|  |  |
| --- | --- |
| BREAST CANCER DETECTION | Abstract  Breast cancer is one of the most common cancer and is causing a huge number of deaths in women. In this project, we developed Machine Learning Classification Models to detect breast cancer based on a set of features calculated from a digitized image of the Fine Needle Aspiration (FNA) of a breast mass from a patient.  Steshi  Btech. Computer Science |

**INTRODUCTION**

Breast cancer is one of the most common cancer in women and the second leading cause of women’s cancer death. Mammography is a traditional method used for diagnosing breast cancer. According to UCHealth’s report, only 78% of breast cancer can be accurately diagnosed by mammography 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. 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 in a large number of projects. 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 paper, we will train our models using this dataset.

The input of our algorithm is a set of features calculated from a digitized image of the Fine Needle Aspiration (FNA) of a breast mass from a patient. We will then use four traditional methods including Logistic Regression, Nearest Neighbour (K-NN), Support Vector Machines (SVM), and Random Forest Algorithm, and a deep learning method to predict whether the case is benign or malignant.

**DATA CONTENT**

The dataset we used is Breast Cancer Wisconsin dataset which is a widely used dataset in study. Features are computed from a digitized image of a Fine Needle Aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. It contains 569 instances with 2 classes (benign and malignant) and 9 features. The class distribution is as follows: 357 benign and 212 malignant. The features are:

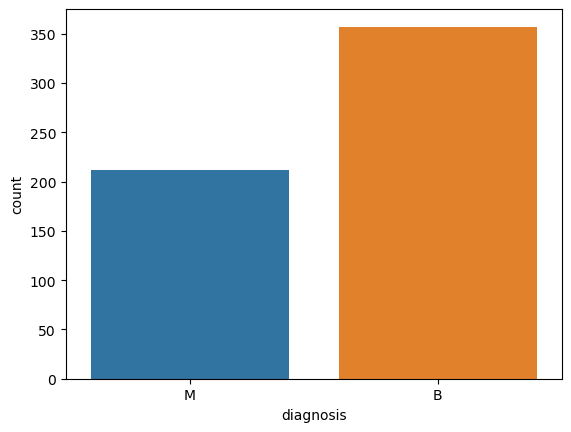
1.ID number  
2.Diagnosis(M=Malignant ,B=Benign)  
3.radius(mean of distances from centre to points on the parameter)  
4.texture(standard deviation of gray-scale values)  
5.perimeter  
6.area  
7.smoothness(local variation in radius lengths)  
8.compactness(perimeter2/area-1.0)  
9.concavity(severity of concave portions of the contour)  
10.concave points(number of the concave portions of the contour)  
11.symmetry  
12.farctal dimension("coastline approximation"-1)  
13.The mean, standard error and "worst" or largest(mean of the three largest values) of these features were computed for each image, resulting in 30 features.

14.All feature values are recorded with four significant digits.  
15.Missing attribute values: none.

**EDA AND PREPROCESSING**

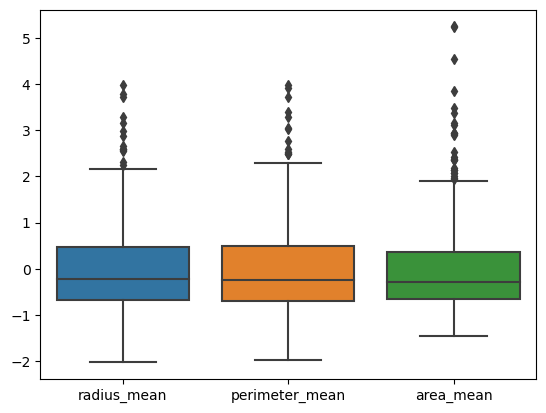
The dataset is not always ready to use. In EDA, we will be doing pre-processing of the data by analysing the features and the target variable, checking for the null values, etc. We began by reading the data and creating a dataframe and describing it’s dimensions. After that we checked for the different classes in our target variable. We obtained two classes: M(Malignant) and B(Benign). Then we checked for the data types of the various features which we found out to be correct for all the features.

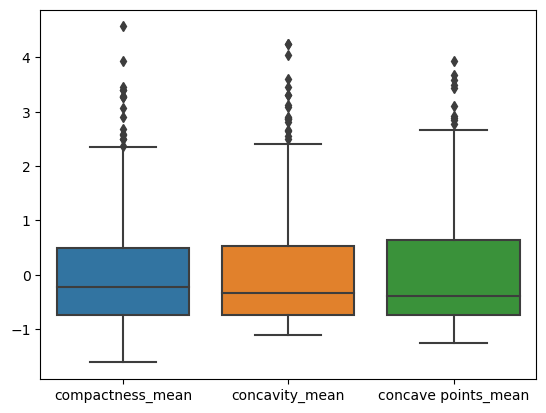
Then we checked for the null values, no null values were found in this dataset. After doing this analysis, we found out that the ‘id’ feature will not give any useful information for the classification purpose. Thus, we removed this column from our dataset. Now, we were left with 30 features and 1 target label. After all this processing our dataset is clean. Now we proceeded to visualize the dataset and draw more results from it.

In the first place, we began by visualizing our target variable using countplot from the seaborn library. The results from the visualization are shown in the figure. After this visualization, it can be said that the Benign Cases are more than the Malignant Cases. 

After the visualization of the target variable we move to visualize the features.

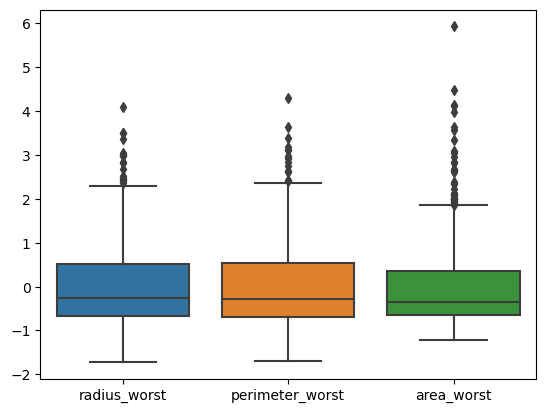
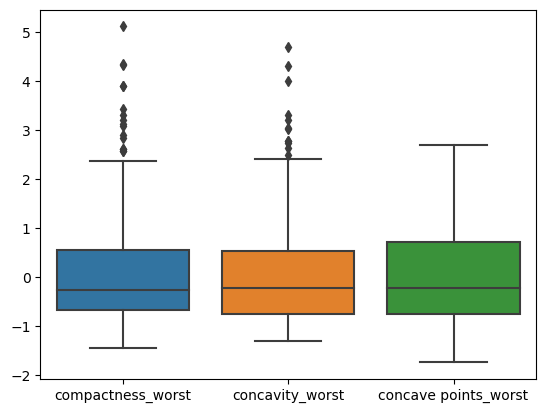
Firstly, we plotted the mean of all the features using the scatter matrix. After this , we came to a conclusion that the some of the features were highly correlated to each other which can be shown by the boxplot figures below:





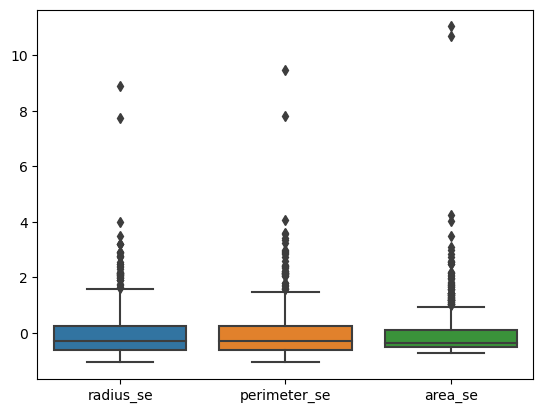
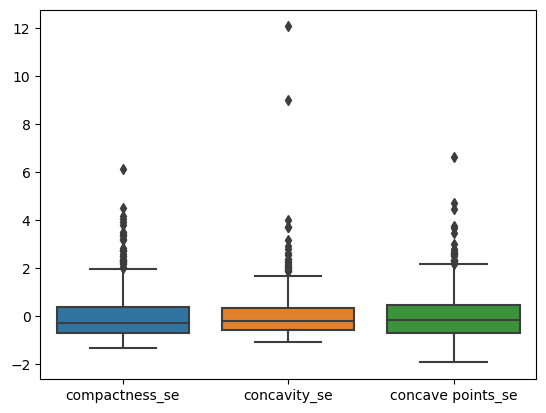
Thus, the correlation in the above features is verified using boxplot. This result will be later used in the selection of the features.

Following this, we plotted the worst of all the features using the scatter matrix. Similar to the previous visualization, some of the features were found correlated in this matrix too. These are shown as below:

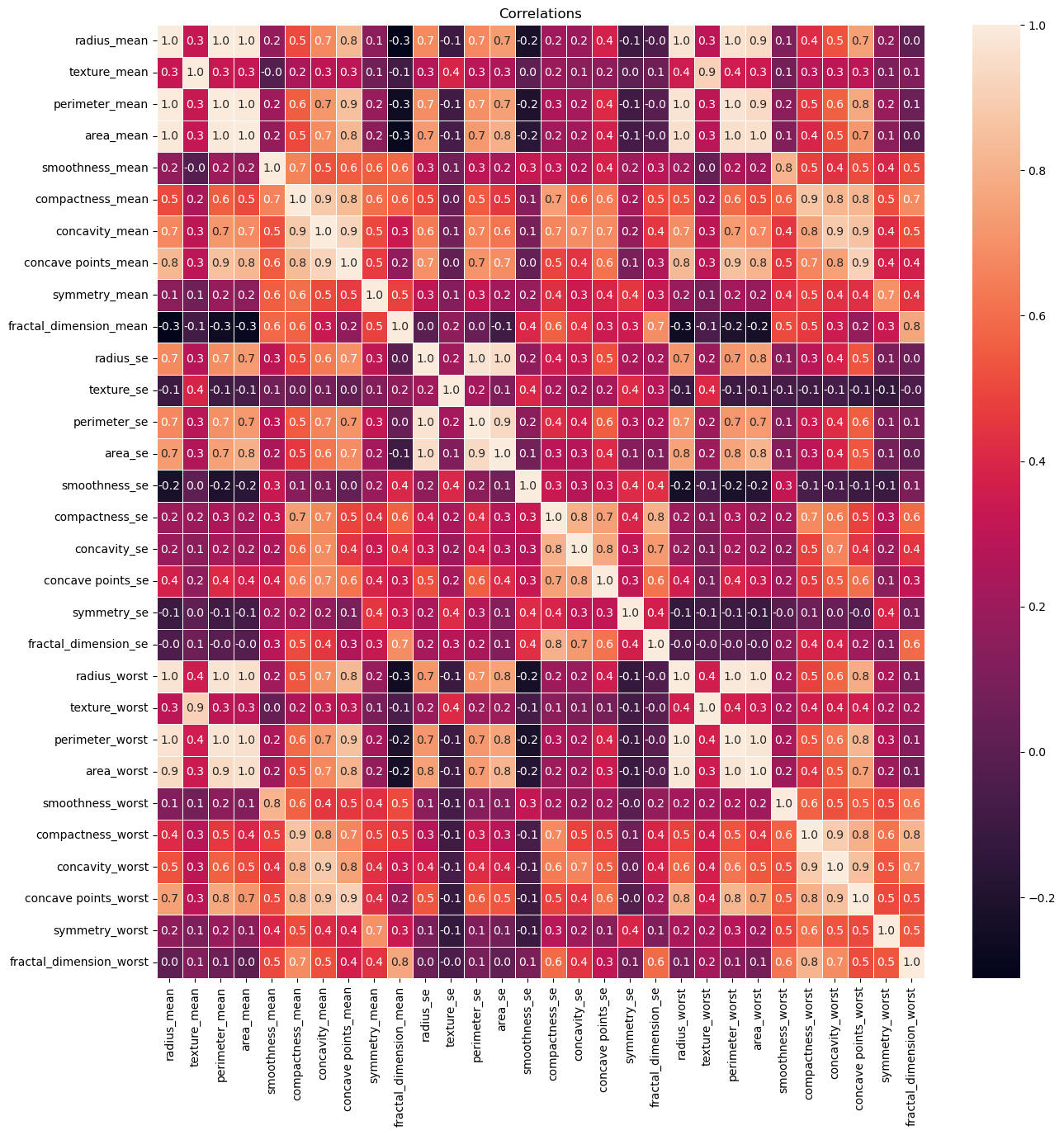
 

Hence, the correlation between these features is also confirmed.

Next, we plotted the Standard Error features with the help of scatter matrix. In this visualization too, we obtained some highly correlated features whose correlation can be verified and shown with the help of boxplot as :

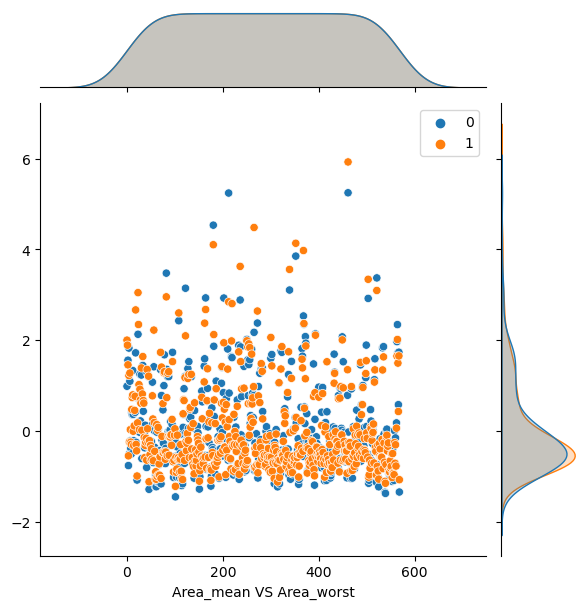


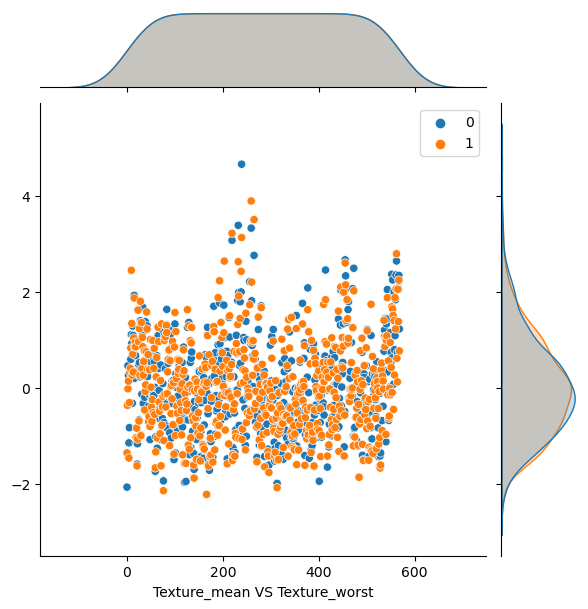
After establishing these correlations , we next plot the heatmap to find more correlations among the features. It is a graphical representation of data using colours to visualize the value of the matrix. It is shown as below:



As it can be seen from the heatmap, area\_mean and area\_worst

and texture\_mean and texture\_worst are correlated. We verified this by using the jointplot as shown:





Then we used results from the above visualizations for feature selection.

As it is clear that radius\_mean,perimeter\_mean and area\_mean are correlated with each other so we used only area\_mean.Compactness\_mean, concavity\_mean and concave points\_mean are correlated with each other, we used concavity\_mean. Also, radius\_worst ,perimeter\_worst and the area\_worst are correlated we used area\_worst and compactness\_worst, concavity\_worst and concave points\_worst are correlated we used concavity\_worst. Apart from these, radius\_se ,perimeter\_se and area\_se and compactness\_se,concavity\_se and concave points\_se are correlated , we used area\_se and concavity\_se. texture\_mean and texture\_worst are correlated, we used texture\_mean. area\_worst and area\_mean are correlated, we used area\_mean. Hence, we dropped the unused features.

After this preprocessing of the dataset, we performed the X-Y or Input -Output Split. In this split, we separate the input features and the target variable. Subsequently , we normalized the range of the values of the input features using the StandardScaler. It standardize the features by removing the mean and scaling to unit variance.

Z=(x-u) / s

Where, u is the mean of the training sample and s is the standard deviation.

After the normalization, we split the data into training and testing data. We used 70% of the data for training and the remaining 30% for testing.

**MODEL CLASSIFICATION**

Since we have to predict the class, it is a classification problem. We used four Machine Learning Models including Logistic Regression, Random Forest Algorithm, SVM and KNN Model. Feature selection is already applied to increase the rate of accurate prediction. Finally, we compared the performance of all the models applied and chose the one with the highest performance.

**LOGISTIC REGRESSION**

It is a supervised learning classification algorithm used to predict the probability of a target variable with dichotomous nature i.e. there would be only two possible classes. It is similar to Linear Regression but instead of fitting a *regression line* we fit a **Sigmoid/Logistic Function**, which predicts two maximum values (0 or 1).

**RANDOM FOREST**

Random forest is a set of individual decision trees. Each decision tree spits out a class prediction.It decides the class of the test object by aggregating the votes from different decision trees.

**SVM(Support Vector Machine)**

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. The kernel function used in SVM model is linear function as shown:

k(xi; xj) = a < xi; xj > +b

**KNN**

K-nearest neighbour assigns a case to the class that is most common among its k nearest neighbours. The distance between the case and its neighbour is measured by using distance functions like Euclidean, Manhattan and Minkowski.

**RESULTS**

|  |  |
| --- | --- |
| Model | Accuracy |
| Logistic Regression | 0.976 |
| Random Forest | 0.964 |
| SVM | 0.982 |
| KNN | 0.954 |

The figure shows the accuracy obtained from different models. The lowest results were obtained from the KNN model and the highest from the SVM model with 95.4% and 98.2% accuracy respectively. Except SVM, the other models did a relatively good work in classifying benign cases, but the number of misclassified malignant cases is relatively large, especially in the case of KNN model, whereas, the SVM Model did a good work in classifying both the classes.

**CONCLUSION**

In conclusion, comparing with the 78% accuracy of mammography, all the four models achieved a higher accuracy of more than 95% for breast cancer diagnosis. Also, it can be concluded that, among all the models, the SVM model provided the best result (more than 98% accuracy) to our dataset.