‘Breast Cancer Prediction’ using ML Models

Project Report

Group Members:
Deep Singh 251122489
Mandeep Singh 251122474
Sanket Salunke 251102392

Instructor: Dr. Abdallah Shami
Course: ECE 9039B

CONTENTS

Table of Figures

1 INTRODUCTION

Breast cancer is the most common cancer among Canadian women (excluding non-melanoma skin cancers). It is the second leading cause of death from cancer in Canadian women [1]. More than 80% of deaths due to cancerous cells occurs in underdeveloped and developing countries. Cancer in an essence is a group of diseases, causing an abnormal growth of body cells which lead to development in malign cells, also termed as cancerous cells. As per some statistics of Canadian government, breast cancer accounts for 13 to 25% of overall deaths due to cancer. It is estimated that in year 2019, around 30,000 cases were diagnosed with breast cancer and 5000 in severe cases.

2 PROBLEM DEFINITION AND MOTIVATION

2.1 Problem Definition

Presently there are many ML models, which can be used to for predictions. However, the accuracy of these models is relatively low. Under these circumstances, the chances for false predictions become certainly high. Since, these predictions affect the life of individuals, the error introduced in predication models can be fatal. The goal of this project is to access ML models and detect the cancerous cells with a higher degree of accuracy.

2.2 Motivation

The early diagnosis of Breast cancer can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Accurate classification of benign tumors can assist medical practitioner in devising correct course of treatment. Machine learning (ML) algorithms are widely employed in pattern classification and forecast modelling, because of its unique advantages in critical features detection from complex datasets.

3 RELATED WORK

Feature selection, in machine learning is the process of selecting a subset of the most relevant features from a number of other available feature subsets. Choosing relevant attributes is an integral part of prediction algorithms in machine learning as it plays an important role in creating a more accurate predictive model.
There are various benefits to applying the attribute selection methods such as:

- It is more effective and faster in training the machine learning model.
- It decreases the complexity of a model and makes it easier to interpret.
- It improves the accuracy of an algorithm if the right subset is chosen.
- It reduces overfitting.

Some features may have a complex interrelation between them making it difficult to select the best subset of features. Different approaches have been proposed in this project for breast cancer diagnosis [1-5]. Usually, there are three types of feature selection methods which are: filter, wrapper, and embedded methods.

4 METHODOLOGY

This project will be built on Python programming language. Python platform has an extensive list of libraries and tools for building an ML project. There are number of predefined libraries, which will be employed in various stages of project, from exploratory data analysis, modelling and optimization.

Below figure will gives the brief idea of project's workflow:

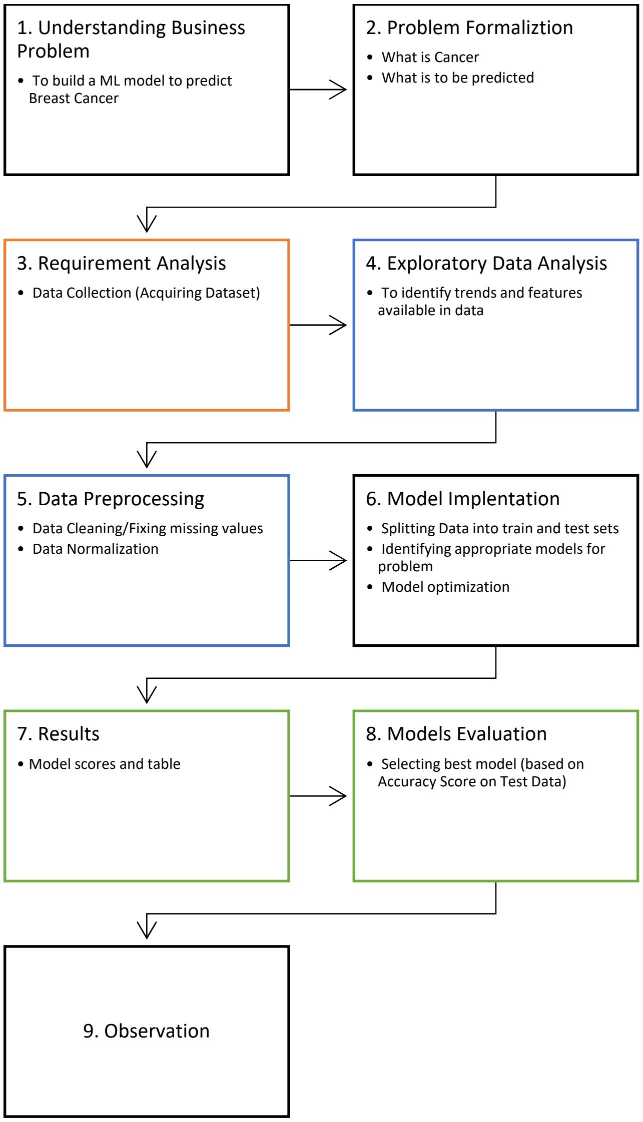

1. Understanding Business Problem
• To build a ML model to predict Breast Cancer
2. Problem Formaliztion
• What is Cancer
• What is to be predicted
3. Requirement Analysis
• Data Collection (Acquiring Dataset)
4. Exploratory Data Analysis
• To identify trends and features available in data
5. Data Preprocessing
• Data Cleaning/Fixing missing values
• Data Normalization
6. Model Implentation
• Splitting Data into train and test sets
• Identifying appropriate models for problem
• Model optimization
7. Results
• Model scores and table
8. Models Evaluation
• Selecting best model (based on Accuracy Score on Test Data)
9. Observation

Figure 1 Workflow

Each activity is discussed as under:

4.1 Understanding Business Problem

The data sample can be analyzed by ML algorithm to detect the presence of malign cell. Using different models, we will be predicting whether a person is diagnosed with cancer or not.

4.2 Problem Formalization

Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions.

In this case, based on the features like radius, texture, perimeter, area etc. given in the data set, model should predict whether it is malignant or benign where malignant means infectious and benign is not harmful. Both Supervised and Unsupervised techniques can be employed to obtain the outcomes. With python library Scikit-Learn (sklearn), different ML algorithms like Logistic regression, KNN, Decision tree, SVM can be used.

4.3 Requirement Analysis

The dataset chosen to address this problem is obtained from Kaggle repository under the heading of Breast Cancer Wisconsin (Diagnostic) Data Set. The dataset is owned by University of Wisconsin, available to students and researchers under open source license. There are 32 features in data set, including

Table 1 Dataset Description

	Feature Name	Feature Description
a.	ID number	

b.	Diagnosis	M = Malign
		B = Benign

remaining features are based on real-valued features from images of cell nucleus, for example:

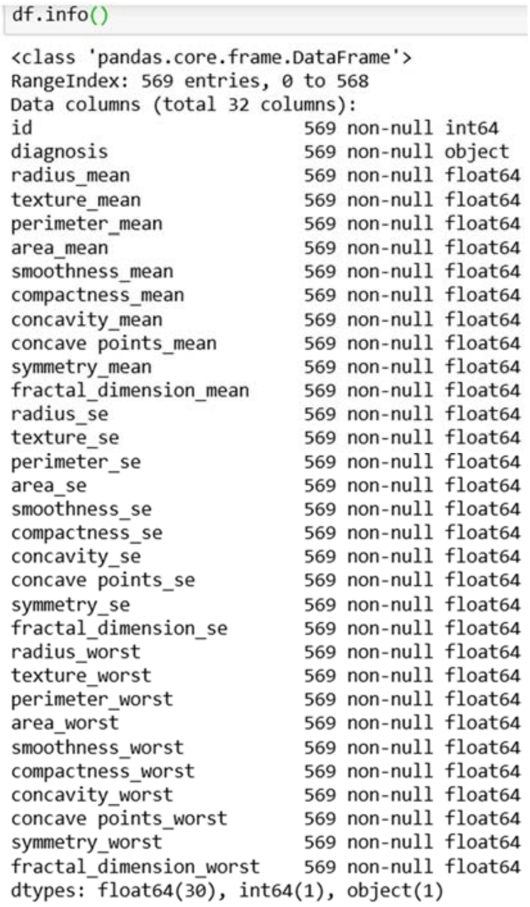
c.	Radius	mean of distances from center to points on the perimeter
d.	Texture	standard deviation of gray-scale values
e.	Perimeter	
f.	Area	
g.	Smoothness	local variation in radius lengths
h.	Compactness	perimeter^2 / area - 1.0
i.	Concavity	severity of concave portions of the contour
j.	Concave points	number of concave portions of the contour
k.	Symmetry	
l.	Fractal Dimension	"coastline approximation" - 1

4.4 Exploratory Data Analysis

In order to summarize main characteristics of dataset EDA techniques are used. During the initial data analysis certain parameters were interrogated, viz.

- Presence of Outliers
- Presence of non-normal
- Association between features
- Pattern in data sets.

For this several analysis techniques/tools are employed, viz. Box Plot, Histogram and Scatterplot



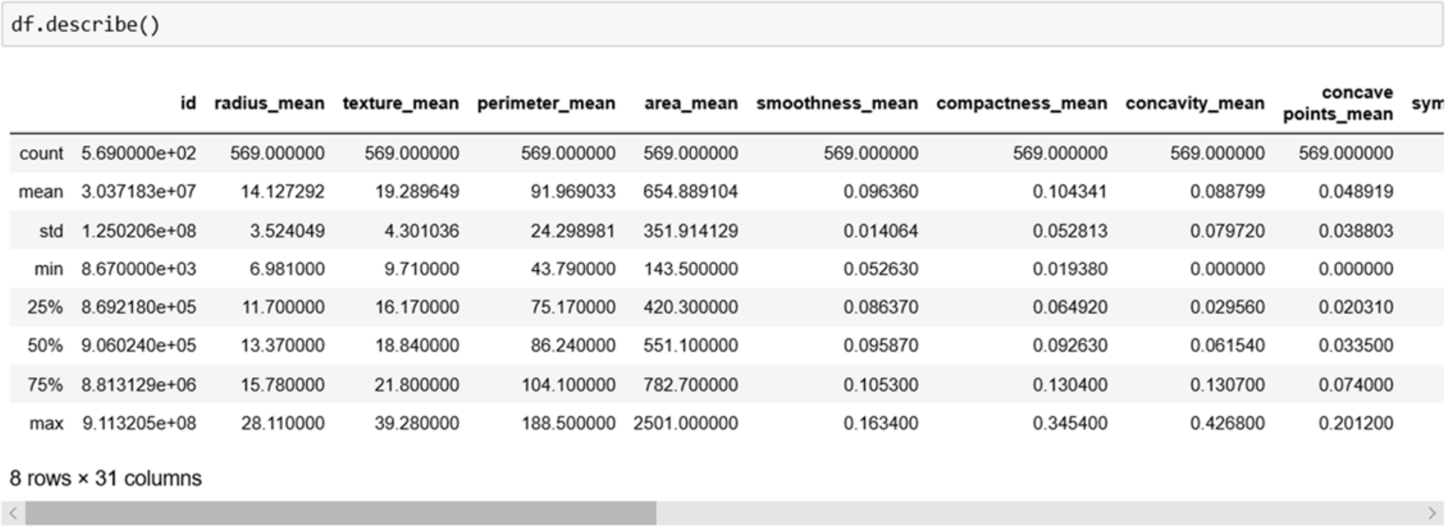
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
dtypes: float64(30), int64(1), object(1)
```

Figure 2

Inferences Figure 2:

- Dataset comprises of 569 observations and 32 features
- All data are in float format except for 'id' which is in integer format
- No variable column has missing value



```
df.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	sym
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	

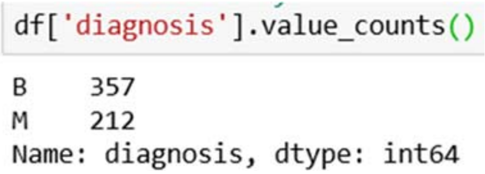
8 rows × 31 columns

Figure 3

Inferences Figure 3:

- Mean value for all features is greater than median value (50%)
- Noticeable difference between 75% and max value for all features, this indicates presence of outliers in data set.

4.4.1 Target Value Analysis



```
df['diagnosis'].value_counts()

B    357
M    212
Name: diagnosis, dtype: int64
```

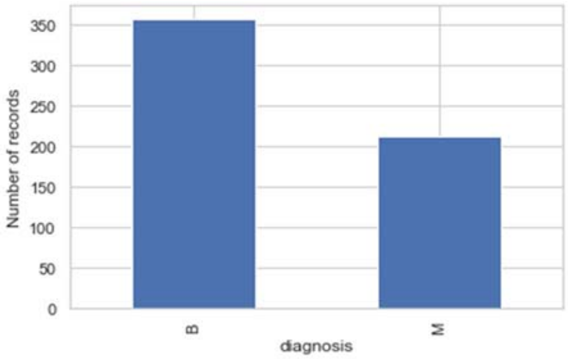


Figure 4 - Target value Analysis

Inferences Figure 4

- Target/dependent feature is categorical in nature with two possible outcomes
- With 357 Benign and 212 Malign outcomes
- Dataset has 60 to 40 ratio for outcomes, which is a balanced ratio.

4.4.2 Visualizations

4.4.2.1 Correlation Matrix

4.4.2.2 Univariate Analysis (Single Variable)

4.4.2.3 Bivariate Analysis (Correlations)

4.4.2.1 Correlation Matrix – is done to inspect relation between varriables

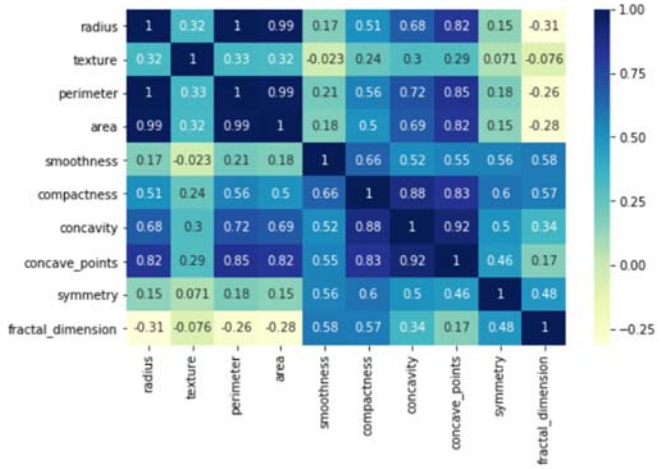


Figure 5 Correlation Matrix - using Heatmap

Inferences Figure 5

- Group 1: (Radius, perimeter, area, concavity and concave_points) have strong positive correlation
- Group 2: (Texture, smoothness, symmetry, fractal_dimension) have relatively low or negative correlation to Group 1 features

4.4.2.2 Univariate analysis – is done by performing frequency counts of different features.

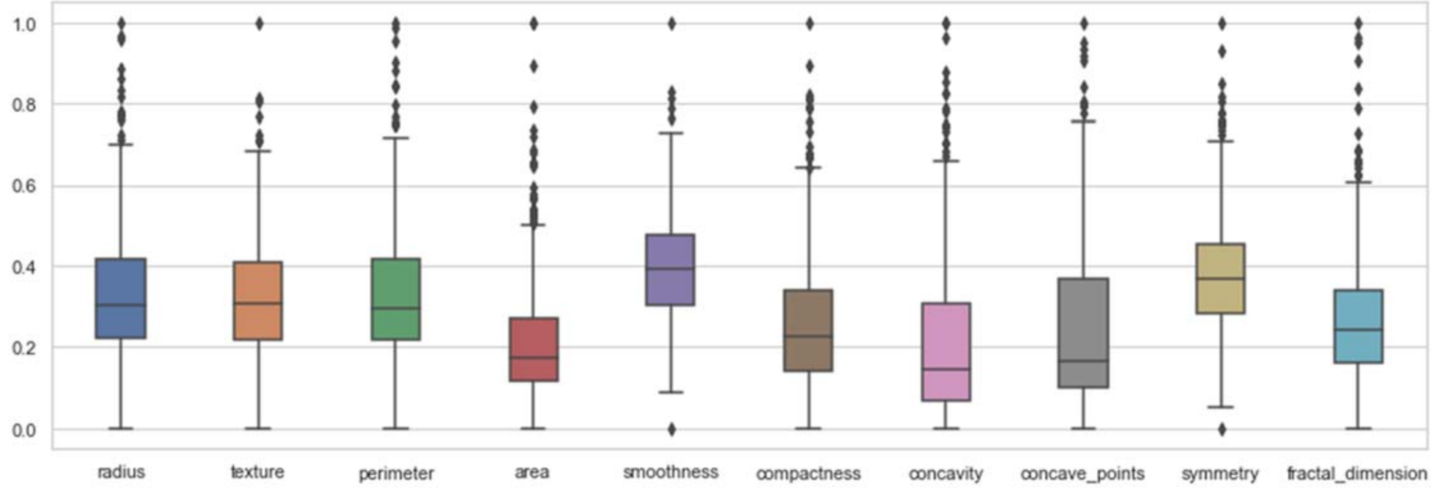


Figure 6 Univariate Analysis – Box plot

Inferences Figure 6:

- Presence of outliers is in 4th quadrant, and rarely in 3rd and 1st as in case of smoothness and symmetry

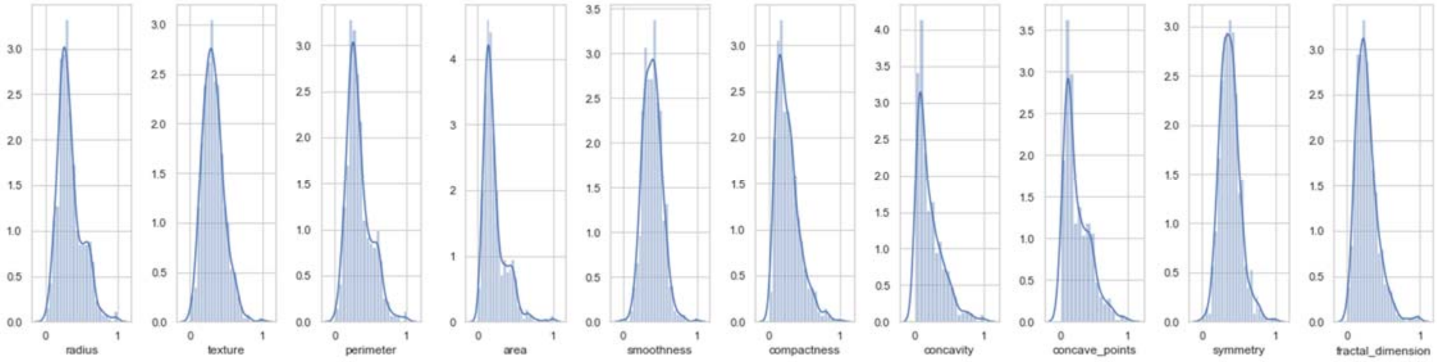
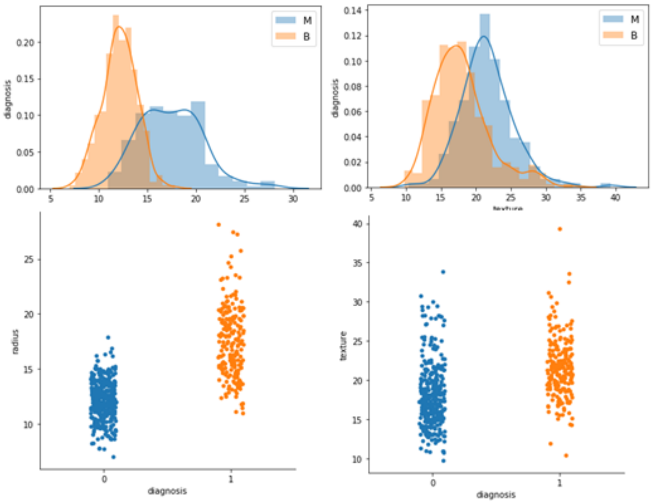


Figure 7 Univariate Analysis - Histogram

Inferences Figure 7:

- Texture, smoothness and symmetry are normally distributed
- Remaining variables are skewed



4.4.2.3 Bivariate analysis

is done to identify association between different features. Plot analysis between (Radius, diagnosis) and (texture, diagnosis) indicates the reason for skewness of radius and normal distribution of texture.

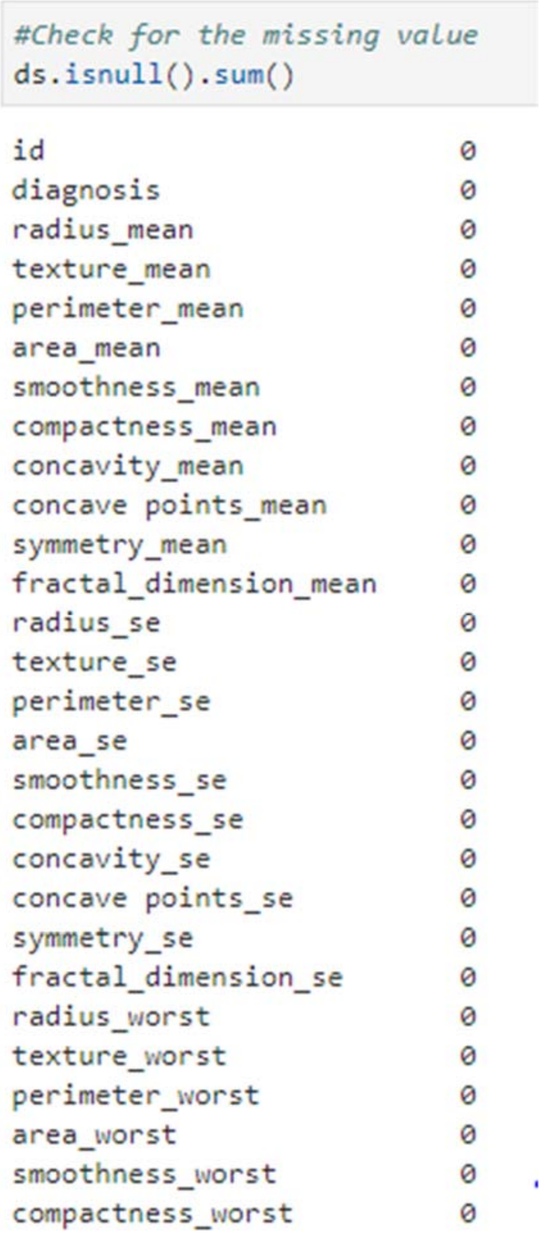
Figure 8 Bivariate Analysis - Histogram and Scatterplot

4.5 *Data Preprocessing***:**

Before starting Data preprocessing, the data quality must be inquired on parameters like:

4.5.1 Completeness (Cases of missing data)*:*

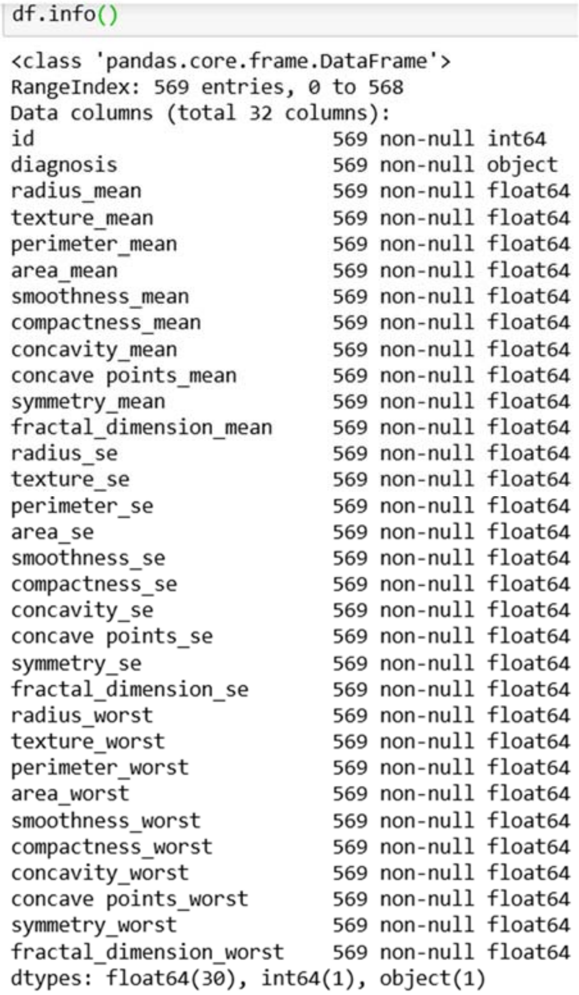
Presence of any NaN value can heavily impact the decision-making ability. There can be various methods to tackle empty or non-null values in dataset, like replacing empty values with mean, median or zero values. In our case the data consistency implies that no such operation is needed.



```
#Check for the missing value
ds.isnull().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
```

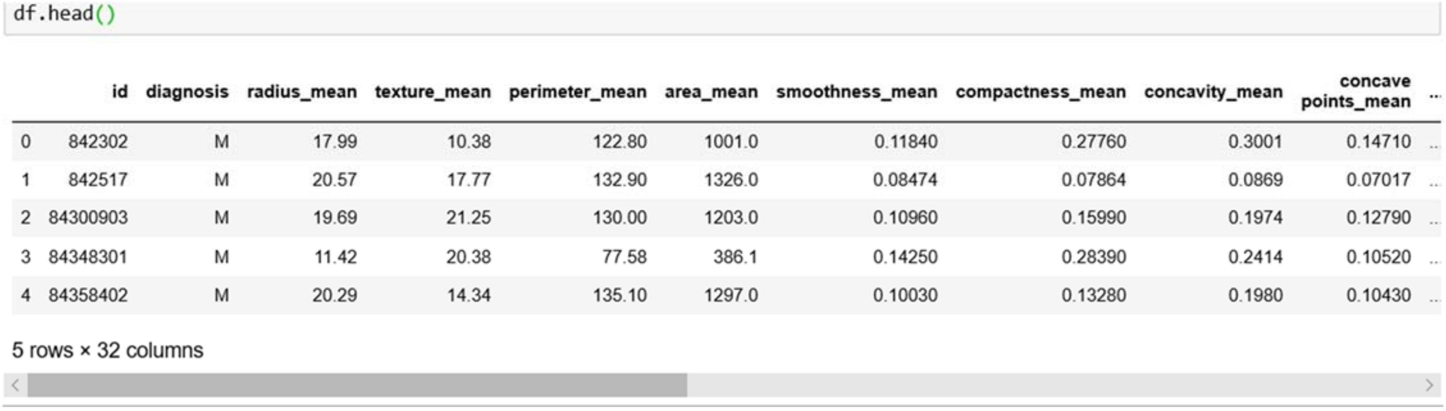
Figure 9



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
dtypes: float64(30), int64(1), object(1)
```

Figure 10



```
df.head()
```

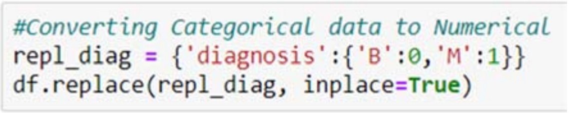
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

5 rows × 32 columns

Figure 11

4.5.2 Presence/Conversion of Categorical Data:

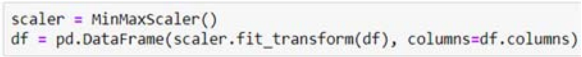
Since most ML model work with numerical values, it is imperative to convert data to suitable forms. To ensure this the categorical feature: diagnosis is mapped to 0s and 1s for benign and malignant.



```
#Converting Categorical data to Numerical
repl_diag = {'diagnosis':{'B':0,'M':1}}
df.replace(repl_diag, inplace=True)
```

Figure 12 Conversion of Categorical Data

Data preprocessing: includes cleaning, instance selection, normalization, transformation, feature extraction and selection. Dataset doesn’t require cleaning, instance selection or transformation as there is no corrupt or missing data. However, there is a requirement for scaling/normalization of data as certain feature are in range of 0-1, while other like area and perimeter are in order of 0-1000.



```
scaler = MinMaxScaler()
df = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
```

Figure 13 Feature Scaling (Range 0-1)

4.6 Modelling (Implementation of models)

4.6.1 KNN Classifier

KNN machine learning model is used for both regression and classification problems. It is based on feature similarity. It is widely used in industry as it requires less calculation time and easy interpreting. KNN is used in our project because in our dataset we have two cases either the cells are malignant or benign. And we are classifying it on the basis of different parameters.

To implement KNN it is very important to choose right value of k as it has very high impact on the models prediction.

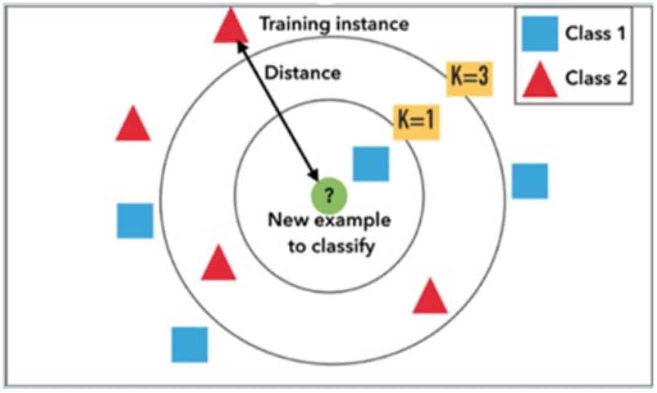


Figure 14 KNN Model

It can be seen in the figure there are two classes and the new data point will be classified depending on the value of k.

If k=1 the only one data point near to new data will be consider and hence it will be classified as class one. But, if k=3 then three nearest data points will be considered and it will be classified as class 2.

Therefore, value of k plays very important role in the model.

Steps to Implement a KNN model:

1. Loaded the data and divided in training and testing set.
2. Changed categorical values of diagnosis to 1 for malignant and 0 for benign.
3. For getting the value of k did parameter tuning by iterate from 1 to 50 with step size of 2
   - Calculated the accuracy, score and confusion matrix for each value of k and stored in an array.
   - Used 'cross_val_score' method to cross validate k value with different folding values.
4. Plotted a graph of accuracy vs. Value of k shown in figure and finalized the best k value for our dataset i.e. k=13

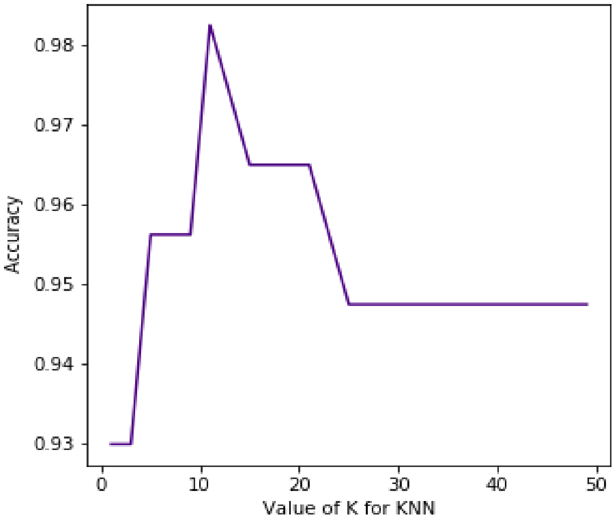


Figure 15 Relation between K and Accuracy of KNN Model

Did performance evaluation of the model by calculated confusion matrix, accuracy and precision.

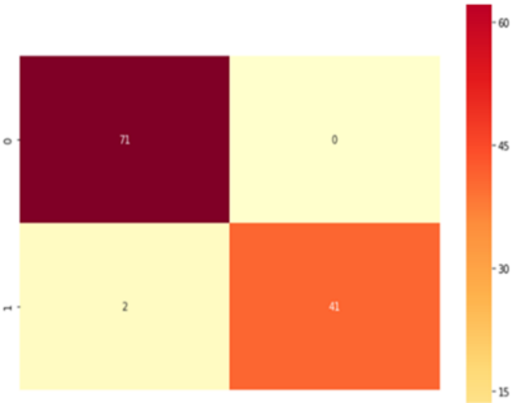


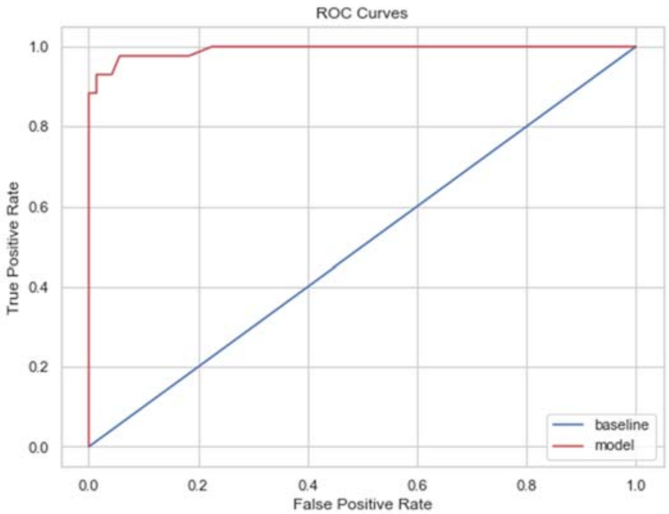
Figure 16 Confusion Matrix for KNN

Confusion matrix is one of the performance evaluation matrixes of machine learning models.
For our model we got 71 True Negative(TN) values which means we predicted no and the patient doesn't have the cancer then, we got 41 True Positive(TP) values that means we predicted yes and the patient do have the cancer. Whereas, our model calculated two False Negative(FN) values means model predicted no but patient actually have the cancer and zero False Positive(FP) values, model predicted yes but actually patient doesn't have the cancer.

Using the values from confusion matrix we calculated accuracy and precision of the model using below formulas.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative) **(0.9824561403508771)**

Precision = True Positive / (True Positive + False Positive) **(1.0)**



4.6.2 Random Forest Classifier

Also known as Random decision forest, is a classification algorithm based on decision trees.

Decision Tree works by forming/splitting the nodes. Each node contains large number of samples of same class. Each class is built by determining the values of feature that can divide the data into class.

Now Random forest combines large number of decision trees, trained on a different set of observation and the final prediction is done by averaging/voting the predictions of each row.

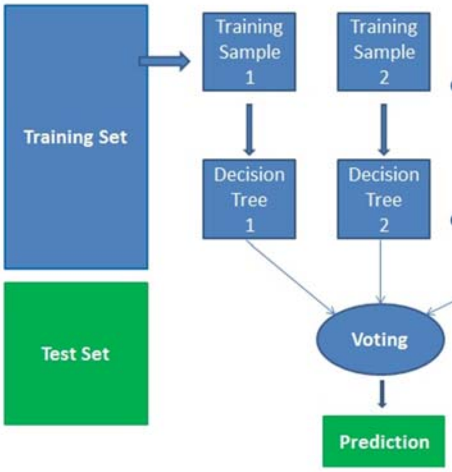
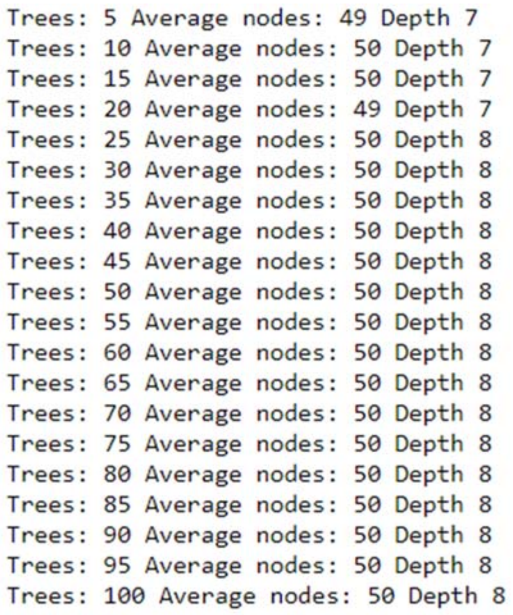


Figure 17 Random Forest

The RF model was tested for number of tree ranging from 1 to 100:



```
Trees: 5 Average nodes: 49 Depth 7
Trees: 10 Average nodes: 50 Depth 7
Trees: 15 Average nodes: 50 Depth 7
Trees: 20 Average nodes: 49 Depth 7
Trees: 25 Average nodes: 50 Depth 8
Trees: 30 Average nodes: 50 Depth 8
Trees: 35 Average nodes: 50 Depth 8
Trees: 40 Average nodes: 50 Depth 8
Trees: 45 Average nodes: 50 Depth 8
Trees: 50 Average nodes: 50 Depth 8
Trees: 55 Average nodes: 50 Depth 8
Trees: 60 Average nodes: 50 Depth 8
Trees: 65 Average nodes: 50 Depth 8
Trees: 70 Average nodes: 50 Depth 8
Trees: 75 Average nodes: 50 Depth 8
Trees: 80 Average nodes: 50 Depth 8
Trees: 85 Average nodes: 50 Depth 8
Trees: 90 Average nodes: 50 Depth 8
Trees: 95 Average nodes: 50 Depth 8
Trees: 100 Average nodes: 50 Depth 8
```

- The average node size was 50 and average depth was 8 for our dataset

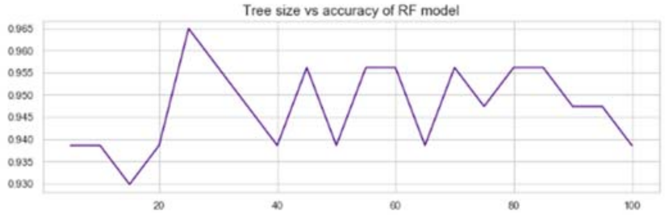
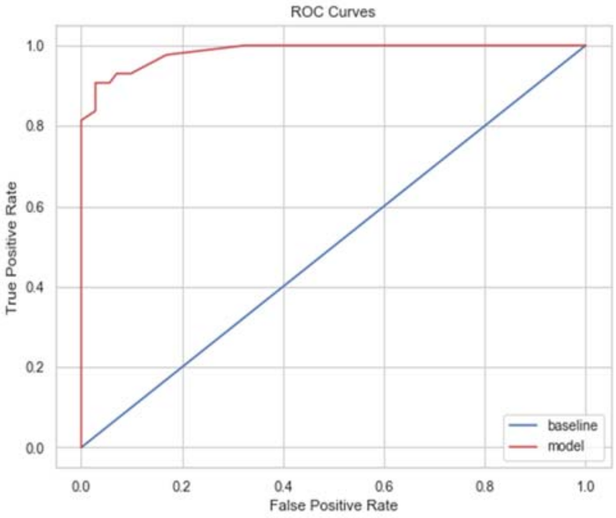
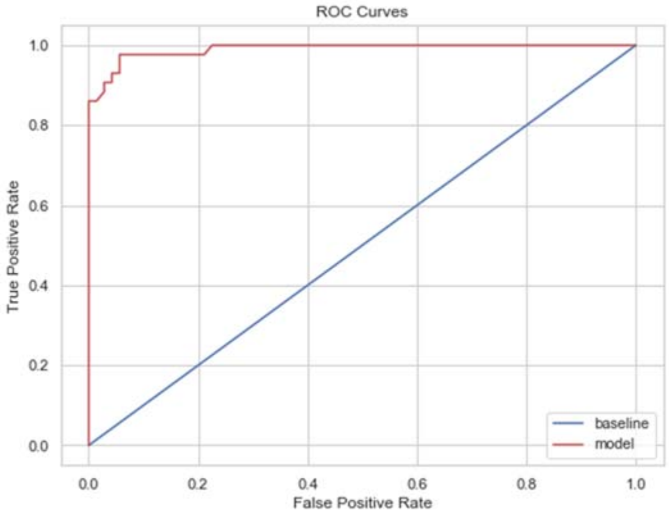


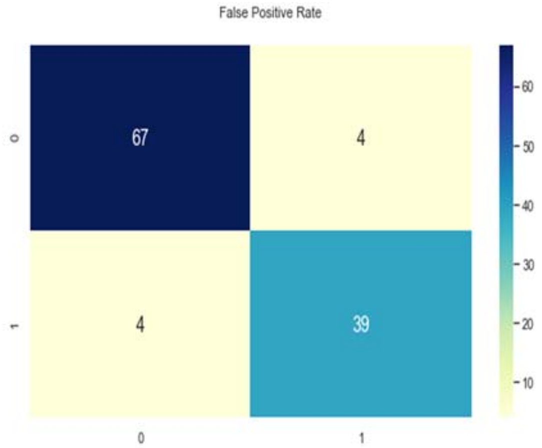
Figure 18 Accuracy vs Tree Size

- Minimum Accuracy: 0.9298245614035088 is for 10 tree nodes



- Maximum Accuracy: 0.9649122807017544 is for 15 tree nodes





4.6.3 SVM Classifier

It is a type of supervised machine learning classification algorithm. It constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.

We have used three different kernels to fit our model i.e. Linear, Polynomial and Gaussian. After applying these three kernels, we have compared them and identified the most accurate kernel for our dataset. Before implementation, we divided the dataset into training and testing set using 'train_test_split' and then changed the categorical values of 'diagnosis' feature to 1 for malignant and 0 for benignant as part of preprocessing.

After this, the model was trained using the three different kernels (Linear, Polynomial and Gaussian) and accuracy and RMSE was calculated for each of them. We have also used cross validation to calculate RMSE for cross validation using 'cross_val_score' with 5 folds to compare for each kernel.

4.6.3.1 Linear Kernel

For the Linear kernel, we have calculated an accuracy of 0.9415204678362573 and RMSE was calculated as 0.2418254167033372

The RMSE for cross validation using 5 folds was obtained as 0.3130596254931703

The following confusion matrix was obtained:

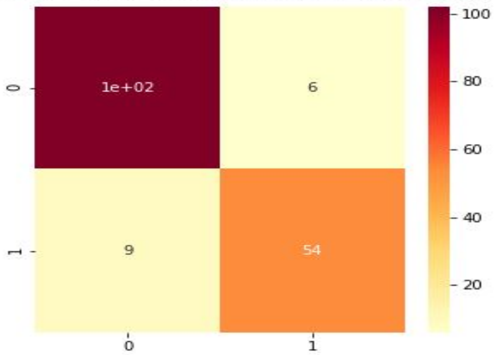


Figure 19 CM for SVM (Linear)

We have obtained 102 values for True Positive, 6 for False Positive, 9 for False Negative and the remaining 54 for True Negative.

4.6.3.2 Polynomial Kernel:

For the Polynomial kernel, we have calculated an accuracy of 0.6666666666666666 and RMSE was calculated as 0.5773502691896257

The RMSE for cross validation using 5 folds was obtained as 0.6576473218982952

The following confusion matrix was obtained:

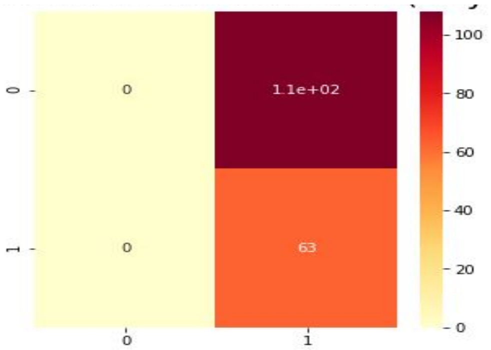


Figure 20 CM for SVM (Polynomial)

We have obtained 0 values for True Positive, 102 for False Positive, 0 for False Negative and the remaining 63 for True Negative.

4.6.3.3 Gaussian Kernel

For the Gaussian kernel, we have calculated an accuracy of 0.7017543859649122 and RMSE was calculated as 0.5461186812727502

The RMSE for cross validation using 5 folds was obtained as 0.5695045592776011

The following confusion matrix was obtained:

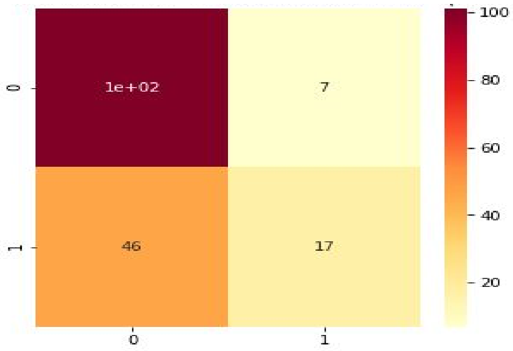


Figure 21 CM for SVM (Gaussian)

We have obtained 102 values for True Positive, 7 for False Positive, 46 for False Negative and the remaining 17 for True Negative.

4.6.3.4 Comparing SVM Kernels:

Table 2

	Linear	Polynomial	Gaussian
Accuracy	0.9415204678362573	0.6666666666666666	0.7017543859649122
RMSE	0.2418254167033372	0.5773502691896257	0.5461186812727502
RMSE for Cross Validation	0.3130596254931703	0.6576473218982952	0.5695045592776011

In the above table, it can be clearly seen that the Linear kernel gave us the best results as accuracy for Linear kernel was the best among the three kernels and if we look at the RMSE and RMSE for Cross Validation, Linear kernel has the least values which makes it the best among other kernels. On the other hand, the polynomial kernel gave us the worst results having an accuracy of 0.66 and higher values of RMSE and RMSE for Cross Validation as compared to Linear and Gaussian Kernels.

4.7 *Model Evaluation*

Parameter tuning for all the models has been done to calculate optimal values of parameters of each model. This resulted in implementing the models with higher degree of accuracy, which was one of the goals of our project

Models trained at these optimal values are then compared. After comparing three algorithms it can be concluded that KNN performs better than SVM and Random forest as it has highest accuracy.

The following table shows the comparison of different models applied to dataset:

Table 3

MODEL	ACCURACY	CONFUSION MATRIX
KNN	0.9824561403508771	[[71 0] [2 41]]
SVM (Linear)	0.9415204678362573	[[71 0] [4 39]]
SVM (Polynomial)	0.6666666666666666	[[65 6] [4 39]]
SVM (Gaussian)	0.7017543859649122	[[71 0] [3 40]]
Random Forest	0.9649122807017544	[[67 4] [3 40]]

5 FINAL OBSERVATIONS AND CONCLUSION

The aim of the project was to design and develop a machine learning model to predict the breast cancer by classifying cells as benign or malignant depending on the features of cells. The object of the project has been accomplished, we implemented three classification models. Additionally, our findings include:

- It is important to choose the right combinations of parameter values to get higher accuracy. By doing parameter tuning we gained insight of how to increase the efficacy of predicted output.
- Also, the key is to select right features from the dataset. Finding right features is done by using ranking and correlation of features used for the predication of the diseases.
- Random forest is a robust learning model which can provide highly accurate predictions, however it is the most memory and time consuming model when compared to other two. Further scope of work is required to address these issue for this model, which may provide even a better predictions.
- SVM perform fastest among the tree models, however it lacks when metrics of all three models are compared.
- For our dataset, KNN performed better than SVM and Random forest. In term of accuracy it provided the best results. Considering Time and memory consumptions this model performed fairly well.

REFERENCES

1. M. F. Akay, “Support vector machines combined with feature selection for breast cancer diagnosis,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3240–3247, 2009.
2. Y. Sun, C. F. Babbs, and E. J. Delp, “A comparison of feature selection methods for the etection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm,” in *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 6532–6535, Shanghai, China, September 2005.
3. B. Zheng, S. W. Yoon, and S. S. Lam, “Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.
4. E. Aličković and A. Subasi, “Breast cancer diagnosis using GA feature selection and Rotation Forest,” *Neural Computing and Applications*, vol. 28, no. 4, pp. 753–763, 2017.
5. M. Banaie, H. Soltanian-Zadeh, H.-R. Saligheh-Rad, and M. Gity, “Spatiotemporal features of DCE-MRI for breast cancer diagnosis,” *Computer Methods and Programs in Biomedicine*, vol. 155, pp. 153–164, 2018.
6. https://stackoverflow.com/questions/19629331/python-how-to-find-accuracy-result-in-svm-text-classifier-algorithm-for-multil