. A beautiful and elegant but simple and easy to use interface is provided.

Keywords: Technical Analysis, Digital Era, Machine Learning, Web-Application.

Table of contents

	Title	Page No.
	Report Approval	ii
	Declaration	iii
	Certificate	iv
	Acknowledgement	v
	Abstract	vi
	Table of Contents	vii
	List of figures	viii
	Abbreviations	ix
Chapter 1	Introduction	01
	1.1 Introduction	02
	1.2 Training Details	02
	1.3 Objectives	03
	1.4 Source of Data	03
	1.5 Plan of Action	04
Chapter 2	System Requirement and Analysis	05
	2.1 Information Gathering	06
	2.2 System Feasibility	06
	2.2.1 Technical	06
	2.2.2 Economical	06
	2.3 Platform Specification	06
	2.3.1 Hardware	07
	2.3.2 Software Implementation Language/ Technology	07
Chapter 3	Design	10
	3.1 Implementation of modules	11
	3.1.1 Back-end	11
	3.1.2 Front-end	23
	3.2 Project Code	25
Chapter 4	Testing	26
	4.1 Testing Objective	27
	4.2 Testing Principles	27
	4.3 Testing Methods Used	28
	4.4 Interface Design	29
Chapter 5	Limitations	31
Chapter 6	Results and Discussions	33
Chapter 7	Conclusions	35
Chapter 8	Future scope	37
	Bibliography	39

List of Figures

Figure No.	Figure Name	Page No.
Fig 1.1	Data Insights	3
Fig 1.2	Plan of action	4
Fig 2.1	Types of Learning	8
Fig 2.2	Supervised Learning	8
Fig 2.3	Unsupervised Learning	9
Fig 3.1	Missing values corresponding to features	11
Fig 3.2	Count-plot of target variable	11
Fig 3.3	Bar-plot of correlation of all features with target variable	12
Fig 3.4	TSNE Distribution	12
Fig 3.5	Heatmap	13
Fig 3.6	Handling imbalanced data	16
Fig 3.7	SVM	17
Fig 3.8	KNN	18
Fig 3.9	Decision Tree	19
Fig 3.10	Random Forest Classifier	20
Fig 3.11	AdaBoostClassifier	21
Fig 3.12	Flask	23
Fig 3.13	Azure Model	25
Fig 4.1	UI Form	29
Fig 4.2	UI Form	30
Fig 4.3	Prediction	30
Fig 6.1	Prediction Performance of the models	34

Abbreviations

Abbreviation	Full Form
SVM	Support Vector Machine
KNN	K Nearest Neighbor
EDA	Exploratory Data Analysis
PCA	Principal Component Analysis
CV	Cross Validation

CHAPTER 1

Introduction

1.1 Introduction

The project involves the development of “Predictive Analytics for Breast Cancer” according to the project allotted by TCS to us. This system performs the task of calculating that a particular individual will claim insurance or not on the basis of details about the individual.

This project is made using machine learning. The most important advantage of Machine Learning (ML) to use in Insurance Industry is to facilitate data sets. Machine learning (ML) can be successfully useful across Structured, Semi Structured or Unstructured datasets. Machine learning can be used accurate across the value chain to identify with risk, claims and customer actions, by means of advanced predictive accurateness. The probable applications of machine learning in insurance are plentiful from perceptive risk appetite and premium leakage, to expense administration, subrogation, proceedings and fraud detection.

Machine learning is not a novel technology; this technology is following from the last few decades. There are 3 main categories of learning they are supervised learning, Unsupervised Learning and reinforcement learning. The greater part of the insurers are following Supervised Learning from last few decades for assessing the risk by means of known parameters in dissimilar combinations to acquire the preferred outcome.

In this paper, a study about breast cancer feature extraction based on data mining methods is discussed. The analysis focuses on the efficiency and effectiveness of testing four models, support vector machine (SVM), artificial neural network (ANN), Naïve Bayes classifier, and AdaBoost tree. In addition, principal component analysis (PCA) as a dimension reduction method is also analyzed in this paper. To evaluate the performance of these models, two data sets are used, Wisconsin Breast Cancer Database (1991), which contains 699 instances and 11 patient attributes for each instance and Wisconsin Diagnostic Breast Cancer (1995), which includes 569 instances and 32 patient attributes. These data sets are commonly used as sample data sets by many researchers. In order to provide a sufficient evaluation of the different models, 10-fold cross-validation method is implemented to estimate testing accuracy. The results performed by this analysis provide a detailed evaluation of the four models and also shows a comprehensive trade-off between different strategies.

1.2 Training details:

Training is done from TCS through Talentsprint after placement TCS is providing training to some students in AI and ML field. Our training includes:

Python, Machine Learning, Flask, Microsoft Azure Model Creation and Deployment, Abby Flexicapture and Chatbots.

Training Period: 5 February 2020 to 15 June 2020

1.3 Objectives

The objective of “Predictive Analytics for Breast Cancer” is to recognize that a person is suffering from Breast Cancer or not. If a person has Breast Cancer its said to be Malignant else Benign. This prediction is done from data not images. Since clearly seen it’s a two class classification problem. Various classification approaches have been applied to predict the breast cancer accurately. Various features are there which helps in predicting breast cancer. Since it’s a health issue it needs to be handled accurately and for this we need to reduce our wrong predictions as much as possible.

There are other objectives of this project too like implementing machine learning tools on real life scenarios & to perform EDA (exploratory data analysis) on the dataset. Exploratory Data Analysis is one of the important steps in the data analysis process. Here, the focus is on making sense of the data in hand – things like formulating the correct questions to ask to your dataset, how to manipulate the data sources to get the required answers, and others. This is done by taking an elaborate look at trends, patterns, and outliers using a visual method. Exploratory Data Analysis is a crucial step before you jump to machine learning or modeling of your data. It provides the context needed to develop an appropriate model – and interpret the results correctly.

1.4 Source of Data

The dataset used is Wisconsin Breast Cancer Dataset which we collected from UCI Learning Repository. This data includes 569 instances and 31 columns. Different Features are present which determine the breast cancer. The dataset includes features such as radius, smoothness, texture, perimeter, area, compactness, concavity, concave points, symmetry, and fractal dimension. Then we have mean, std and worst values for each of these features and that how we get 30 columns and we also have an Id and diagnosis column.

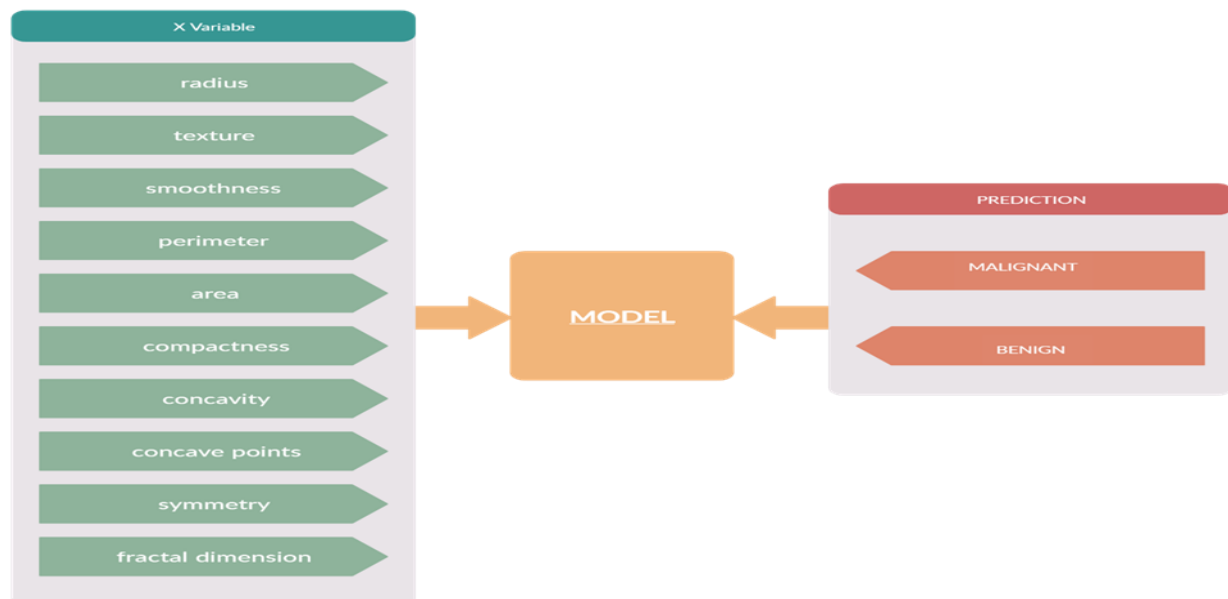


Figure 1.1 Data Insights

1.5 Plan of Action

For the initialization of project our major concern was to see data which is best suited for this purpose. As we discussed data gathering in previous section we then came to data analysis the preprocessing then we applied models and also evaluated our model. As shown below:

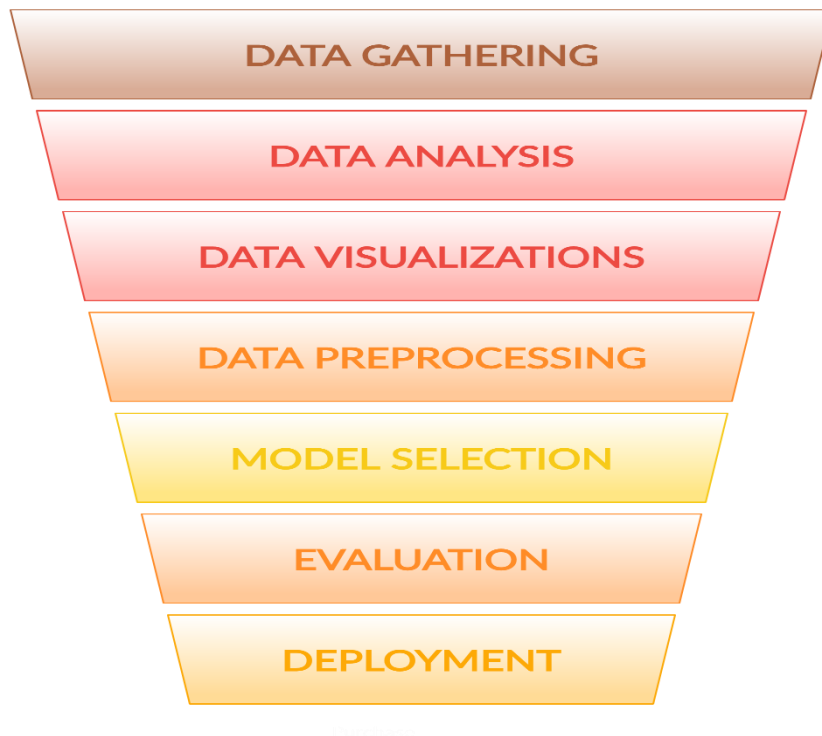


Figure 1.2: Plan of Action

CHAPTER-2

System Requirement and Analysis

2.1 INFORMATION GATHERING

2.1.1 Function Requirement

- Functional requirements specifies a function that a system or system component must be able to perform. It can be documented in various ways. The most common ones are written description in document, and use case.
- Use cases can be textual enumeration lists as well as diagrams, describing user actions. Each use case illustrates behavioral scenarios through one or more functional requirements.

2.1.2 Non-Functional Requirement

- Non-functional requirements are any other requirements than functional requirements. These are the requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.
- Non-functional requirements are in the form of what "system shall be ", an overall property of the system as a whole or of a particular aspect and not a specific function.

2.2 System Feasibility

2.2.1 Technical

The project covers all the scope of practicality and is technically feasible.

- It will be built considering industry standards, and using the latest technologies which will ensure its smooth functioning.
- The project will run on user devices with internet.

2.2.2 Economical

- This application uses open source platform for development and it is live and can be accessed anytime.
- The only hardware requirements to access this project is appropriate computing device.
- Since android phones are present with almost everyone today and the software required are open source and free, there doesn't arise any question of why this project may not be economically feasible in any way.

2.3 Platform Specification (Development & Deployment)

2.3.1 Software Requirements

- Python
- Flask
- HTML5, CSS3

2.3.2 Hardware Requirements

- Windows XP or Above
- 2GB of RAM
- Any Dual Core Processor or above

2.3.3 Software Implementation Language/ Technology

Languages and Tools Used

1. **Python:** Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.
2. **Flask:** Flask is a micro web framework written in Python. Flask is a lightweight WSGI web application framework. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.
However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask Program
3. **Front End (HTML):** Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. HTML describes the structure of a web page semantically and originally included cues for appearance of the document.
4. **Machine Learning:** Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Two of the most widely adopted machine learning methods are **supervised learning** which trains algorithms based on example input and output data that is labeled by humans, and **unsupervised learning** which provides the algorithm with no labeled data in order to allow it to find structure within its input data.

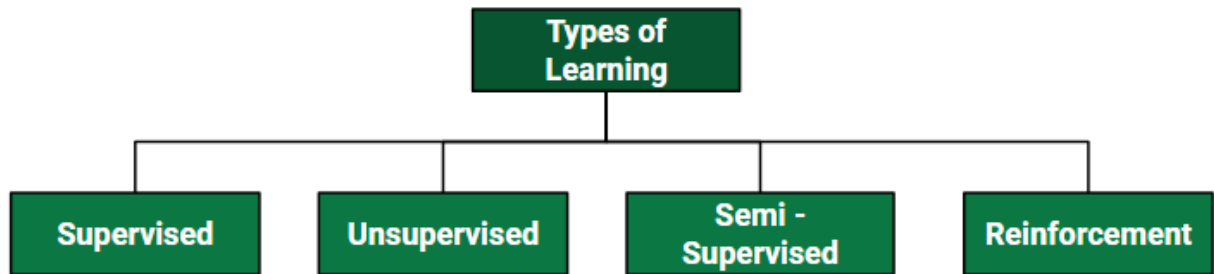


Figure 2.1 Types of learning

Supervised Learning

Supervised Learning is the process of making an algorithm to learn to map an input to a particular output. This is achieved using the labelled datasets that you have collected. If the mapping is correct, the algorithm has successfully learned. Else, you make the necessary changes to the algorithm so that it can learn correctly. Supervised Learning algorithms can help make predictions for new unseen data that we obtain later in the future.

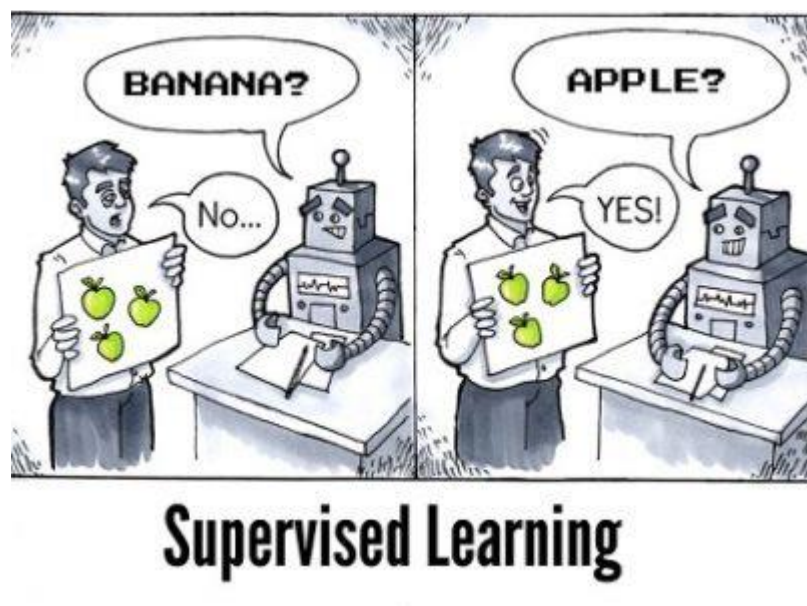


Figure 2.2 Supervised learning

Types Of Supervised Learning

- **Classification:** The goal is to predict discrete values, e.g. {1,0}, {True, False}, {spam, not spam}.
- **Regression:** The goal is to predict continuous values, e.g. home prices.

Unsupervised Learning

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabeled data.

Unsupervised learning algorithms allows you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods.

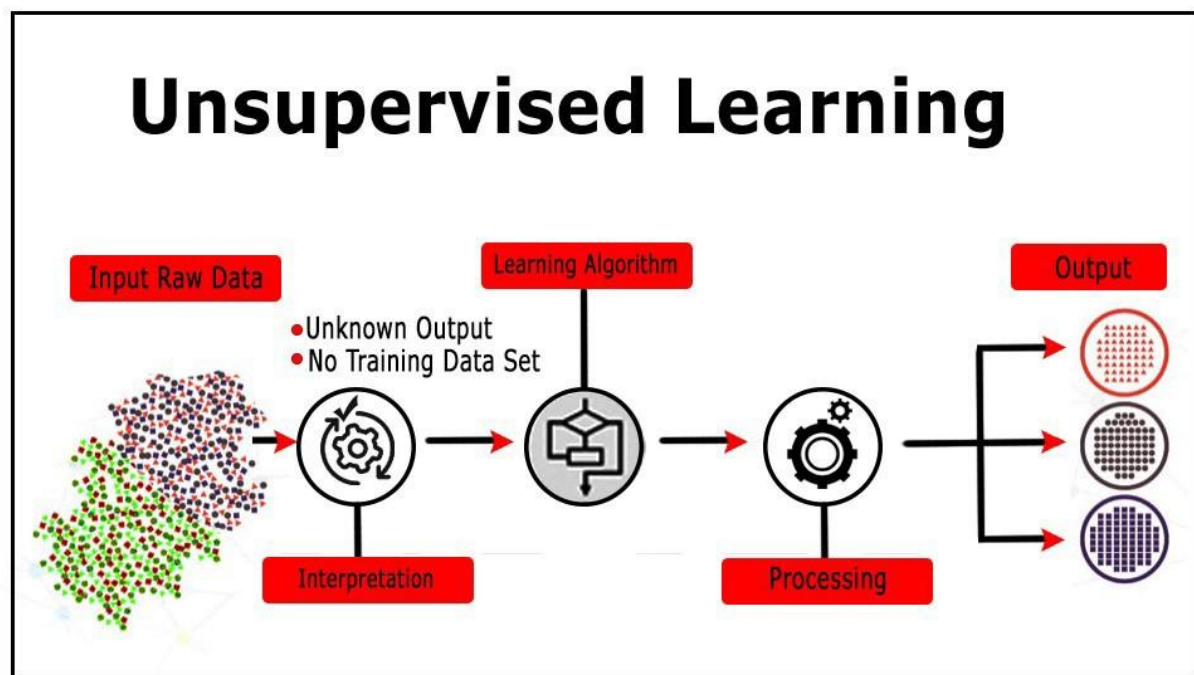


Figure 2.3 Unsupervised Learning

CHAPTER-3

DESIGN

3.1 Implementation of modules

Modules implementation can be divided into two parts front end and backend

3.1.1 Backend

Data Analysis and Visualization:

In this step we have analyzed the data by describing the data, determining the correlation values etc. Here we have plotted various graphs to analyze the dataset including count-plot, bar-plot, heatmap, etc. in order to better understand the data and the complexities involved in processing the dataset to carry out the predictions. Following are the various visualizations.

- Missing Values corresponding to features

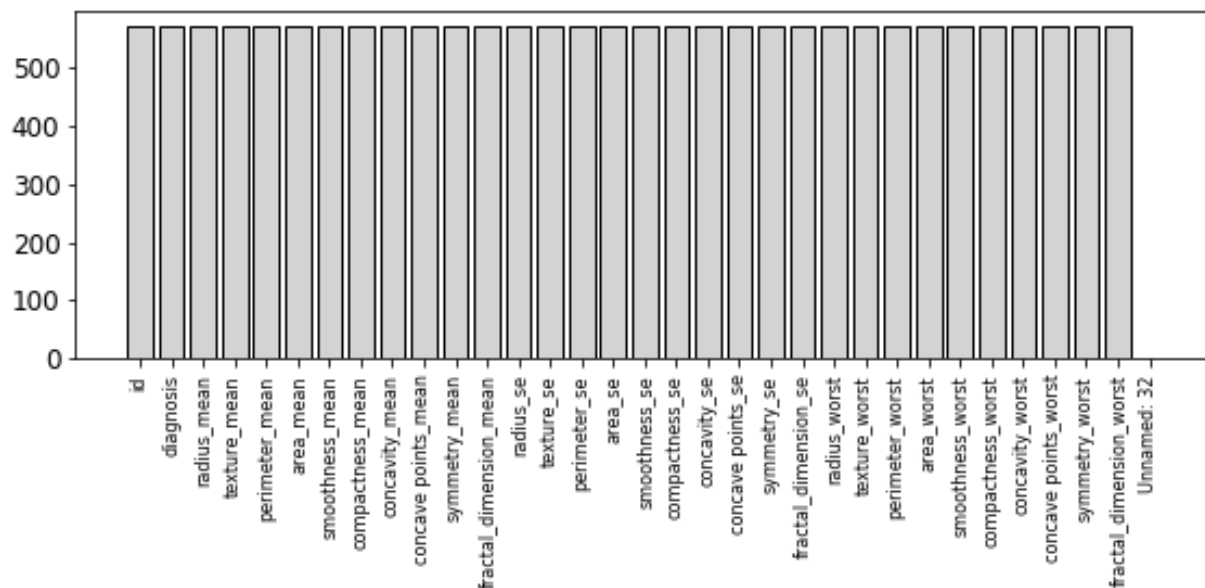


Figure 3.1 Missing Values corresponding to features

- Count-plot

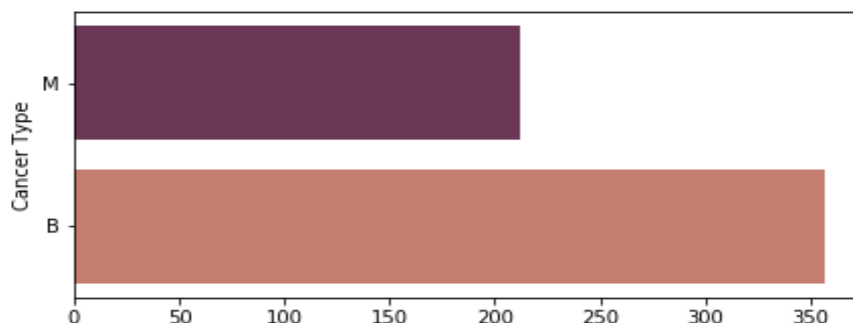


Figure 3.2 Count plot of target variable

- **Bar-plot of Correlation Between Features**

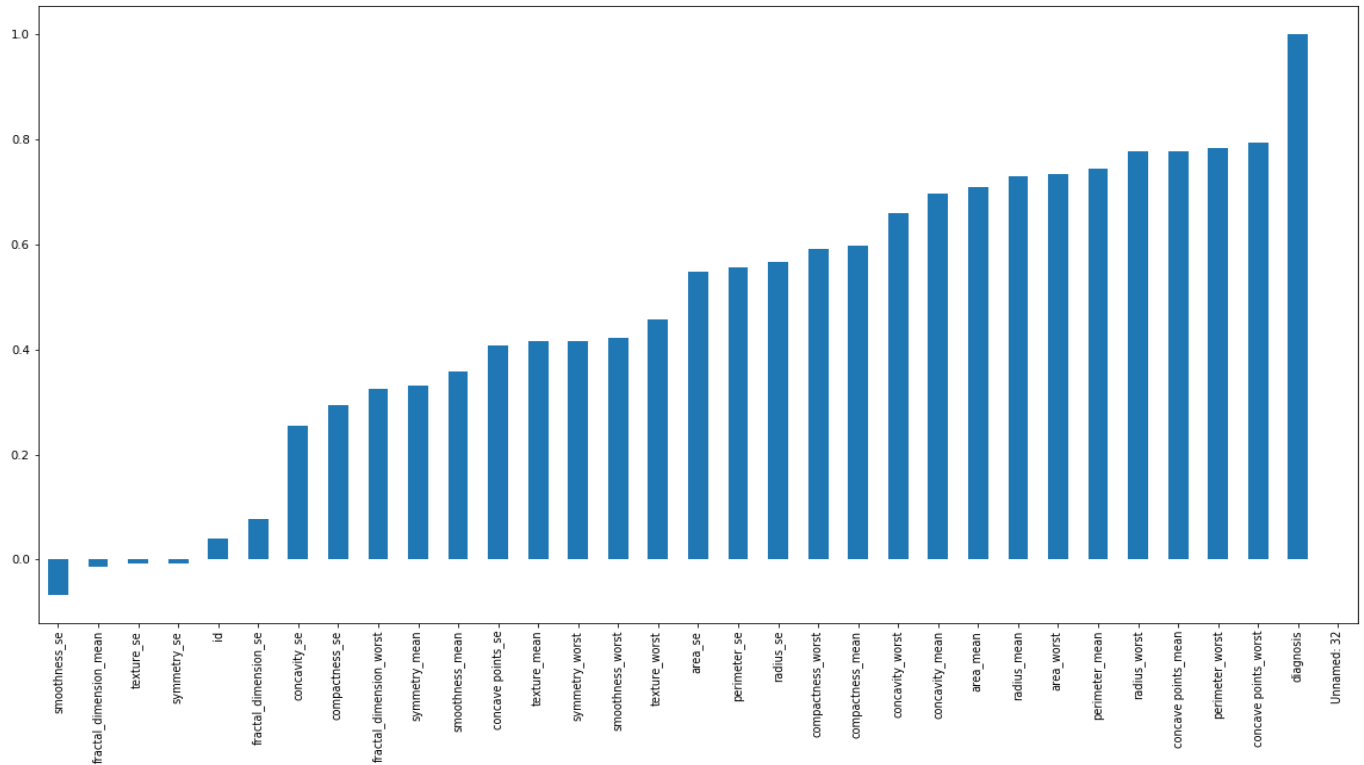


Figure 3.3 Bar Chart of correlation with target variable

- **TSNE Distribution**

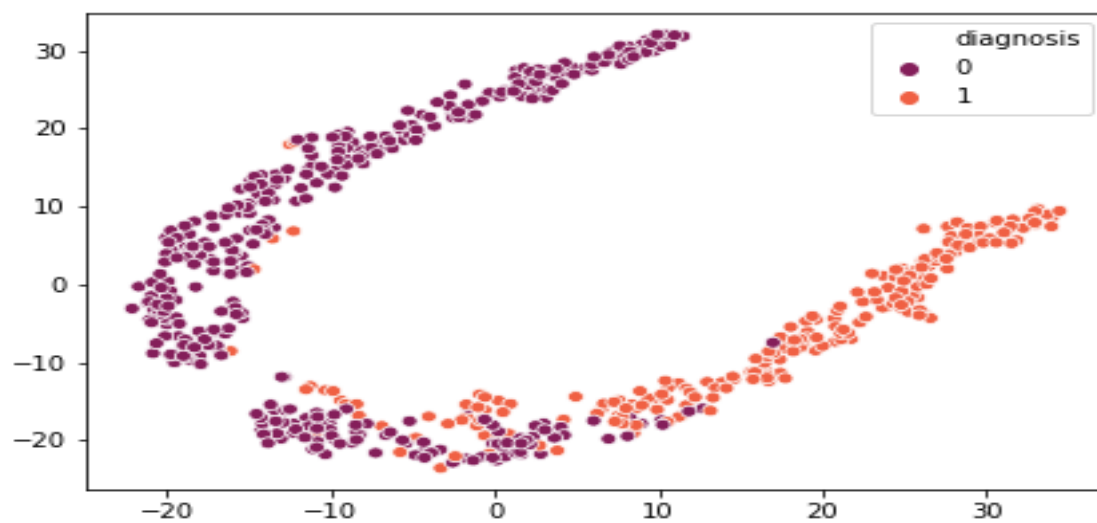


Figure 3.4 TSNE Distribution

- **Heatmap**

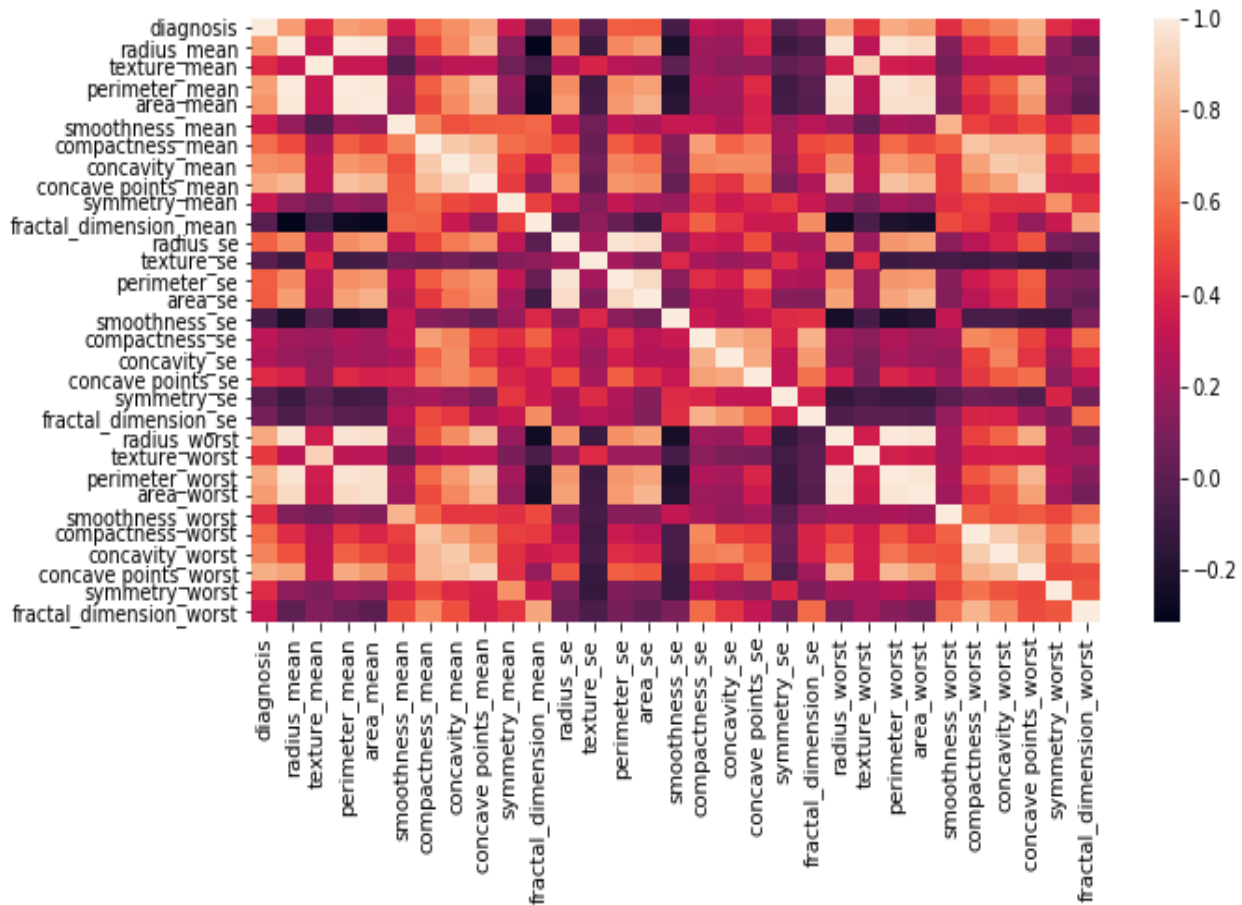


Figure 3.5 Heatmap

From the various visualizations we concluded that the dataset is highly imbalanced in terms of the two sets of classes i.e. Malignant and benign. There are 31 features in the dataset out of which one contains only null values and others are highly correlated.

Data Preprocessing:

Data Preprocessing is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. A dataset can be viewed as a collection of data objects, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities. Data objects are described by a number of features, that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc. Features are often called as variables, characteristics, fields, attributes, or dimensions.

After data visualization we have to do preprocessing of the dataset. As our data is not highly imbalanced so we did not have to use data balancing techniques. After checking that the dataset was imbalanced or not, the next step was to scale the features. Age, Gender, BMI, Steps, Children, Smoker, Charges are the features that were not preprocessed in the dataset we get from Kaggle. We manually scaled both the features using various scaling techniques like Robust Scalar and Standardized Scalar, also there is another feature available in the dataset which is region feature but we have to exclude it because of the correlation problem.

The next thing to be resolved in the preprocessing part was to check that if there are any null values present in the dataset. It is very important to check that there are null values present in the dataset or not. This can be done using the dropna() function. Pandas Data Frame dropna() function is used to remove rows and columns with Null/NaN values. By default, this function returns a new Data Frame and the source Data Frame remains unchanged. It returns the Data Frame from which NA entries has been dropped.

The features of data were in different range so there is a need of scaling as there might be a case of biasness towards high range data in classification problem. So we have done standard scaling of our data.

$$z = \frac{x_i - \mu}{\sigma}$$

After that removing outliers using Z-score method that is remove the data points which come after 3 standard deviation as these are considered as outliers.

Checking correlation between feature helps us to figure out whether the features are related or not for this purpose we plot a heatmap. If absolute value of correlation is very close to one that is features are highly correlated. Negative correlation corresponds to negative dependency of two features that is if one increase other decreases and vice versa for positive correlation.

$$r = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2} \sqrt{\sum(Y-\bar{Y})^2}}$$

Where, \bar{X} - mean of X variable
 \bar{Y} - mean of Y variable

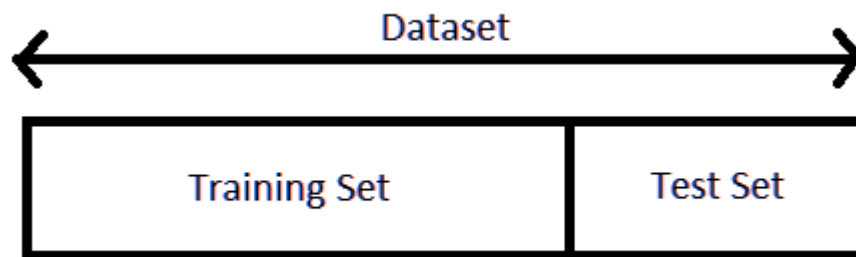
After that we try to find the correlation between these variables. The correlation values between the variables were observed with the help of a Heat-Map. A heatmap is a graphical representation where individual values of a matrix are represented as colors. A heatmap is very useful in visualizing the concentration of values between two dimensions of a matrix. This

helps in finding patterns and gives a perspective of depth. We made the correlation matrix and after that done the feature selection. We determined fifteen features which are necessary to make our model.

After doing the feature selection, we got the data-frame consisting of 15 features.

We also used PCA for feature extraction which performs decorrelation of variables and also rotates them in the direction of increasing variance.

Also for performing model selection we need to split the data into train and test.



PCA

PCA stands for Principal Component Analysis. Principal Component Analysis (PCA) is a linear dimensionality reduction technique that can be utilized for extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space. It tries to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variation. Dimensions are nothing but features that represent the data. For example, A 28 X 28 image has 784 picture elements (pixels) that are the dimensions or features which together represent that image.

One important thing to note about PCA is that it is an Unsupervised dimensionality reduction technique, you can cluster the similar data points based on the feature correlation between them without any supervision (or labels). According to Wikipedia, PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

Principal components are the key to PCA; they represent what's underneath the hood of your data. In a layman term, when the data is projected into a lower dimension (assume three dimensions) from a higher space, the three dimensions are nothing but the three Principal Components that captures (or holds) most of the variance (information) of your data.

Principal components have both direction and magnitude. The direction represents across which principal axes the data is mostly spread out or has most variance and the magnitude signifies the amount of variance that Principal Component captures of the data when projected onto that axis. The principal components are a straight line, and the first principal component holds the most variance in the data. Each subsequent principal component is orthogonal to the

last and has a lesser variance. In this way, given a set of x correlated variables over y samples you achieve a set of u uncorrelated principal components over the same y samples.

The reason you achieve uncorrelated principal components from the original features is that the correlated features contribute to the same principal component, thereby reducing the original data features into uncorrelated principal components; each representing a different set of correlated features with different amounts of variation. Each principal component represents a percentage of total variation captured from the data.

Handling Imbalanced Data

One of the main challenges faced by the utility industry today is electricity theft. Electricity theft is the third largest form of theft worldwide. Utility companies are increasingly turning towards advanced analytics and machine learning algorithms to identify consumption patterns that indicate theft. However, one of the biggest stumbling blocks is the humongous data and its distribution. Fraudulent transactions are significantly lower than normal healthy transactions i.e. accounting it to around 1-2 % of the total number of observations. The ask is to improve identification of the rare minority class as opposed to achieving higher overall accuracy.

Machine Learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced datasets. For any imbalanced data set, if the event to be predicted belongs to the minority class and the event rate is less than 5%, it is usually referred to as a rare event.

SMOTE stands for Synthetic Minority Over Sampling. SMOTE technique is followed to avoid overfitting which occurs when exact replicas of minority instances are added to the main dataset. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original dataset. The new dataset is used as a sample to train the classification models.

Like SMOTE oversampling under-sampling can also be used.

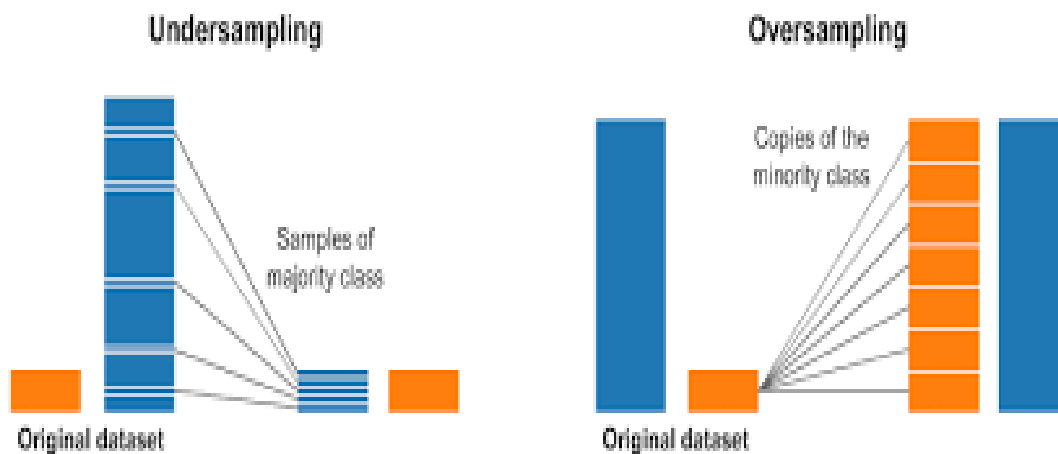


Figure 3.6 Handling Imbalanced Data

Model Selection

We applied different types of classification algorithms on our training set to figure out which comes up with the best accuracy. We applied all the models on our feature selection data and also on our feature extracted data that is PCA.

In this step the dataset is splitted in training and testing data. After splitting the dataset various classification techniques are applied like

- Logistic Regression
- K Nearest Neighbors (KNN)
- Support Vector Classifier
- Random Forest Classifier
- Decision Tree Classifier
- Ada-Boost Classifier
- Naïve Bayes Classifier

Support Vector Machine (SVM)

Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

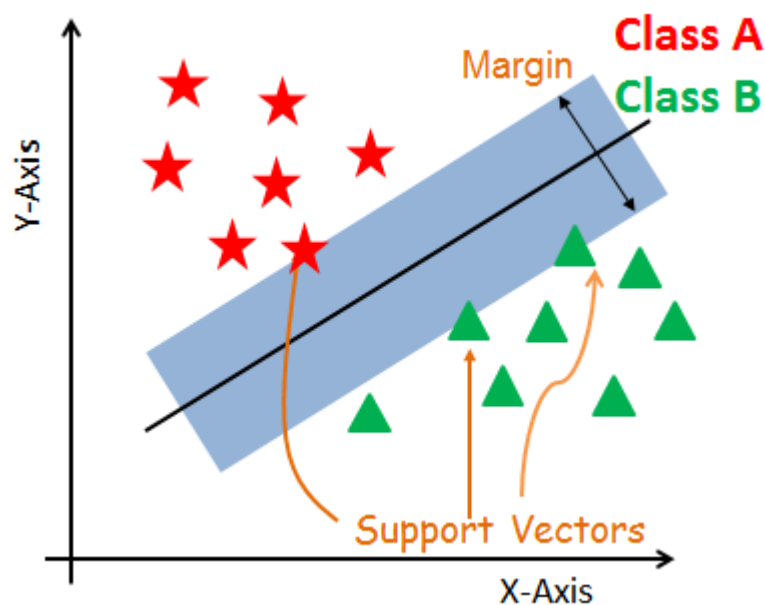


Figure 3.7 SVM

Support Vectors-Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier

Hyperplane-A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin- A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

K-Nearest Neighbor

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and testing phase slower and costlier. Costly testing phase means time and memory. In the worst case, KNN needs more time to scan all data points and scanning all data points will require more memory for storing training data.

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When $K=1$, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P_1 is the point, for which label needs to predict. First, you find the one closest point to P_1 and then the label of the nearest point assigned to P_1 .

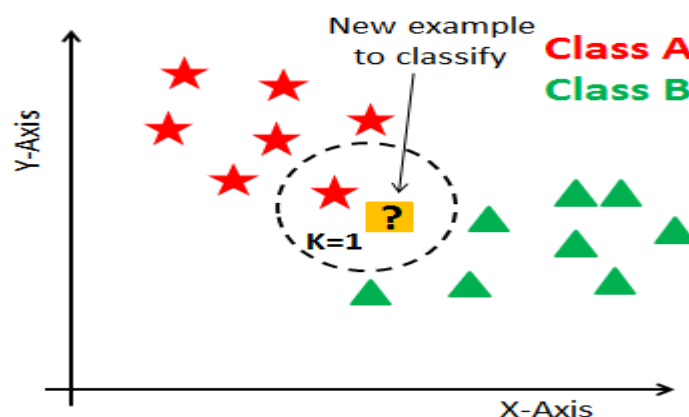


Figure 3.8 KNN

Suppose P1 is the point, for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbours. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance. KNN has the following basic steps:

1. Calculate distance
2. Find closest neighbors
3. Vote for labels

Decision Tree

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

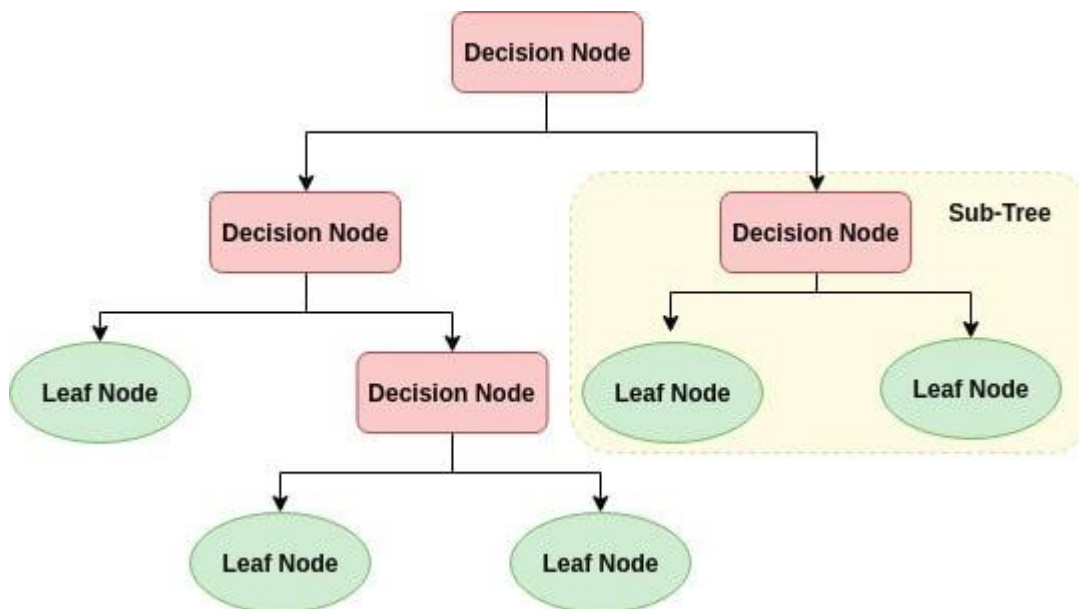


Figure3.9 Decision Tree

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability

distribution assumptions. Decision trees can handle high dimensional data with good accuracy. But decision tree overfits the data.

Random Forest Classifier

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forests has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset. It works in four steps:

Select random samples from a given dataset.

Construct

a decision tree for each sample and get a prediction result from each decision tree.

Perform a vote for each predicted result.

Select the prediction result with the most votes as the final prediction.

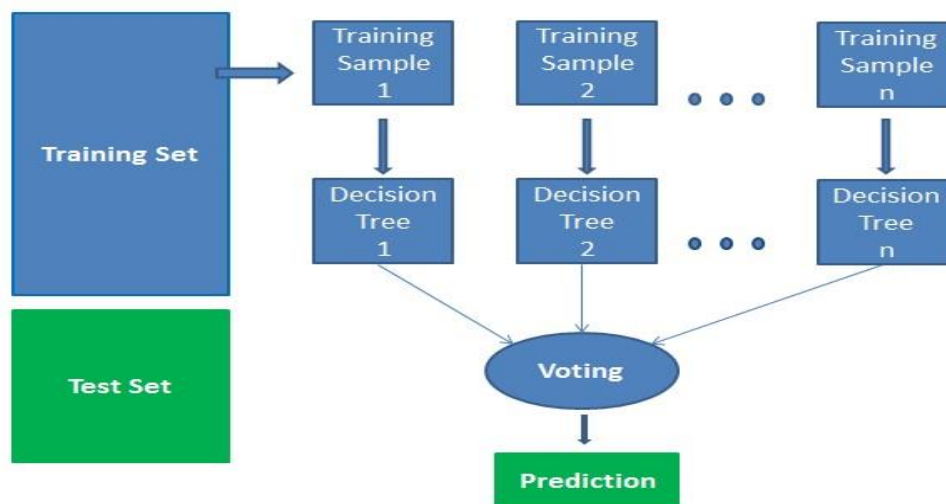


Figure 3.10 Random Forest Classifier

Naïve Bayes

Naive Bayes is the most straightforward and fast classification algorithm, which is suitable for a large chunk of data. Naive Bayes classifier is successfully used in various applications such as spam filtering, text classification, sentiment analysis, and recommender systems. It uses Bayes theorem of probability for prediction of unknown class.

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets. Naive Bayes classifier assumes that

the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

AdaBoost

Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

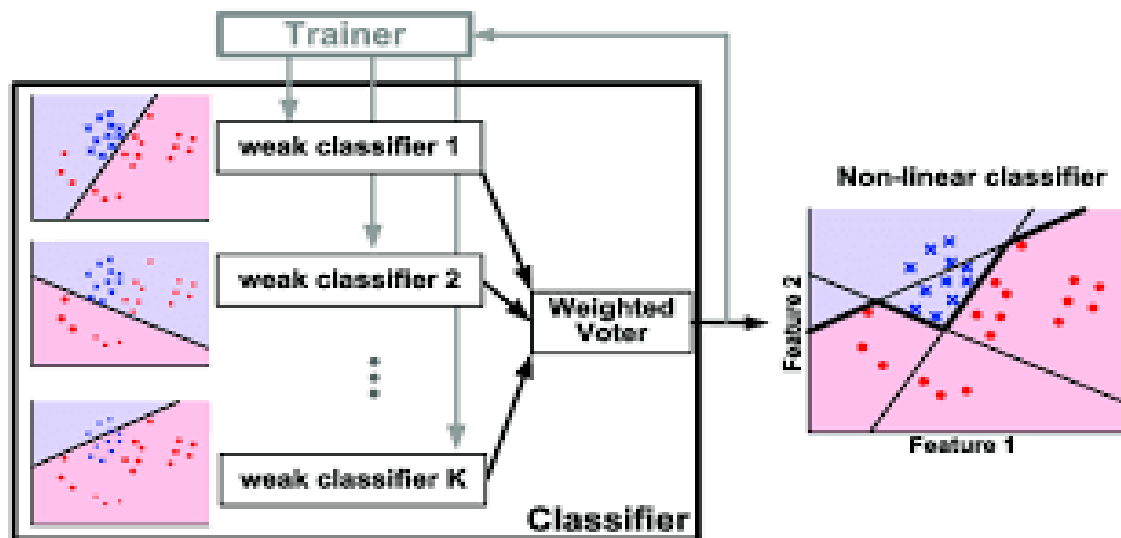


Figure 3.11 AdaBoostClassifier

Logistic Regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence. It is a special case of linear regression where the target variable is categorical in

nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Linear Regression Equation:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

Sigmoid Function:

$$p = 1 / (1 + e^{-y})$$

Apply Sigmoid function on linear regression:

$$p = 1 / (1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)})$$

Properties of Logistic Regression:

- The dependent variable in logistic regression follows Bernoulli Distribution.
- Estimation is done through maximum likelihood.
- No R Square, Model fitness is calculated through Concordance, KS-Statistics.

Model Refining:

In model refining we have done hyperparametric tuning on various models to increase various scores and determined which model performs the best. Hyperparameter tuning is done using GridSearchCV from scikit-learn

Model Evaluation

After applying models what is to be next becomes a major part how to compare the models. Since our problem is a classification problem the metrics which can be used for model evaluation are –

- Confusion Matrix
- F1 Score
- Accuracy
- Precision
- Recall

Deploying Web Services:

In this step we have deployed the best fit model which was giving the best predictions for the dataset with the help of pickle. After that we have integrated the model with a web application

using flask, along with a front-end html page to take values from the user for the various transaction details.

FLASK: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. Applications that use the Flask framework include Pinterest and LinkedIn. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja has become one of the most popular python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.



Figure 3.12 Flask Image

3.1.2 Front End

HTML: Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plainXML, SVG, and XUL.

AZURE-

Steps performed:

- Import the dataset
- Clean Missing values
- Edit Metadata
- Normalize data
- Apply Train Test split
- Model selection SVM in our case
- Train your model
- Score Model
- Evaluate Model
- For better accuracy perform hyperparameter tuning by providing a range of parameters
- Evaluate your model using request response and also using batch execution

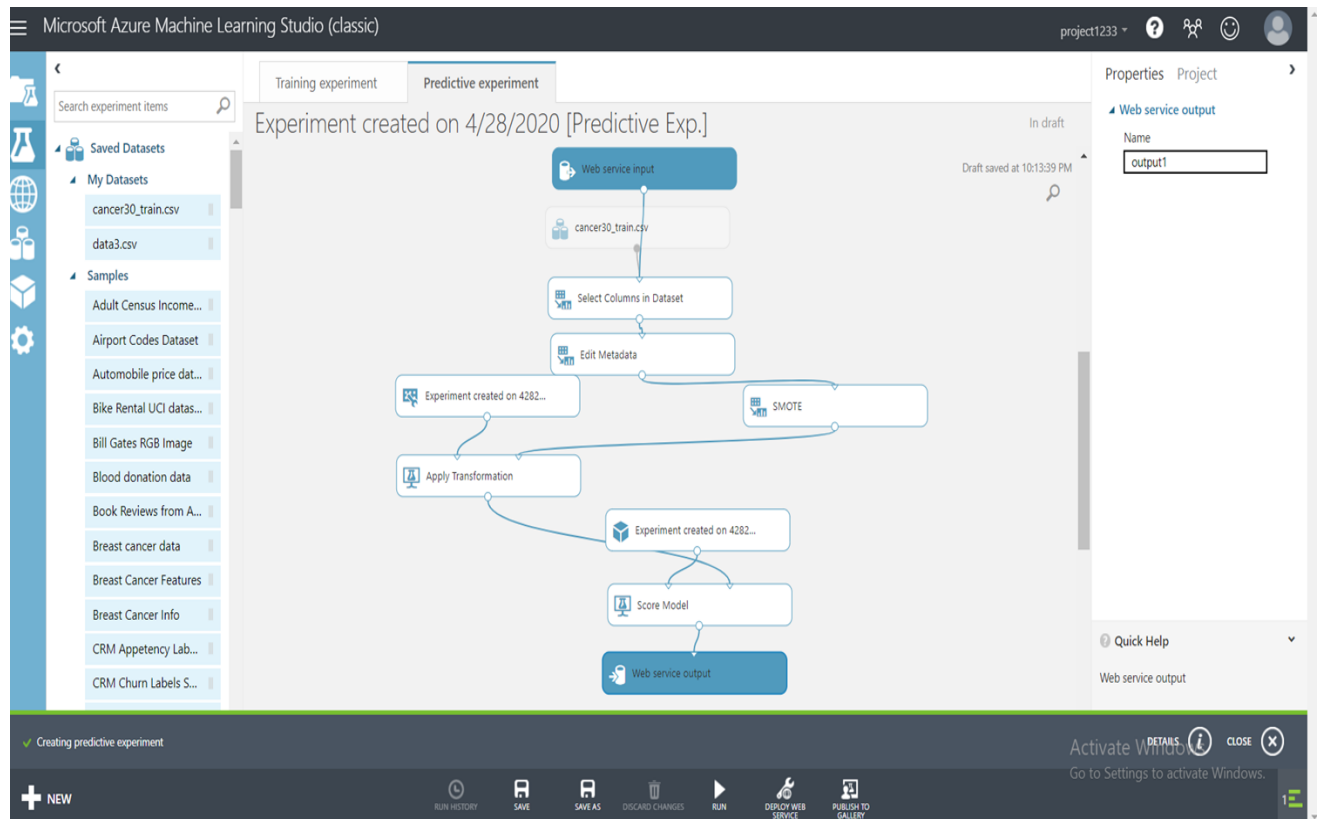


Figure 3.13 Azure Model

3.2 PROJECT CODE:

You can find our complete project code at following GitHub repositories:

<https://github.com/shikha-1902/Breast-Cancer>

CHAPTER 4

Testing

4.1 Testing Objective

Software testing has different goals and objectives. The major Objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.
- To ensure that software under test is 'bug' free before release. It helps validate application's compatibility with the implementation environment, various devices, Operating Systems, user requirements, among other things.

4.2 Testing Principles

Software testing is a process of executing a program with the aim of finding the error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

There are seven principles in software testing:

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence of errors fallacy

Testing shows presence of defects: The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of 17 defects and doesn't talk about the absence of defects. Testing can reduce the number of defects but not removes all defects.

- **Exhaustive testing is not possible:** It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as

exhaustive testing. Exhaustive testing is impossible means the software can never test at every testcases.

- **Early Testing:** To find the defect in the software, early test activity shall be started. The defect detected in early phases of SDLC will very less expensive. For better performance of software, software testing will start at initial phase i.e. testing will perform at the requirement analysis phase.
- **Defect clustering:** In a project, a small number of the module can contain most of the defects. Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.
- **Pesticide paradox:** Repeating the same test cases again and again will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.
- **Testing is context dependent:** Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the Android application.
- **Absence of errors fallacy:** If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it also mandatory to fulfil all the customer requirements.

4.3 Testing Method Used

Testing methodologies are approaches to testing, from unit testing through system testing and beyond. There is no formally recognized body of testing methodologies, and very rarely will you ever find a unified set of definitions. But here are some common methodologies:

- **Acceptance testing:** Also known as acceptance tests, build verification tests, basic verification tests, these are rudimentary tests which prove whether or not a given build is worth deeper testing. The term "smoke test" is a colloquial term -- when machines rebuilt, engineers will power them up and just let them run, looking for smoke as a sign of serious problems.
- **Functional testing:** Functional testing takes a user story or a product feature and tests all of the functionality contained within that feature. For example, in a photo

application like Photoshop, functional testing would cover all the functionality contained within a feature like opening files (resolving file paths, determining appropriate format filters, passing the file path off to the filter) as well as handling errors within that functionality.

- **System Testing:** Testing the project as a collective system. For the Photoshop application, an example would be to open a file in a given format, manipulate that file in various ways, and then output the file. System testing generally combines multiple features into an end-to-end process or scenario.
- **Performance Testing:** Tests an application's performance characteristics, be it file size, concurrent users, or mean-time-to-failure.
- **Security Testing:** A collection of tests focused on probing an application's security, or its ability to protect user assets

4.4 Interface Design

The user need to input all the feature values so that model can predict. Also we used pipeline for a continuous workflow of the data inputted by user.

The screenshot displays a web application interface for predicting cancer. The interface is centered on a purple rectangular panel. At the top of the panel is a white circle containing a black mountain icon. Below the icon, the text 'PREDICT CANCER' is displayed in white. The panel is divided into three sections, each with a title and three input fields:

- Radius:** Three input fields with values 17.99, 1.095, and 25.38.
- Texture:** Three input fields with values 10.38, 0.9053, and 17.33.
- Perimeter:** Three input fields with values 122.8, 8.589, and 184.6.

The browser's address bar shows the URL '127.0.0.1:5000'. The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock.

Figure 4.1 UI Form

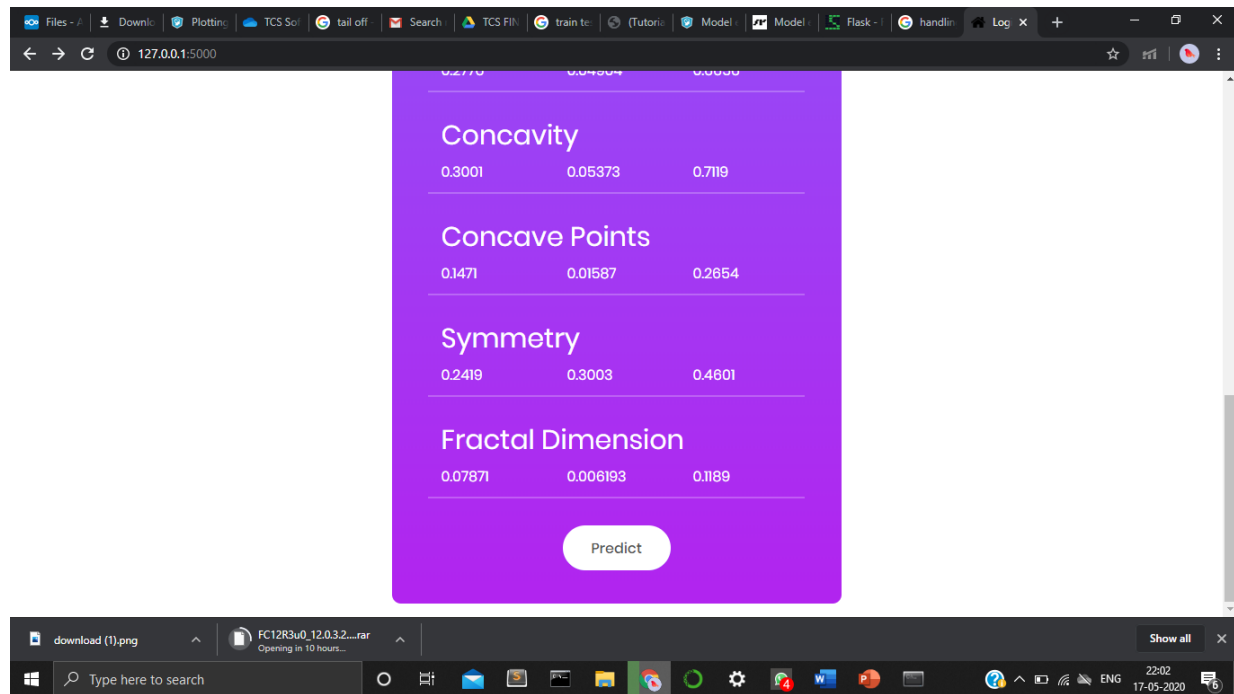


Figure 4.2 UI Form

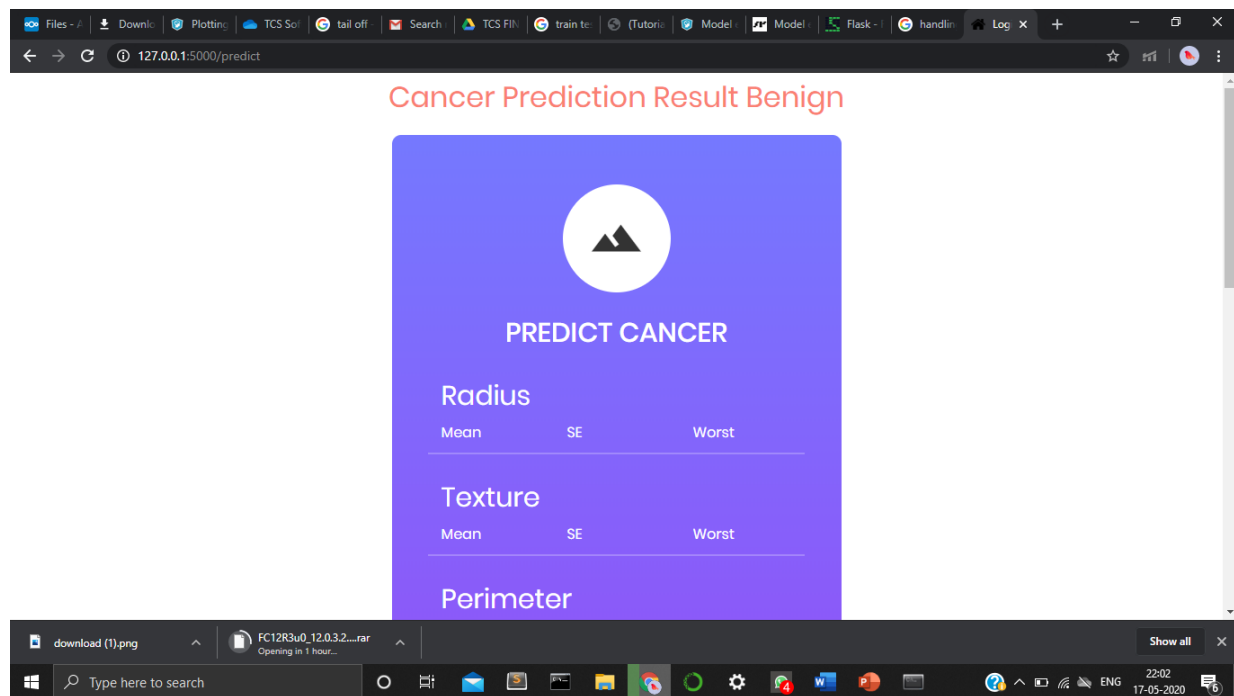


Figure 5.3 Prediction

CHAPTER 5

LIMITATIONS

5.1 Limitations

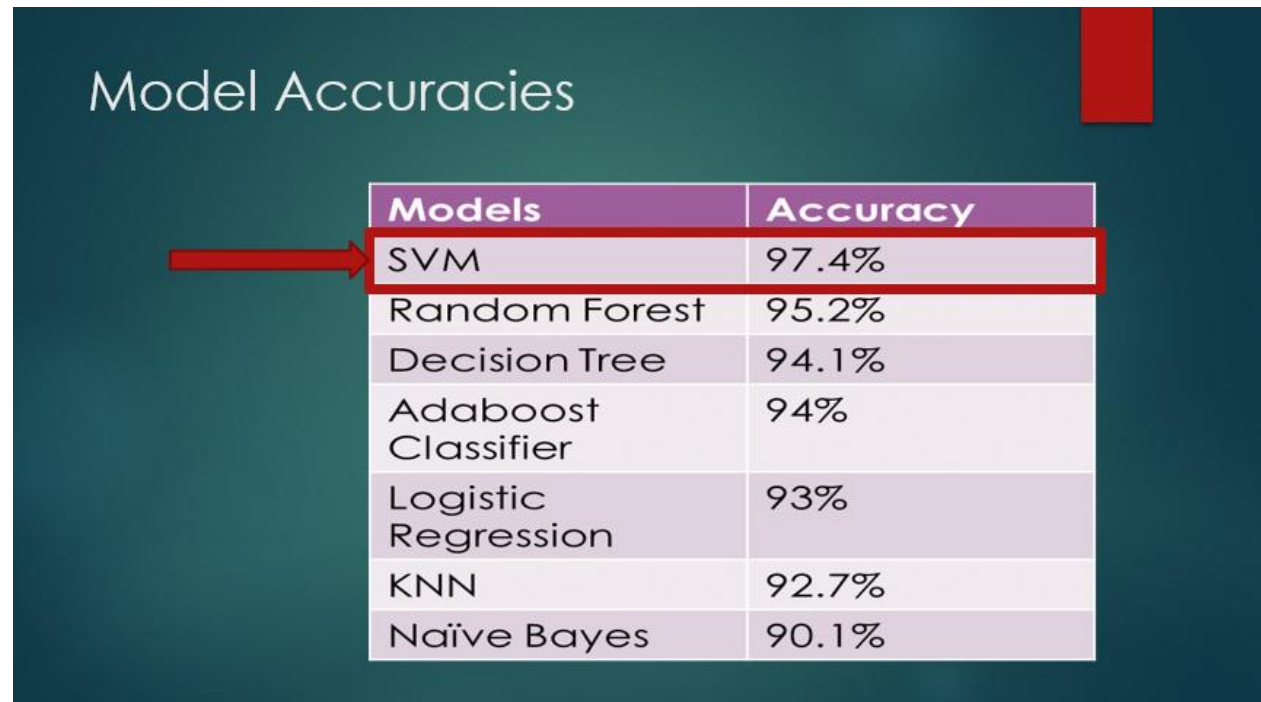
- There are limited datasets available for breast cancer prediction, so it creates a restriction in terms of training the models and getting accurate results for the predictions.
- The quality of data used to train predictive models is equally important as the quantity, in the case of machine learning. The datasets need to be representative and balanced so that they can give a better picture and avoid bias. This is important to train predictive models. Generally, insurers struggle to provide relevant data for training AI models.
- Since we had to work with an already preprocessed data therefore, we are required to put the scaled values as an input to test our Web Application, which is troublesome from a user point of view.
- Since there are large numbers of features which are all very important for classifying an individual will claim insurance or not, it is very difficult to predict or choose which combination of features will be best suited for accurate results.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results and Discussion

We performed a series of operations on the datasets with the algorithms and found some significant results. The models we used are DECISION TREE, RANDOM FOREST, KNN, LOGISTIC REGRESSION, SVM. These results have been listed below.



Models	Accuracy
SVM	97.4%
Random Forest	95.2%
Decision Tree	94.1%
Adaboost Classifier	94%
Logistic Regression	93%
KNN	92.7%
Naïve Bayes	90.1%

Figure 6.1 Prediction Performance of the models

Here, we can see that SVM is the best performing model and the logistic regression is the least accurate model we can also use performance evaluation metrics to cross verify our results.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

After completion of the project, we can summarize the following:

- SVM offers highest accuracy over all other algorithms.
- Naïve Bayes model giving us the least accurate result.
- Breast Cancer Prediction is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which makes sure that the individual will claim his insurance or not.
- Modern technologies are moving extremely fast making their ways into various fields of the business. In this respect, the medical industry does not lack behind the others.
- Thus, the fact that medical sector is actively using data science analytics is not amazing.

CHAPTER 8

FUTURE SCOPE

8.1 FUTURE SCORE

In the age of big data, data mining tasks suffer from plenty of challenges. The results of this study give a view of applying data mining techniques in the healthcare system to help cancer detection. In this research, four data mining models as well as eight hybrid models are tested based on two data sets. Especially, PCA, as a dimension reduction technique, manifests some advantages in terms of prediction accuracy and efficiency. However, there are still some points that can be continually studied in the future. PCA is a linear method, which converts feature space into uncorrelated variables based linear function. In terms of nonlinear feature reduction methods, some other techniques can also be tested, such as -means. Furthermore, the data sets used here are all standard data sets, some raw data set, such as SEER, can be studied in the future.

As far as future scope is concerned we are planning to move in to predict histopathologically using an image rather than the data. So it would be a better approach and it would also help to lead us to learn some deep learning concepts and also we think moving this project to deep learning would provide better accuracy I would say not only say accuracy instead I would say the case where false predictions in case of true should be much less as it's a sensitive matter.

BIBLIOGRAPHY

1. www.google.com
2. www.towardsdatascience.com
3. www.github.com
4. www.w3schools.com
5. www.scikitlearn.org
6. www.geeksforgeeks.com
7. www.w3resource.com
8. www.wikipedia.com