ML MAJOR PROJECT – ML10B1 PROJECT SUMMARY

PROBLEM STATEMENT

The aim of this project is to diagnose patients with breast cancer by analysing the data of patients and classifying them into two categories, having diagnosis results as

1. Benign (**B**)
2. Malignant (**M**)

LIBRARIES USED

1. **NUMPY -** The fundamental package for scientific computing in python
2. **PANDAS** – An open source BSD-licenced library providing high-performance, easy-to-use data structures and data analysis tools for python programming language
3. **SKLEARN.MODEL_SELECTION** –
    - Split arrays or matrices into random train and test subsets
    - Selecting optimal features
4. **MATPLOTLIB.PYPLOT** – Provides a MATLAB-like plotting framework
5. **SEABORN** – It is a data visualisation library based on matplotlib. It provides high – level interface for drawing attractive and informative statistical graphics.
6. **SKLEARN.METRICS**
    - Accuracy score : In multilabel classification, this function computes subset accuracy; the set of labels predicted for a sample must exactly match the corresponding set of labels in y
    - Confusion matrix : Compute confusion matrix to evaluate the accuracy of classification

DETAILS OF DATA:

The dataset was collected from Kaggle:

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

No of rows : 569

No of columns : 33

FEATURE DESCRIPTION

1) **ID Number**

    The unique ID allotted to each patient to differentiate between them

2) **Diagnosis**

    The target variable i.e., the conclusion if the tumour is malignant or benign

3) **Columns 3-33**

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

    a) Radius
    b) Texture
    c) Perimeter
    d) Area

e) Smoothness
f) Compactness
g) Concavity
h) Symmetry
i) Fractal Dimension

The mean, standard deviation and worst or largest (mean of three largest values) of these features were computed for each image resulting in 30 features.

For instance, field 3 is texture_mean, field 13 is texture_se and field 23 is texture_worst.

TASKS PERFORMED:

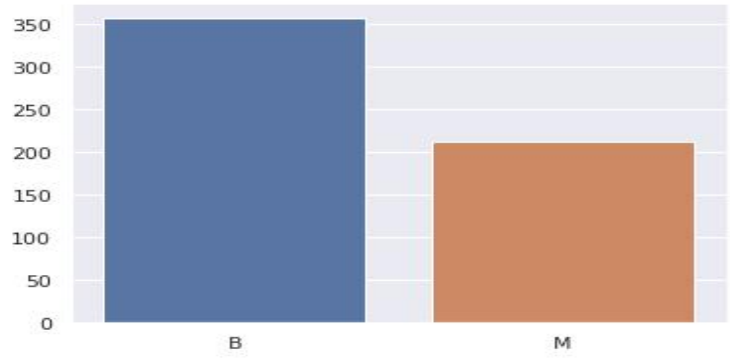
1. Exploratory data analysis
    a) Visualization
    b) Mapping string to numeric data
    c) Checking for duplicate values
    d) Data Cleaning
2. Questions asked on the dataset with answers
    a) What is the average area of a tumour of a person having cancer?
    b) How does the area of tumour impact diagnosis?
    c) How does radius impact diagnosis results?
3. Feature Selection
    a) Using Feature Importance
    b) Using Correlation heatmap
4. Normalising data
5. Ensemble Machine Learning Modelling
    a) K Nearest Neighbors Classifier
    b) Random Forest Classifier
    c) Support Vector Classifier
6. Accuracy Calculation
7. Summary

1. EXPLORATORY DATA ANALYSIS

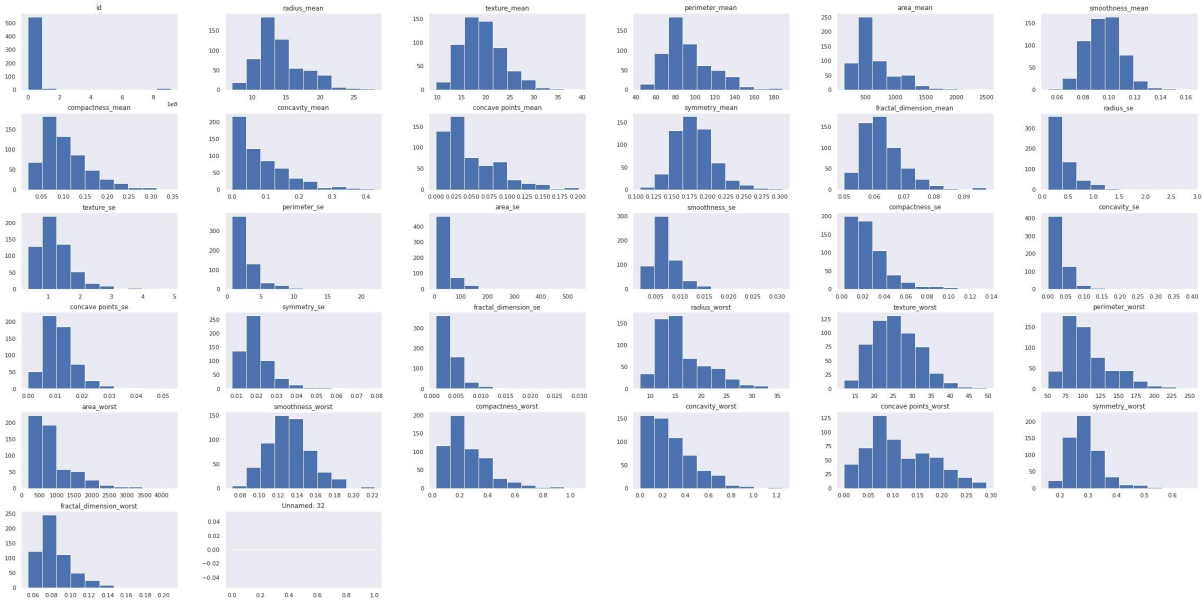
The shape and top columns of the dataset are analysed and checked for the presence of null values

a) Visualisation

The distribution of data with respect to diagnosis categories is visualised using barplot.



The distribution of data based on the frequency of unique values in the features is visualised using histogram.



b) Mapping string to numeric data

The categories M and B of the target variable are mapped to numeric values 1 and 0 respectively in order to ease computation.

c) Checking for duplicate values

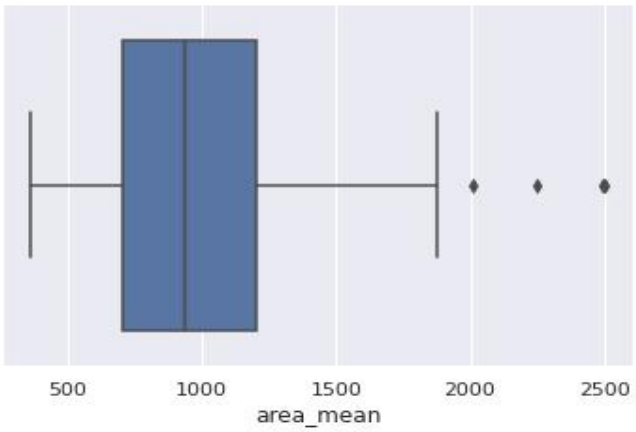
The dataset is checked for the presence of duplicate values if any. No duplicate data is present in our dataset.

d) Data Cleaning

The irrelevant columns analysed previously are dropped from the dataframe to make the model efficient.

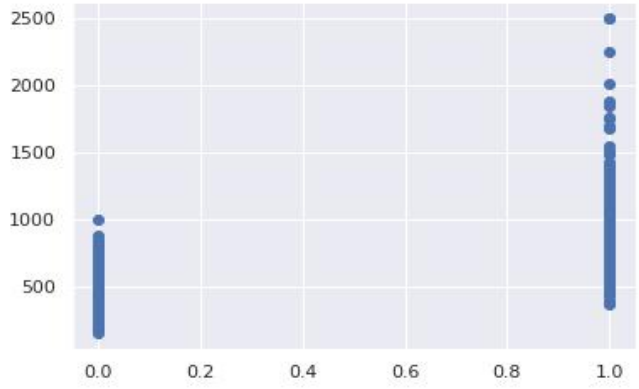
2. QUESTIONS ASKED ON THE DATASET WITH ANSWERS

a) What is the average area of a tumour of a person having cancer?



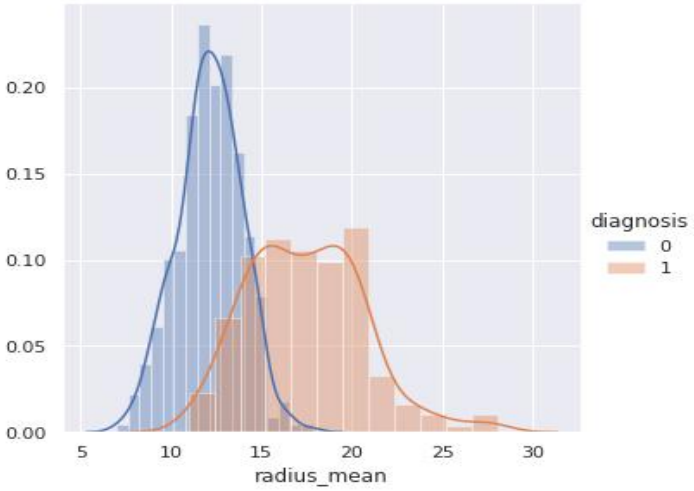
Answer : 952.67211 square units

b) How does the area of tumour impact diagnosis?



Answer : The above graph concludes that the person having large area of tumour is in risk and have to be tested immediately

c) How does the radius impact diagnosis results?



Answer : The person having large radius of tumour is at higher risk

3. FEATURE SELECTION

The features which are highly correlated (show dependency) with the target variable (diagnosis) are highly relevant for our classification problem

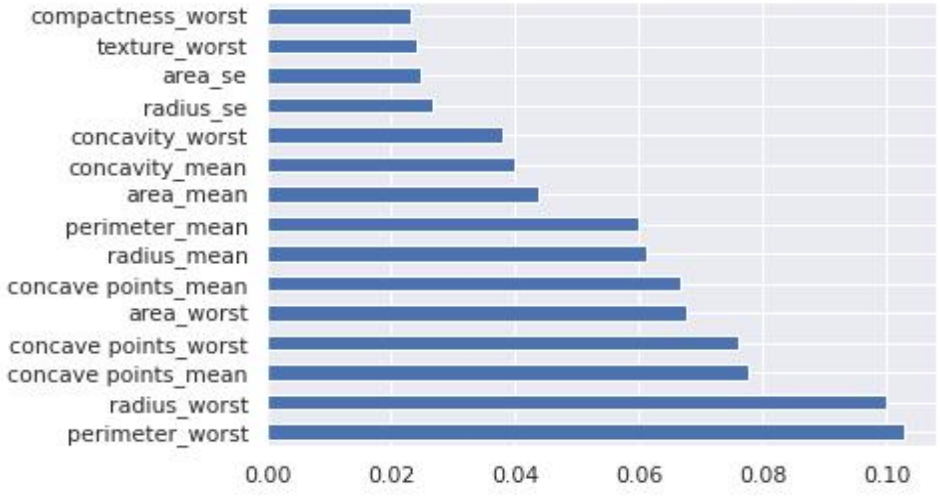
The irrelevant attributes are removed in order to reduce the size of data for easier computation.

Here, Y stores the target or class variable and X stores the non-class attributes.

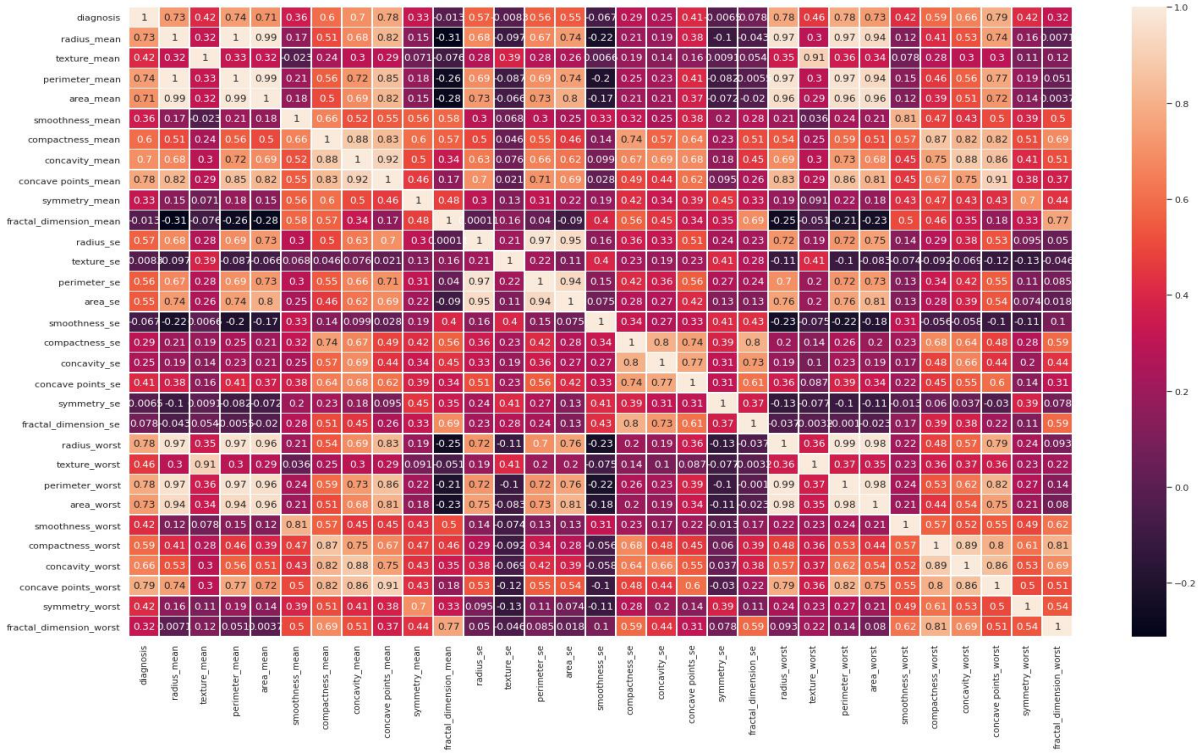
The irrelevant attributes can be found by computing the correlation among the non-class attributes and then we can reduce a subset of highly co-relevant non-class attributes to a single or less number of attributes which would reduce the size of data.

Therefore we create a feature importance plot and heatmap to display the correlation between all the features.

Feature Importance Plot



Correlation heatmap:



This heatmap visualizes the correlation between each pair of attribute in the dataset. We can select those attributes which have high correlation with each other according to a threshold value, and then we will implement the classification algorithm with a reduced set of attributes by taking various smaller subsets of these highly correlated (dependent) attributes and compare the results.

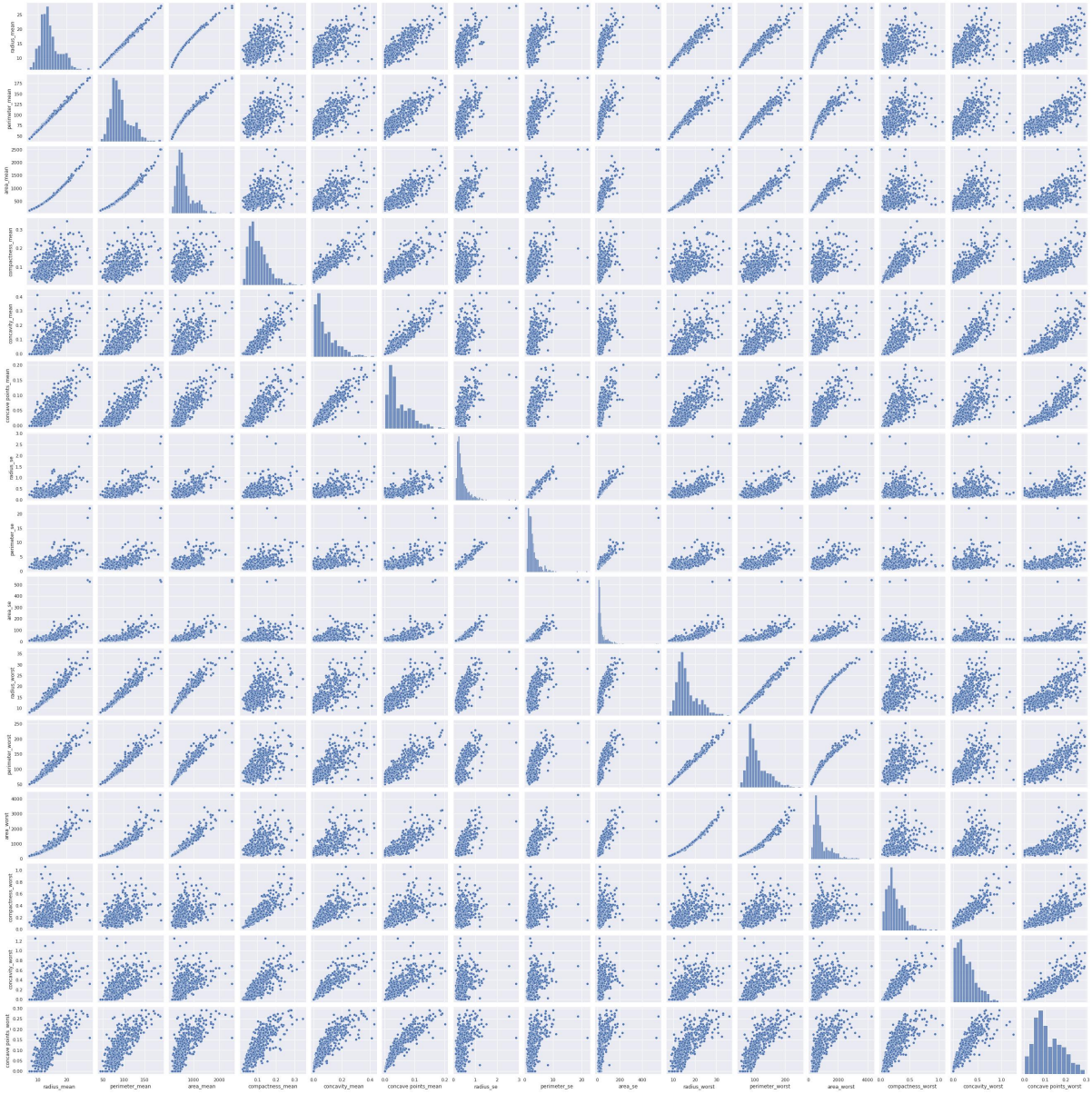
Here, we take threshold value : 0.50

From the heatplot we infer that the following features are the most related (corr > 0.5) to the target variable:

radius_mean float64
perimeter_mean float64
area_mean float64
compactness_mean float64
concavity_mean float64
concave points_mean float64

radius_se float64
perimeter_se float64
area_se float64
radius_worst float64
perimeter_worst float64
area_worst float64
compactness_worst float64
concavity_worst float64
concave points_worst float64

PAIR PLOT FOR HIGHLY CORRELATED FEATURES



4. NORMALISING DATA

The new dataset with selected features is normalised using preprocessing library from sklearn module and the data is split into train and test dataset

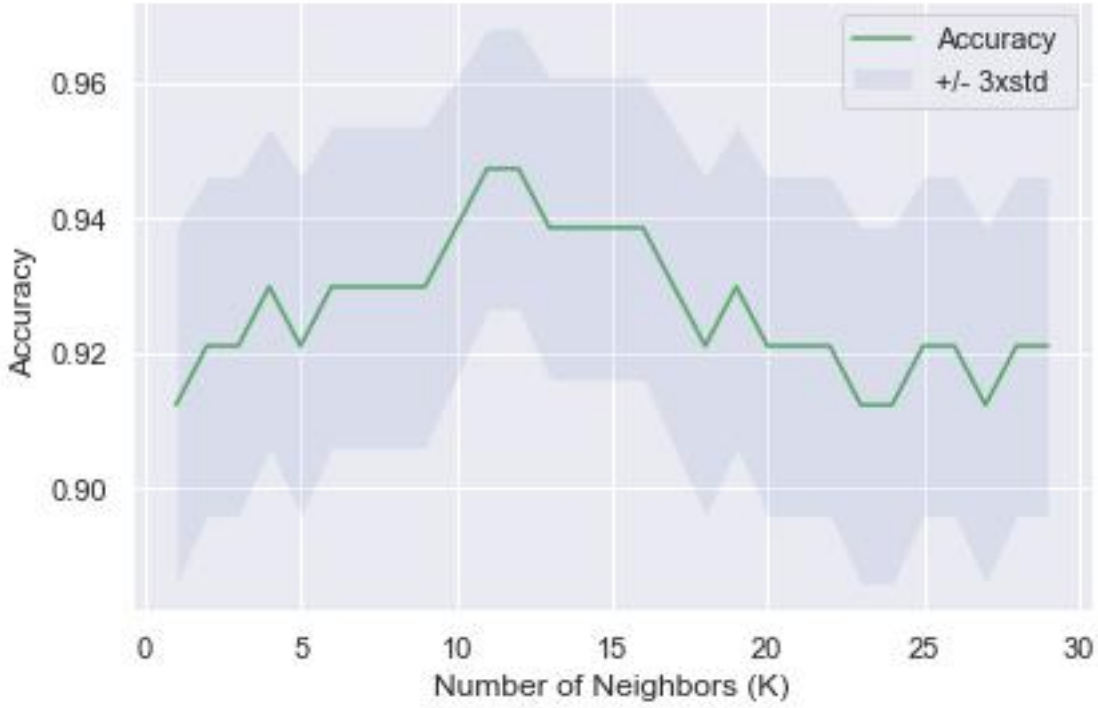
5. ENSEMBLE MACHINE LEARNING MODELLING

a) K Nearest Neighbors Classifier

Initially we took k value as 4

```
from sklearn.neighbors import KNeighborsClassifier
k = 4
#Train Model and Predict
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train,y_train)
neigh
```

We got best accuracy for k=1 below graph demonstrates the scenario




```
print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

```
The best accuracy was with 0.9473684210526315 with k= 11
```

B) Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators= 1000, random_state = 1)
rf.fit(X_train, y_train)
yhat2 = rf.predict(X_test)
```

We got accuracy as following

```
print('The Accuracy of RandomForest Model is',metrics.accuracy_score(y_test,yhat2)*100)
```

```
The Accuracy of RandomForest Model is 90.35087719298247
```

C) Support Vector Classifier

The Last Model We Used is SVM

Support Vector Classifier

```
#support vector classifier
from sklearn.svm import SVC
svm = SVC(random_state = 1)
svm.fit(X_train, y_train)
yhat3=svm.predict(X_test)
```

The Accuracy of SVM we Got is

```
print('The Accuracy of SVM Model is',metrics.accuracy_score(y_test,yhat3)*100)
```

```
The Accuracy of SVM Model is 91.22807017543859
```

Conclusion :-

In this way, I have performed data analysis and have checked the accuracy of 3 different classification algorithm namely KNN, Random Forest and SVM. We Got Best Accuracy For KNN Model (94%)