Machine Learning Engineer Nanodegree

Capstone Project Godaba Hari Chandana

February 10th, 2019

Breast Cancer Wisconsin (Diagnostic) Classification

I. Definition

Project Overview:

In these days the Domain of Health Care is stepping to get engaged with Artificial Intelligence and Machine Learning. This project is based on the Domain of Health Care. This gives accurate results regarding the classification of Breast Cancer. It takes the features of cells in the body and predicts that the woman is affected with Breast Cancer or not. It's a good initiation to implement this project in HealthCare Centers. The aim of this application is to avoid the *Human Errors* in the Domain of **Health Care** .

The dataset is collected from UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

cite or link of an academic paper where machine learning was applied to this type of problem is :

https://www.sciencedirect.com/science/article/pii/S1877050916302575

Problem Statement:

Given a dataset which consists the features of cells in the body and an algorithm needs to be developed to classify the Breast Cancer and to determine if it is malignant or benign .This can then be used to automatically detect the women who are suffering from Breast Cancer.

Hence the goal is to classify a new data point to either Malignant or Benign on basis of the features of the given data point. The tasks involved in it:

1. Download the data from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
2. Cleaning the data and removing the unnecessary features which are not used to predict the target variable.
3. Visualising the data to know the characteristics of the features.

4. Train the data with Classification Models (without Feature Reduction) and then test to find which classifier performs good on the dataset by using appropriate methods then finally identifies the model which gives the best accuracy score.
5. Also Train the data with the Classification models (with feature reduction) and then test to find which the classifier performs good on the dataset and then finally identifies the model which gives the best accuracy score.

Evaluation Metrics:

For the prediction of Breast Cancer Wisconsin I want to use accuracy score as an evaluation metric. It is the number of correct predictions made by the model over all kinds of predictions made. Firstly I will predict the accuracy score given by the classification models without feature reduction and then I will select a model which gives the best accuracy score. Then I will also predict the accuracy scores given by all classification models with feature reduction then I will select a model which gives the best accuracy.

Accuracy Score can be also calculated as:

Accuracy Score = TP+TN/TP+FP+FN+TN

True positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.

True negatives (TN): We predicted no, and they don't have the disease.

False positives (FP): We predicted yes, but they don't actually have the disease. (Also known as "Type 1error")

False negatives (FN): We predicted no, but they actually do have the disease.(Also known as "Type 2 error"

II. Analysis

Data Exploration:

The dataset which is used for Breast Cancer Wisconsin(Diagnostic) Classification is taken from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

This dataset consists of 569 instances with 32 attributes and it is a multivariate because it consists Categorical, real and integer valued attributes. For the best result I will split the data into training and testing sets. On a whole I will assign 70% of data to training set and 30% of data testing set.

Attributes are:

1. ID number

2. Diagnosis (M = malignant, B = benign)

3-32) Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)

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. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recoded with four significant digits.

Missing attribute values: none

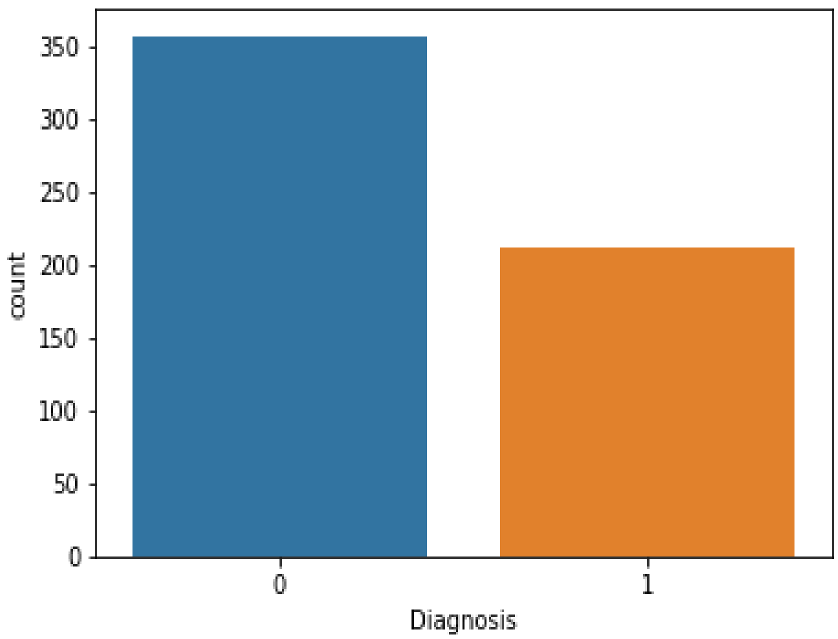
Class distribution: 357 benign, 212 malignant

Exploratory Visualization:

Histogram:

A histogram is a display of statistical information that uses rectangles to show the frequency of data items in successive numerical intervals of equal size. In the most common form of histogram, the independent variable is plotted along the horizontal axis and the dependent variable is plotted along the vertical axis. The data appears as coloured or shaded rectangles of variable area.

The blue part of the histogram represents the features with Diagnosis==”benign” while the orange part represents the features with Diagnosis==”malignant”.



By representing the features with Histograms we can know how the data is structured and if any preprocessing is required.

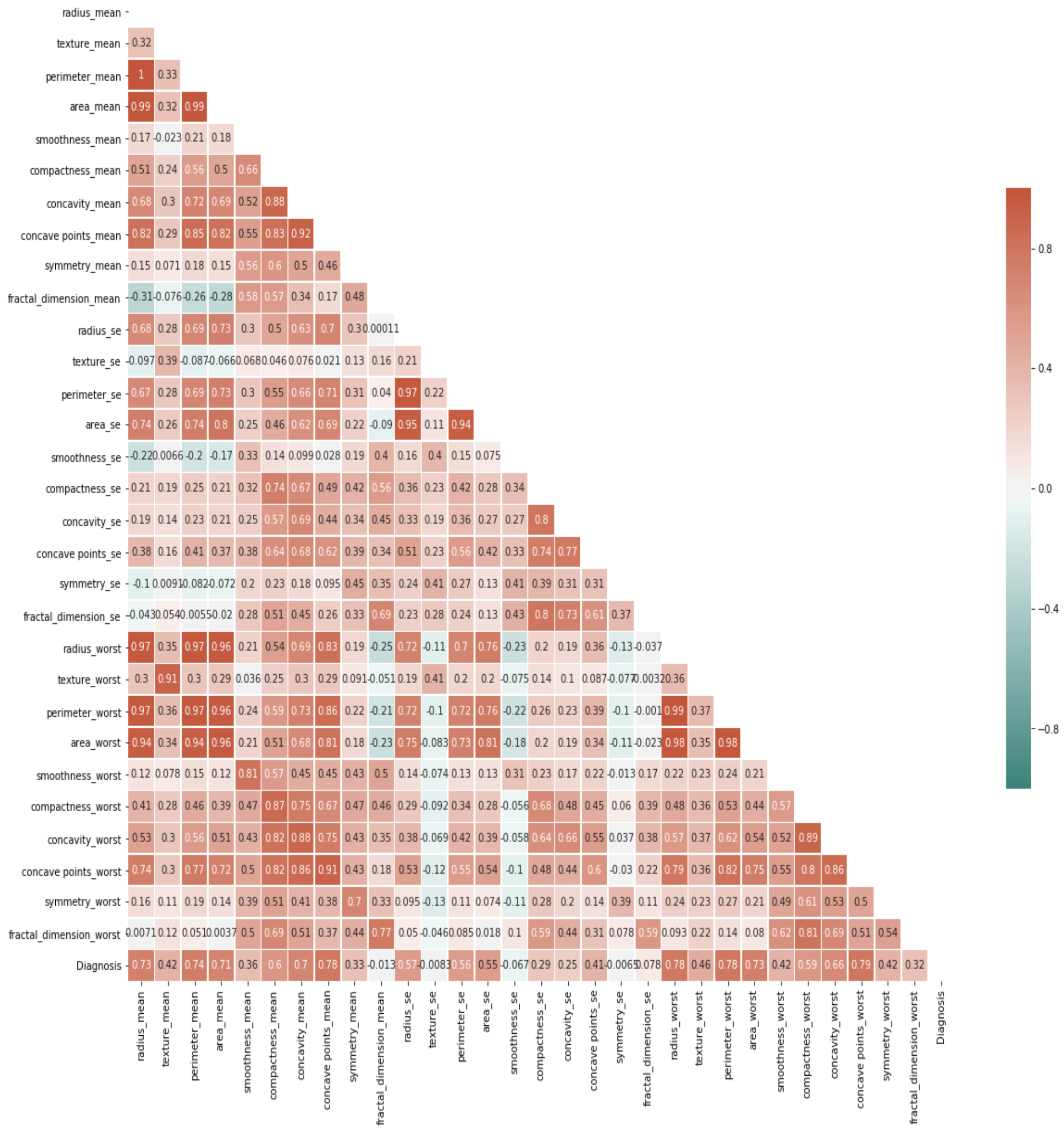
Correlation Matrix:

A correlation matrix is a table or matrix showing correlation coefficients between sets of variables. Each random variable (Xi) in the table is correlated with each of the other values in the table (Xj). This allows you to see which pairs have the highest correlation.

The diagonal of the table is always a set of ones, because the correlation between a variable and itself is always 1. You could fill in the upper-right triangle, but these would be a repeat of the lower-left triangle in other words correlation matrix is also a symmetric matrix.

Correlation matrix is also used for feature Reduction. The below image shows correlation matrix of the features of Breast Cancer Wisconsin(Diagnosis) data.

One of the dataset's hallmarks is relatively high correlation coefficient score - only score no higher than 0.6 will be considered acceptable. Correlation not necessarily means causation, that is why features will not be excluded only for their low correlation with diagnosis. Hence the selected features are 'smoothness_mean', 'radius_se', 'texture_se', 'smoothness_se', 'symmetry_se','fractal_dimension_se', 'texture_worst', 'symmetry_worst','fractal_dimension_worst'.



Algorithms and Techniques:

The algorithms used in BreastCancerWisconsin(Diagnosis) classification are:

1. Random Forest
2. SVM
3. KNeighbors
4. GaussianNB

5. DecisionTree
6. LogisticRegression

Random Forest Algorithm:

- Random forest is a predictive modelling algorithm (not any descriptive modelling algorithm).
- The random forest can be used for both classification and regression tasks.
- It works well with default hyper-parameters.
- It can be used to rank the importance of variables in a regression or classification problem.
- The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
- A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.
- It runs efficiently on large datasets.

Parameters:

Class sklearn.ensemble.RandomForestClassifier (n_estimators=’warn’, criterion=’gini’, max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=’auto’, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None)

Advantages:

Reduction in over fitting: by averaging several trees, there is a significantly lower risk of over fitting.

Less variance: By using multiple trees, you reduce the chance of stumbling across a classifier that doesn’t perform well because of the relationship between the train and test data.

Disadvantages:

It takes more time to train samples.

Support Vector Machine:

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges.
- a support-vector machine 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.
- Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

Parameters:

```
class sklearn.svm.SVC(C=1.0, kernel=’rbf’, degree=3, gamma=’auto_deprecated’,
coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, decision_function_shape=’ovr’,
random_state=None)
```

Advantages:

It works really well with clear margin of separation

It is effective in high dimensional spaces.

It is effective in cases where number of dimensions is greater than the number of samples.

It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Disadvantages:

It doesn’t perform well, when we have large data set because the required training time is higher

It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping

SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

KNeighbors Algorithm:

- K-NN is a type of instance based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

- Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

- In pattern recognition ***k*-NN** is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In *k-NN* regression, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

Parameters:

class sklearn.neighbors.**KNeighborsClassifier***(*n_neighbors=5*,* weights='uniform'*,* algorithm='auto'*,* leaf_size=30*,* p=2*,* metric='minkowski'*,* metric_params=None*,* n_jobs=None*,* **kwargs*)*

Advantages:

Robust to noisy training data (especially if we use inverse square of weighted distance as the "distance")

Effective if the training data is large.

Disadvantages:

Need to determine the value of parameter K (number of nearest neighbors)

Distance based learning is not clear which type of distance and which attribute to use to produce the best results.

Computational cost is quite high because we need to compute distance of each query instance to all training samples.

GaussianNB Algorithm:

- Naive bayes is a straight forward and powerful algorithm for the classification task.
- Naive bayes classifier gives great results when we use it textual data analysis such as Natural Language Processing.
- GaussianNB works by using Bayes' Theorem where Bayes'Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.
- Bayes' Theorem is stated as:

  P(h|d) = (P(d|h) * P(h)) / P(d)

Where

P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.

P(d|h) is the probability of data d given that the hypothesis h was true.

P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

P(d) is the probability of the data (regardless of the hypothesis).

Parameters:

class sklearn.naive_bayes.**GaussianNB**(priors=None, var_smoothing=1e-09)

Advantages:

Naive Bayes Algorithm is a fast, highly scalable algorithm.

Naive Bayes can be use for Binary and Multiclass classification. It provides different types of Naive Bayes Algorithms like GaussianNB, MultinomialNB, BernoulliNB.

It is a simple algorithm that depends on doing a bunch of counts.

Great choice for Text Classification problems. It's a popular choice for spam email classification.

It can be easily train on small dataset

Disadvantages:

It considers all the features to be unrelated, so it cannot learn the relationship between features.

Decision Tree Algorithm:

- Decision tree is one of the most popular machine learning algorithms used for both classification and regression problems.
- Decision trees often mimic the human level thinking so it is very simple to understand the data and make some good interpretations.
- Decision trees actually make you see the logic for the data to interpret(not like black box algorithms like SVM, NN, etc..)
- It is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).

Parameters:

class sklearn.tree.**DecisionTreeClassifier**(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False*)*

Advantages:

Decision trees are able to generate understandable rules.

Decision trees perform classification without requiring much computation.

Decision trees are able to handle both continuous and categorical variables.

Decision trees provide a clear indication of which fields are most important for prediction or classification.

Disadvantages:

Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.

Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

Decision tree can be computationally expensive to train. The process of growing a decision tree is also expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

Logistic Regression:

- Logistic Regression is one of the most used Machine Learning algorithms for binary classification and It gives a discrete binary outcome between 0 and 1 to say it in simple words it's outcome is either one thing or another.

- Logistic Regression works by measuring the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (our features), by estimating probabilities using it's underlying logistic function. These probabilities must then be transformed into binary values in order to actually make a prediction.

Parameters:

class sklearn.linear_model.**LogisticRegression**(penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='warn', max_iter=100, multi_class='warn', verbose=0, warm_start=False, n_jobs=None)

Advantages:

Outputs have a nice probabilistic interpretation, and the algorithm can be regularized to avoid overfitting.

Logistic models can be updated easily with new data using stochastic gradient descent.

Disadvantages:

Logistic regression tends to underperform when there are multiple or non-linear decision boundaries.

They are not flexible enough to naturally capture more complex relationships.

Benchmark:

In my project I will take benchmark model as DecisionTree and the Accuracy score will be compared between the different classification models (Random Forest Classifier, Logistic Regression, GaussianNB, KNeighborsClassifier, Decision Tree Classifier, SVM) and select the best one.

1. Before feature reduction the Benchmark value is an accuracy score of 92%.
2. After feature reduction the Benchmark value is an accuracy score of 81%.
   The model which gives the accuracy score greater than these accuracy scores then those models are considered to be good.

III. Methodology

Data preprocessing:

According to my project the process of data preprocessing involves reading the data by using read_csv and perform visualizations on it to get some insights about the data. After that I will clean the data that is remove the features which are not used for the prediction of target variable and also remove the unwanted data or replacing null values with some constant values or removing duplicates

Training and Testing the data:

After the data processing, I will split the whole data into training and testing sets by using train_test_split from sklearn.model_selection and then train the data by using different classification models like Random Forest Classifier, Decision Tree Classifier, SVC, Logistic Regression, KNeighborsClassifier, GaussianNB and then test all the models with testing data and then find the accuracy score given by the models and then finally identifies which model that gives the best accuracy score. Hence this model is declared as the best model to detect the Breast Cancer Wisconsin.

Before feature reduction the accuracy scores given by different models are:

Random Forest Classifier	-- 94%
Support Vector Classifier	--60%
KNeighborsClassifier	--95%
GaussianNB	--94%

Decision Tree Classifier	--92%
Logistic Regression	--95%

From the above information I can conclude that before feature reduction KNeighborsClassifier and LogisticRegression gives best accuracy score which is 95% and it is also greater than the accuracy score given by the benchmark model which is DecisionTreeClassifier. In this scenario KNeighborsClassifier, LogisticRegression, Random Forest Classifier and GaussianNB seems to be good models and KNeighborsClassifier and LogisticRegression taken as the best models.

After the feature reduction the accuracy scores given by different models are:

Random Forest Classifier	-- 90%
Support Vector Classifier	--80%
KNeighborsClassifier	--82%
GaussianNB	--88%
Decision Tree Classifier	--81%
Logistic Regression	--90%

From the above scores I can conclude that after feature reduction Random Forest Classifier and LogisticRegression gives best accuracy score which is 90% and it is also greater than the accuracy score given by the benchmark model which is DecisionTreeClassifier. In this scenario KNeighborsClassifier, LogisticRegression, Random Forest Classifier and GaussianNB seems to be good models and Random Forest Classifier and LogisticRegression taken as best models.

Optimized model:

Finally According to my knowledge I considered Logistic Regression model as the best and optimized model to detect the Breast Cancer Wisconsin. It is because It gives the best accuracy score both in the case of with feature reduction and without feature reduction.

Final accuracy score on testing data without feature reduction – 95%

Final accuracy score on testing data with feature reduction – 90%

IV. Results

Model Evaluation and Validation:

```
from sklearn.linear_model  import  LogisticRegression

import pandas as pd

db = pd.read_csv('data.csv')

db.drop("id", axis = 1, inplace = True)

diagnosis_coding = {'M':1, 'B':0}

db.diagnosis = db.diagnosis.map(diagnosis_coding)

#Before feature reduction

X =db[['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean',
'symmetry_mean','fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se',
'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se',
'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst',
'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave
points_worst', 'symmetry_worst','fractal_dimension_worst']]

y = db['Diagnosis']

#Dividing the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)

#try with LogisticRegression

model =LogisticRegression()

model.fit(X_train,y_train)

prediction=model.predict(X_test)

metrics.accuracy_score(prediction,y_test)
0.9532163742690059
```

```
#After feature reduction

X_corr = db[['smoothness_mean', 'radius_se', 'texture_se', 'smoothness_se',
'symmetry_se',  'fractal_dimension_se', 'texture_worst',
'symmetry_worst','fractal_dimension_worst']]

y_corr = db['Diagnosis']

#Dividing the dataset into a separate training and test set:

X_train1, X_test1, y_train1, y_test1 = train_test_split(X_corr,y_corr,test_size=0.3)

#try with LogisticRegression

model =LogisticRegression()

model.fit(X_train1,y_train1)

prediction=model.predict(X_test1)

metrics.accuracy_score(prediction,y_test1)

0.9005847953216374
```

Conclusion:

	Without feature reduction	With feature reduction
Accuracy score of		
Benchmark model	92%	81%
Optimized model	95%	90%

The accuracy score given by the optimized model is greater when compared with Benchmark model both in the cases of without feature reduction and with feature reduction.

From the correlation matrix it can be conclude that 'smoothness_mean', 'radius_se', 'texture_se', 'smoothness_se', 'symmetry_se','fractal_dimension_se', 'texture_worst', 'symmetry_worst','fractal_dimension_worst' are the first nine most predictive features.

Reflection:

1. I have learnt how to visualize, and understand the data.

2. I have learnt that Data Cleaning plays crucial part in Data Analysis.

3. I observe, how the accuracy score given by the different classification models changes when I execute the process of feature reduction.

4. I got to know how to use best technique for the data using appropriate techniques.

5 .On total, I have learnt how to grab a data set and applying cleaning techniques on it and to fit to best techniques to get best scores.

Improvement:

This can be improved to classify a disease in Human Body by increasing the features and number of instances. The model is a part of my research in **Health Care**. This application can be taken to next level with the engagement of Internet Of Things and can be implemented in large scale in Health Care Centers which also reduces the *Human Errors* which is a Serious Problem in Health Care Domain since from past.