Machine Learning in Breast Cancer Test-Result Visualisation

A Project Report Submitted for the partial fulfillment of the Degree of
Bachelor of Technology in Information Technology
Under
Maulana Abul Kalam Azad University of Technology, West Bengal
(Formerly known as West Bengal University of Technology)

Project Submitted by:

Saurabh Kumar
Roll No: 10400218048, Registration No: 181040110274 of 2018-19

Academic Year: 2018 – 2022

Under the guidance of
Prof. Subhabrata Sengupta

Department of Information Technology
Institute of Engineering & Management, Kolkata
Gurukul, Y-12, Block – EP, Sector -V, Salt Lake Electronics Complex,
Kolkata – 700091, West Bengal, India

May, 2022

INSTITUTE
OF ENGINEERING & MANAGEMENT
Salt Lake Electronics Complex, Kolkata-700 091, WB, INDIA

Phone : (033) 2357-2969/2059/2995
(033) 2357-8189/8908/5389
Fax : 91-33-2357-8302
E-mail : director@iemcal.com
Website : www.iem.edu.in

CERTIFICATE OF APPROVAL

TO WHOM IT MAY CONCERN

This is to certify that the project report titled "**Machine Learning in Breast Cancer Test-Result Visualisation**", under the research domain "Machine Learning", is submitted by Saurabh Kumar (Roll No: 10400218048, Registration Number: **181040110274** of **2018-19**), students of **Institute of Engineering & Management** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology** in **Information Technology**, is a bonafide work carried out under the supervision of **Prof. Subhabrata Sengupta**, and under the guiding light of **Prof. Moutushi Singh, The Head Of The IT Department, Dr. Satyajit Chakrabarti,Director, IEM**, during the final year of the academic session of **2018-2022**. The content of this report has not been submitted to any other university or institute for the award of any other degree. It is further certified that the work is entirely done by him and the performance has been found to be satisfactory.

Prof. Subhabrata Sengupta

Assistant Professor
Department of Information Technology
Institute of Engineering&Management

Dr. Moutushi Singh

Head of the Department
Department of InformationTechnology
Institute of Engineering&Management

20/05/2022

prof. Dr. Arun Kumar Bar
Principal
Institute Of Engineering And Management

DECLARATION FOR CONDUCTING THE PROJECT WORK

I, Saurabh Kumar, student of B.Tech in the Department of Information Technology, Institute of Engineering & Management has submitted the project report in partial fulfillment of the requirements to obtain the above noted degree. We declare that I have done this project work myself, under the guidance of my mentor, Prof. Subhabrata Sengupta, and based on this work I have written this report and have acknowledged the applicable resources from where the inspiration of acknowledgement is procured with a note of relevant publications in this domain I've been a part of in the due course of time.

Saurabh Kumar _____ 20/05/2022

Saurabh Kumar Date

Section: A, Class Roll no. 24
University Roll no- 10400218048
Registration no- 181040110274 of 2018-19
Department of Information Technology, IEM

INDEX

ABSTRACT

Breast cancer is one of the leading cause for the death of women. In women Breast cancer is treated as the most significant issue. Breast cancer has now overtaken lung cancer as the most commonly diagnosed cancer in women worldwide. India has witnessed 30% of the cases of breast cancer during the last few years and it is likely to increase. Breast cancer in India accounts that one woman is diagnosed every two minutes and every nine minutes, one woman dies. Early detection and diagnosis can save the lives of cancer patients. Early diagnosis of this helps to prevent the cancer. If breast cancer is detected in early stage, then Survival rate is very high.

This project presents a novel method to detect breast cancer by employing techniques of Machine Learning. Machine Learning methods are effective ways to classify data. Especially in the medical field, where these methods are widely used in diagnosis and analysis for decision making. In this project, Data Visualization and performance comparisons between different machine learning algorithms: Support Vector Machine (SVM), Logistic Regression, Decision Tree, Naive Bayes (NB), K Nearest Neighbours (k-NN), XGboost and Random Forest conducted on UCI ML Breast Cancer Wisconsin (Diagnostic) dataset.

The main objective of this project is to evaluate the accuracy in the classification of data in terms of efficiency and effectiveness of each algorithm in terms of accuracy, precision, sensitivity and specificity. Our aim is to review various Techniques to detect early, efficiently and accurately Using Machine Learning.

INTRODUCTION

Breast cancer is one of the most common cancer in women and the second leading cause of women's cancer death. Despite the lack of effective treatment, the low accuracy of diagnosis is also a major cause of the high incidence and mortality of breast cancer. Mammography was a traditional method used for diagnosing breast cancer. According to UCHealth's report, only 78% of breast cancer can be accurately diagnosed by mammography . Many cases such as doctors' negligence or incompetence in addition to a mammography error may also result in a late diagnosis or misdiagnosis, which can be considered a cause of breast cancer death. In the long term, early-stage diagnosis could significantly increase the survival rate of breast cancer, therefore, it is important to improve the accuracy of breast cancer diagnosis. Machine learning has been applied in medical diagnosis since from it's existance. In order to increase the accuracy of breast cancer diagnosis, we aim to use machine learning models and choose the model with higher performance. Breast Cancer Wisconsin is a widely used dataset provided by UC Irvine machine learning repository. In this project,I will train Different models using this dataset. I will then use seven different methods including Support Vector Machine (SVM), Logistic Regression, Decision Tree, Naive Bayes (NB), K Nearest Neighbours (k-NN), XGboost and Random Forest Regression method to predict whether the case is benign or malignant and will try to find the best model with high accuracy as much as we can acchieve through different feature scalling as well as data scalling processes.

Language in project: Python

Modules in language: Sklearn, Scikit learn,XgBoost, Pandas, Numpy, Matplotlib, Seaborn, Pickle

Predefined Algorithms used: Support Vector Machine (SVM), Logistic Regression, Decision Tree, Naive Bayes (NB), K Nearest Neighbours (k-NN), XGboost and Random Forest Regression

Dataset Source: UCI ML Breast Cancer Wisconsin (Diagnostic) dataset.

DATASET AND FEATURES

The dataset we used is Breast Cancer Wisconsin dataset which is a widely used dataset in study. It contains 569 instances with 31 features and 2 classes (benign and malignant.) The class distribution is as follows: 357 benign and 212 malignant. The features are:

#	Column	Non-Null Count	Dtype
0	mean radius	569 non-null	float64
1	mean texture	569 non-null	float64
2	mean perimeter	569 non-null	float64
3	mean area	569 non-null	float64
4	mean smoothness	569 non-null	float64
5	mean compactness	569 non-null	float64
6	mean concavity	569 non-null	float64
7	mean concave points	569 non-null	float64
8	mean symmetry	569 non-null	float64
9	mean fractal dimension	569 non-null	float64
10	radius error	569 non-null	float64
11	texture error	569 non-null	float64
12	perimeter error	569 non-null	float64
13	area error	569 non-null	float64
14	smoothness error	569 non-null	float64
15	compactness error	569 non-null	float64
16	concavity error	569 non-null	float64
17	concave points error	569 non-null	float64
18	symmetry error	569 non-null	float64
19	fractal dimension error	569 non-null	float64
20	worst radius	569 non-null	float64
21	worst texture	569 non-null	float64
22	worst perimeter	569 non-null	float64
23	worst area	569 non-null	float64
24	worst smoothness	569 non-null	float64
25	worst compactness	569 non-null	float64
26	worst concavity	569 non-null	float64
27	worst concave points	569 non-null	float64
28	worst symmetry	569 non-null	float64
29	worst fractal dimension	569 non-null	float64
30	target	569 non-null	float64

DATA-VISUALLISATION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

For Data-Visuallisation Seaborn provides a great tool. To create Seaborn plots, one must import the Seaborn library and call functions to create the plots.

Importantly, Seaborn plotting functions expect data to be provided as Pandas DataFrames. This means that if you are loading your data from CSV files, you must use Pandas functions like **read_csv()** to load your data as a DataFrame. When plotting, columns can then be specified via the DataFrame name or column index.

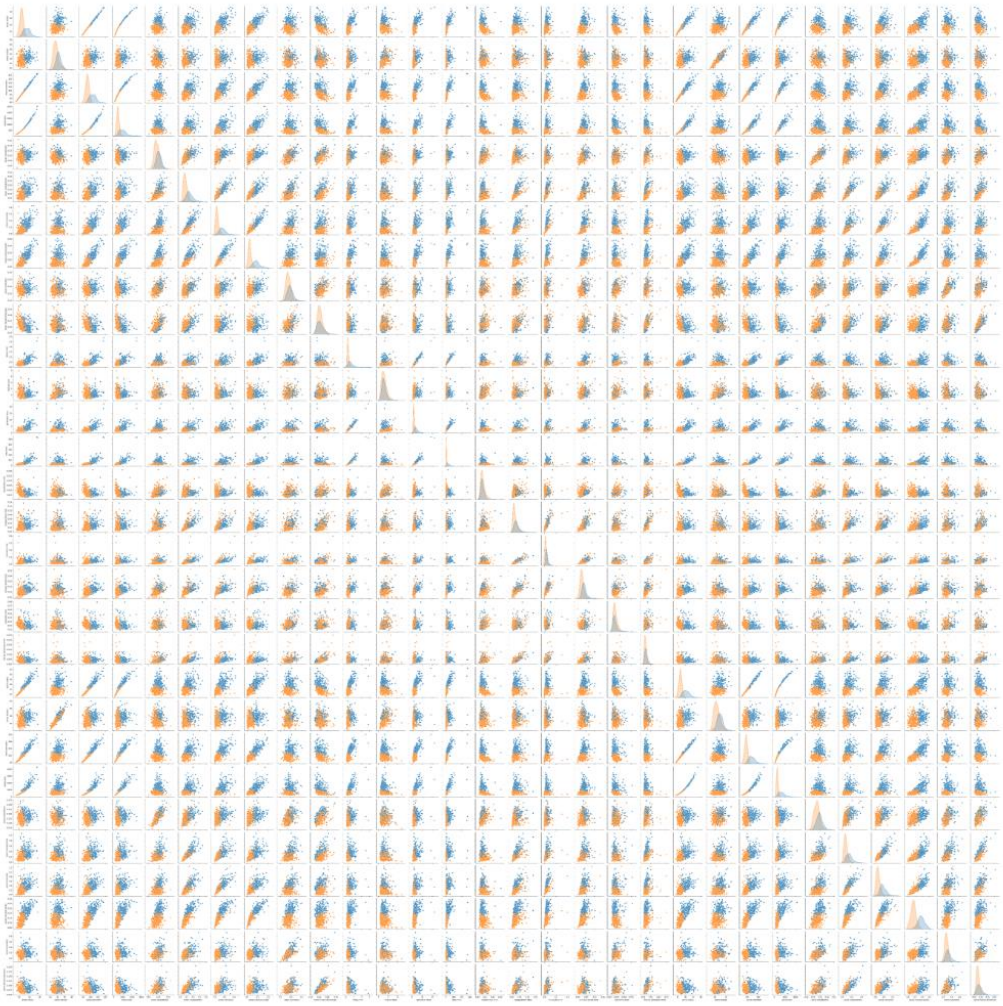
To Suitable Data Visualisation I used Following:-

1. Pairplot of Cancer Data-Frames :-

```
# Paiplot of cancer dataframe
sns.pairplot(cancer_df, hue = 'target')
```

Out[43]:

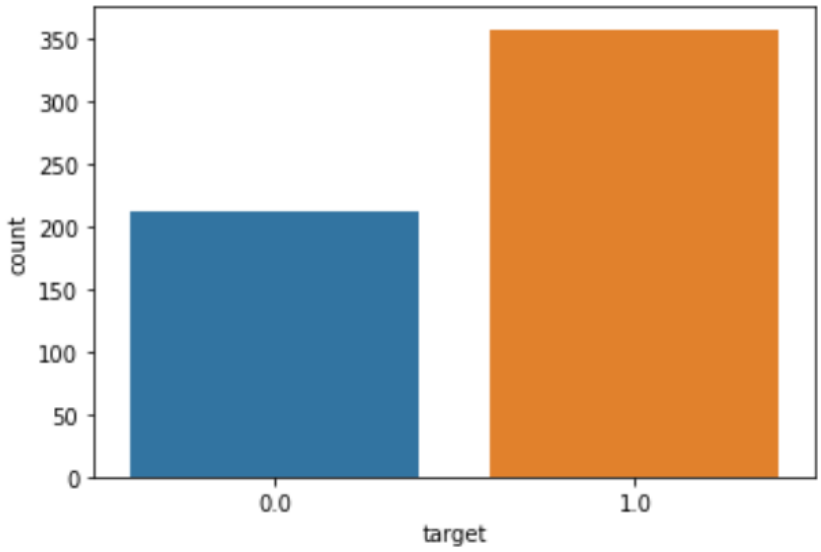
```
<seaborn.axisgrid.PairGrid at 0x173e4282fa0>
```



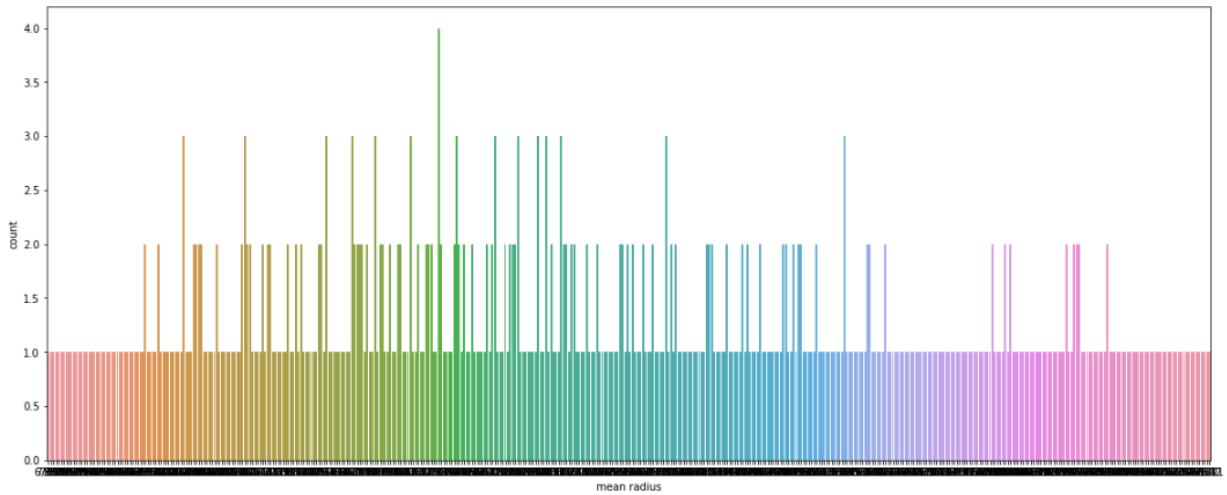
In [44]:

2. *Pair plot of some main features like mean radius, mean texture, mean perimeter, mean area, mean smoothness as follows :-*

3. Visualisating NO. of Different Data Targets using counting method of target class :-



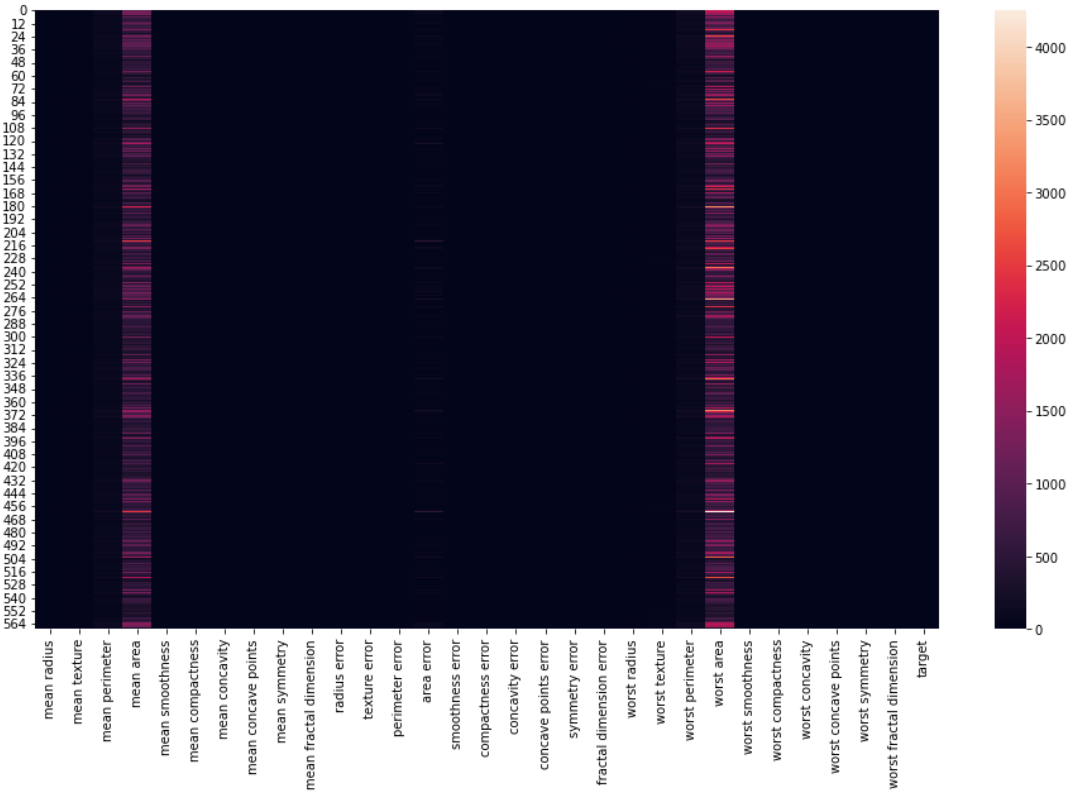
4. *counter plot of feature mean radius which plays a very curucial role in predicting Breast Cancer :-*



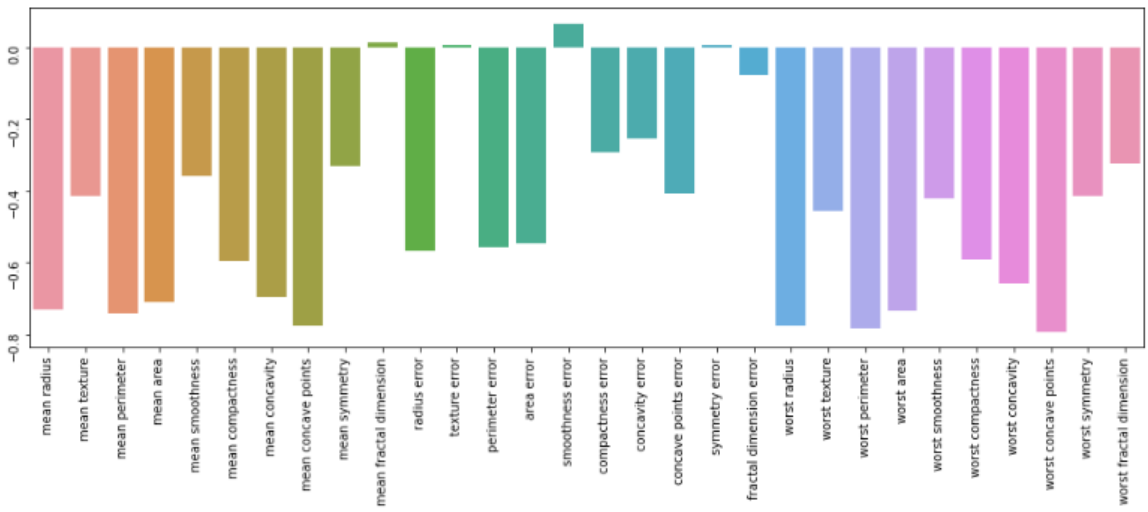
5. HeatMap method of Data Visuallisation :-

```
In [47]: # heatmap of DataFrame
         plt.figure(figsize=(16,9))
         sns.heatmap(cancer_df)

Out[47]: <AxesSubplot:>
```



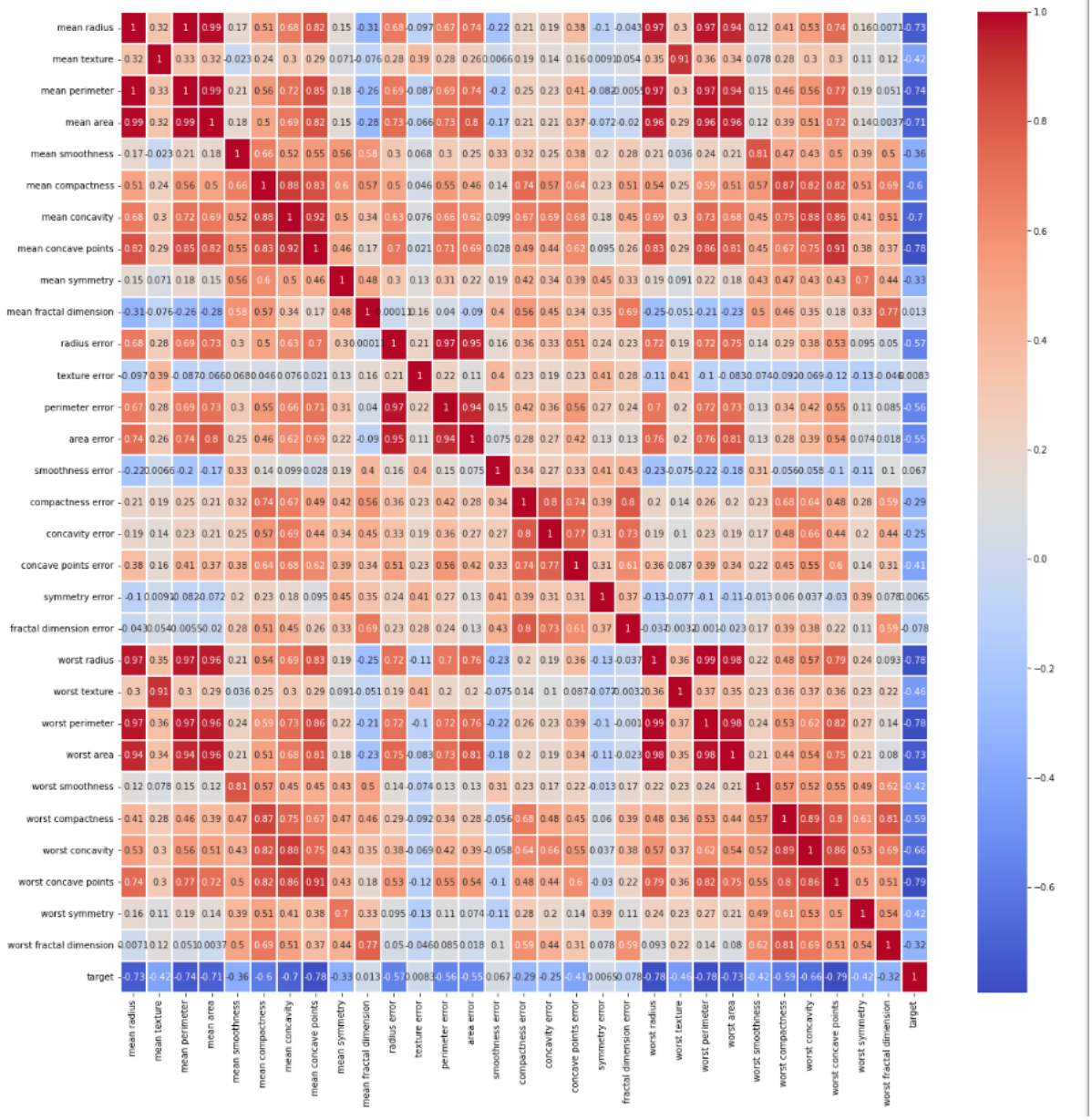
6. *Visualising Co-Relation Barplot by Dropping Target :-*



7. *Heatmap of Correlation matrix of breast cancer DataFrame :-*

```
# Heatmap of Correlation matrix of breast cancer DataFrame
plt.figure(figsize=(20,20))
sns.heatmap(cancer_df.corr(), annot = True, cmap ='coolwarm', linewidths=2)
```

```
<AxesSubplot:>
```



DATA PRE-PROCESSING

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real world data is often incomplete, inconsistent, and lacking certain to contain many errors. Data pre-processing is a proven method of resolving such issues. Data pre-processing prepares raw data for further processing. For pre-processing we have used standardization method to pre-process the UCI dataset. This step is very important because the quality and quantity of data that we gathered will directly determine how good our predictive model can be. In this case we collect the Breast Cancer samples which are Benign and Malignant. This will be our training data. we have splitted our dataset into two sub-sets as Training Dataset and Testing Dataset for this purpose we have taken 80% data as training of model and remaining of 20% data for testing of our model.

```
# split dataset into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state= 5)
```

After appling Data Splitting we get our data splitted into 4 categories as follows:-

a:-

```
X_test
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	com
28	15.30	25.27	102.40	732.4	0.10820	0.16970	0.16830	0.08751	0.1926	0.06540	...	20.27	36.71	149.30	1269.0	0.1641	
163	12.34	22.22	79.85	464.5	0.10120	0.10150	0.05370	0.02822	0.1551	0.06761	...	13.58	28.68	87.36	553.0	0.1452	
123	14.50	10.89	94.28	640.7	0.11010	0.10990	0.08842	0.05778	0.1856	0.06402	...	15.70	15.98	102.80	745.5	0.1313	
361	13.30	21.57	85.24	546.1	0.08582	0.06373	0.03344	0.02424	0.1815	0.05696	...	14.20	29.20	92.94	621.2	0.1140	
549	10.82	24.21	68.89	361.6	0.08192	0.06602	0.01548	0.00816	0.1976	0.06328	...	13.03	31.45	83.90	505.6	0.1204	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
414	15.13	29.81	96.71	719.5	0.08320	0.04605	0.04686	0.02739	0.1852	0.05294	...	17.26	36.91	110.10	931.4	0.1148	
515	11.34	18.61	72.76	391.2	0.10490	0.08499	0.04302	0.02594	0.1927	0.06211	...	12.47	23.03	79.15	478.6	0.1483	
186	18.31	18.58	118.60	1041.0	0.08588	0.08468	0.08169	0.05814	0.1621	0.05425	...	21.31	26.36	139.20	1410.0	0.1234	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	
261	17.35	23.06	111.00	933.1	0.08662	0.06290	0.02891	0.02837	0.1564	0.05307	...	19.85	31.47	128.20	1218.0	0.1240	

b:-

```
X_train
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	com
306	13.200	15.82	84.07	537.3	0.08511	0.05251	0.001461	0.003261	0.1632	0.05894	...	14.41	20.45	92.00	636.9	0.11280	
410	11.360	17.57	72.49	399.8	0.08858	0.05313	0.027830	0.021000	0.1601	0.05913	...	13.05	36.32	85.07	521.3	0.14530	
197	18.080	21.84	117.40	1024.0	0.07371	0.08642	0.110300	0.057780	0.1770	0.05340	...	19.76	24.70	129.10	1228.0	0.08822	
376	10.570	20.22	70.15	338.3	0.09073	0.16600	0.228000	0.059410	0.2188	0.08450	...	10.85	22.82	76.51	351.9	0.11430	
244	19.400	23.50	129.10	1155.0	0.10270	0.15580	0.204900	0.088860	0.1978	0.06000	...	21.65	30.53	144.90	1417.0	0.14630	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	15.49	30.73	106.20	739.3	0.17030	
73	13.800	15.79	90.43	584.1	0.10070	0.12800	0.077890	0.050690	0.1662	0.06566	...	16.57	20.86	110.30	812.4	0.14110	
400	17.910	21.02	124.40	994.0	0.12300	0.25760	0.318900	0.119800	0.2113	0.07115	...	20.80	27.78	149.60	1304.0	0.18730	
118	15.780	22.91	105.70	782.6	0.11550	0.17520	0.213300	0.094790	0.2096	0.07331	...	20.19	30.50	130.30	1272.0	0.18550	
206	9.876	17.27	62.92	295.4	0.10890	0.07232	0.017560	0.019520	0.1934	0.06285	...	10.42	23.22	67.08	331.6	0.14150	

c:-

```
y_train
```

```
306    1.0
410    1.0
197    0.0
376    1.0
244    0.0
      ...
8      0.0
73     0.0
400    0.0
118    0.0
206    1.0
```

d:-

```
y_test
```

```
28     0.0
163    1.0
123    1.0
361    1.0
549    1.0
      ...
414    0.0
515    1.0
186    0.0
3      0.0
261    0.0
```

Most of the times,our dataset contains features which

are highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidian distance between two data points in their computations. So, We need to bring all features to the same level of magnitudes. This is done by scaling.

After geeting desired splitting of data we do scalling so that our models can get better understanding between slight difference of data.

Feature scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.transform(X_test)
```

MODELS PERFORMANCES

We used 7 traditional models for the classification of breast cancer cases. Feature selection is applied to increase the rate of accurate prediction. Additionally, deep learning model is also built for the diagnosis system. Finally, we compare the performance of all the models applied and choose the one with the highest performance.

Different Models performance are :-

1. Support Vector Machine (SVM): Support Vector Machine finds an optimal hyperplane that best separates the classes based on the support vectors. The function of kernel for SVM is to take data as input and transform it into the required form.[13] The kernel function used in SVM model is linear function.

Suppor vector Classifier

```
# Support vector classifier
from sklearn.svm import SVC
svc_classifier = SVC()
svc_classifier.fit(X_train, y_train)
y_pred_scv = svc_classifier.predict(X_test)
accuracy_score(y_test, y_pred_scv)
```

```
0.9385964912280702
```

```
# Train with Standard scaled Data
svc_classifier2 = SVC()
svc_classifier2.fit(X_train_sc, y_train)
y_pred_svc_sc = svc_classifier2.predict(X_test_sc)
accuracy_score(y_test, y_pred_svc_sc)
```

```
0.9649122807017544
```

From above we see that Support Vector Machine performed very good as it achived about 96% accuracy with Standard Scaled Data.

2. Logistic Regression (LR): Logistic regression predicts the probability of the default class (e.g. Class 2 in this case) and transforms the probability into a binary value (0 or 1) for classification using ”sigmoid” function.

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
lr_classifier = LogisticRegression(random_state = 51, penalty = 'none')
lr_classifier.fit(X_train, y_train)
y_pred_lr = lr_classifier.predict(X_test)
accuracy_score(y_test, y_pred_lr)
```

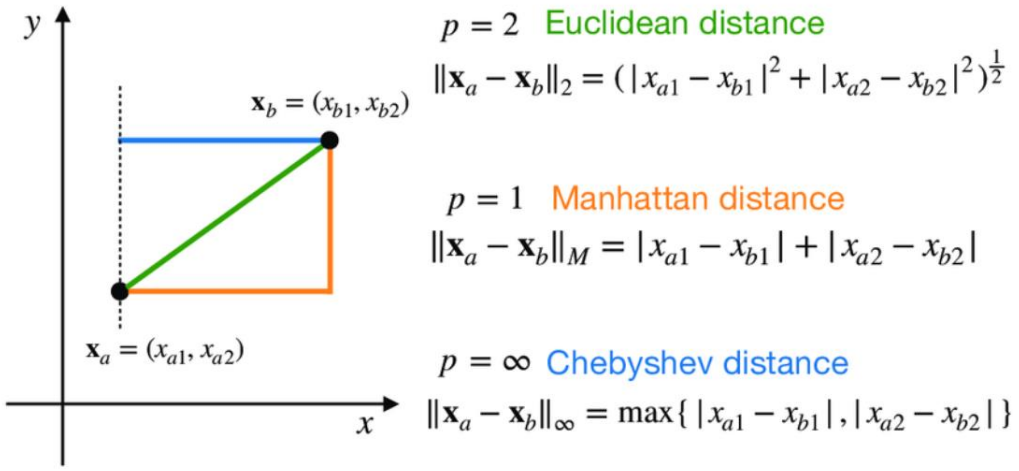
```
0.9649122807017544
```

```
# Train with Standard scaled Data
lr_classifier2 = LogisticRegression(random_state = 51, penalty = 'none')
lr_classifier2.fit(X_train_sc, y_train)
y_pred_lr_sc = lr_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_lr_sc)
```

```
0.45614035087719296
```

From logistic regression model we see that this model performs very well but it failed to perform good with scalled data.

3. K-Nearest Neighbor (k-NN): K-nearest neighbor assigns a case to the class that is most common among its k nearest neighbors. The distance between the case and its neighbor is measured by using distance functions like :-



Three typical Minkowski distances, i.e., Euclidean, Manhattan and Chebyshev distances.

K – Nearest Neighbor Classifier

```
# K - Nearest Neighbor Classifier
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn_classifier.fit(X_train, y_train)
y_pred_knn = knn_classifier.predict(X_test)
accuracy_score(y_test, y_pred_knn)
```

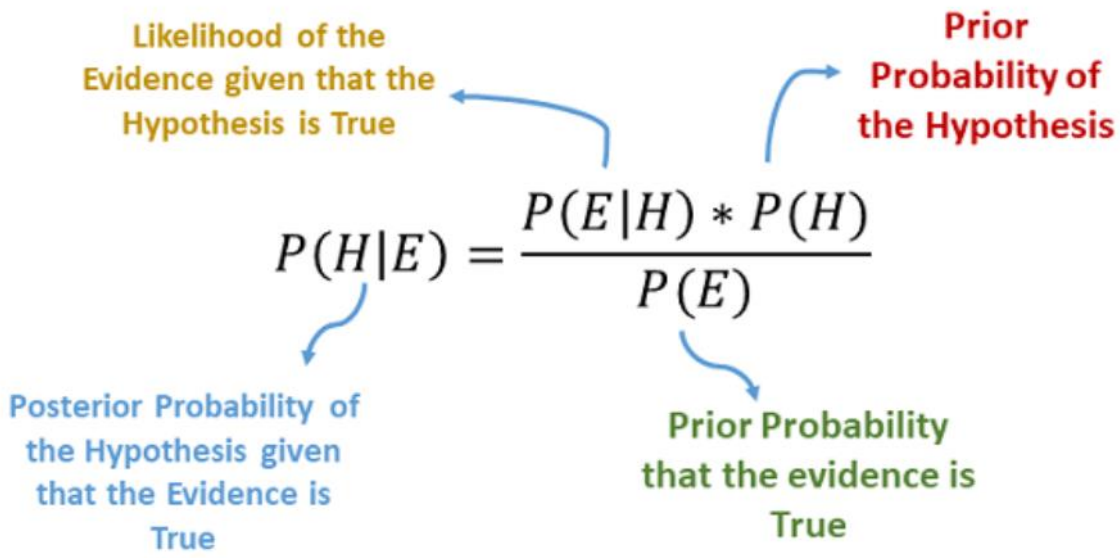
```
0.9385964912280702
```

```
# Train with Standard scaled Data
knn_classifier2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn_classifier2.fit(X_train_sc, y_train)
y_pred_knn_sc = knn_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_knn_sc)
```

```
0.5789473684210527
```

From K-Nearest Neighbour Classifier we see that this model performs very well but it failed to perform good with scalled data.

4. Naive Bayes Classifier: Naive Bayes algorithm makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting using the given formuls:-



Naive Bayes Classifier

```
# Naive Bayes Classifier
from sklearn.naive_bayes import GaussianNB
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred_nb = nb_classifier.predict(X_test)
accuracy_score(y_test, y_pred_nb)
```

```
0.9473684210526315
```

```
# Train with Standard scaled Data
nb_classifier2 = GaussianNB()
nb_classifier2.fit(X_train_sc, y_train)
y_pred_nb_sc = nb_classifier2.predict(X_test_sc)
accuracy_score(y_test, y_pred_nb_sc)
```

```
0.9385964912280702
```

From Naive Bayes we see that that it produces a good result of 94% and 93% of accuracy with normal form of data and scalled form of data respectively.

5. Decision Tree Classifier :- The decision tree are presented with a tree structure. The test objects are classified by their feature values. A node in a decision tree represents an instance, outcomes of the test represented by branch, and the leaf node opitomized the class label.

Decision Tree Classifier

```
# Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
dt_classifier.fit(X_train, y_train)
y_pred_dt = dt_classifier.predict(X_test)
accuracy_score(y_test, y_pred_dt)
```

```
0.9473684210526315
```

```
# Train with Standard scaled Data
dt_classifier2 = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
dt_classifier2.fit(X_train_sc, y_train)
y_pred_dt_sc = dt_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_dt_sc)
```

```
0.7543859649122807
```

From here we see that Decision tree performed well with normal form of data with an accuracy of approx 94% whereas it’s performance decreases with the standard scaled form of data.

6. Random Forest classifier :- Random forest, like its name implies, consists of many individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

Random Forest Classifier

```
# Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
accuracy_score(y_test, y_pred_rf)
```

```
0.9736842105263158
```

```
# Train with Standard scaled Data
rf_classifier2 = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 51)
rf_classifier2.fit(X_train_sc, y_train)
y_pred_rf_sc = rf_classifier.predict(X_test_sc)
accuracy_score(y_test, y_pred_rf_sc)
```

```
0.7543859649122807
```

From random forest decision tree we see that it performed well with normal form of data with an accuracy of approx 94% whereas it's performance decreases with the standard scaled form of data with an accuracy of approx 75%.

7. XgBoost Classifier:- XGBoost, or extreme gradient boosting, is a top machine learning model and the dominating algorithm among competitions. it supports many tunings parameters such as learning rate and early stopping. As it is a tree boosting algorithm, minimal data-preprocessing is needed (i.e. data normalization/scaling).

XGBoost Classifier

```
# XGBoost Classifier
#from xgboost import XGBClassifier
from xgboost import XGBClassifier
xgb_classifier = XGBClassifier()
xgb_classifier.fit(X_train, y_train)
y_pred_xgb = xgb_classifier.predict(X_test)
accuracy_score(y_test, y_pred_xgb)
```

```
0.9824561403508771
```

```
# Train with Standard scaled Data
xgb_classifier2 = XGBClassifier()
xgb_classifier2.fit(X_train_sc, y_train)
y_pred_xgb_sc = xgb_classifier2.predict(X_test_sc)
accuracy_score(y_test, y_pred_xgb_sc)
```

```
0.9824561403508771
```

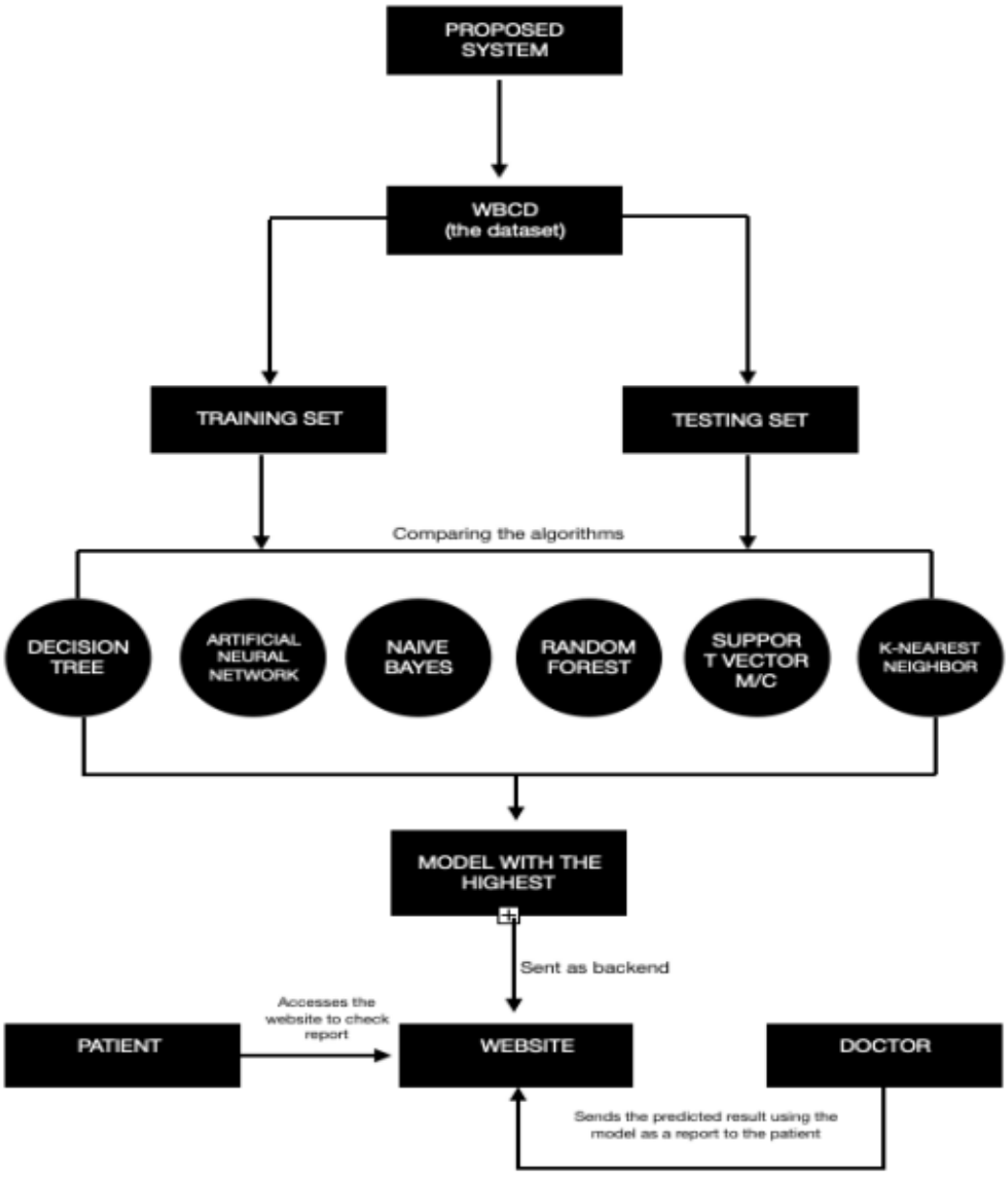
From XgBoost classifier we see that it performed very well with normal form of data with an accuracy of approx 98% as well as it performed well with the standard scaled form of data with an accuracy of approx 98% and with this we sucussfully beats all the tested model as it outperformed in all of them.

CONCLUSION AND FUTURE SCOPE

This project examined different machine learning techniques for detection of breast cancer. The objectives of our study was to analyse the Wisconsin breast cancer dataset by visualizing and evaluating Machine Learning Predictions with this research project we can see that among Naïve Bayes, Support Vector Machine, Adaboost, Random Forest Classifier, KNN, Decision Tree, XGboost etc. We concluded that XGboost is the most accurate algorithm for best accurate result for detection of breast cancer with the efficiency of 98.24%. However, it is required that before running the algorithm, the dataset must be pre-processed.

In future, A comparison between all the six algorithms will be made. The algorithm that gives the best results will be supplied as a model to the website. The website will be made from a python framework, called flask. And it will host the database on Xampp or Firebase or inbuilt Python and flask libraries. This data set is available on the UCI Machine Learning Repository. It consists of 31 real world attributes which are multivariate. The total number of instances is 569 and there are no missing values in this data set. The process of the proposed system is as follows,

1. The patient books an appointment through our website.
2. The patient will then meet the doctor offline for the respective appointment.
3. The doctor will first check the patient manually, then perform a breast mammogram or an ultrasound. That ultra sound will show an image of the breast consisting the lumps or not.
4. If the lumps are detected, a biopsy will be performed. The digitised image of the Fine Needle Aspirate (FNA) is what forms the features of the dataset.
5. Those numbers will be provided to the system by the doctor and the model will detect if it’s a benign or a malignant cancer.
6. The report will be then forwarded to the patient on their respective account.



The entire open-sourced code is linked below:-
https://github.com/Raj-saurabh/Machine-Learning-Various-Models-to-Breast-Cancer-Prediction

References

Breast Cancer Detection using Machine Learning Algorithm
https://www.irjet.net/archives/V8/i2/IRJET-V8I2129.pdf

Breast cancer detection by leveraging Machine Learning
https://www.sciencedirect.com/science/article/pii/S2405959520300801

Breast Cancer Detection using Machine Learning
https://www.ijraset.com/research-paper/breast-cancer-detection-with-machine-learning

Breast Cancer Wikipedia
https://en.wikipedia.org/wiki/Breast_cancer

Breast Cancer Detection Using Machine Learning Classifier
https://indianaiproduction.com/breast-cancer-detection-using-machine-learning-classifier/

Breast Cancer Classification and Prediction using Machine Learning
https://www.ijert.org/research/breast-cancer-classification-and-prediction-using-machine-learning-IJERTV9IS020280.pdf

BREAST CANCER DISANOSIS USING MACHINE LEARNING METHOD
http://noiselab.ucsd.edu/ECE228-2020/projects/Report/31Report.pdf

BREAST CANCER PREDICTION USING MACHINE LEARNING
https://www.researchgate.net/publication/341508593_BREAST_CANCER_PREDICTION_USING_MACHINE_LEARNING

Github Repository:-
https://github.com/kwantommy/breast-cancer-diagnosis