Data Science Project

*Breast Cancer Data Analysis And Classification.*

Submitted to Dr. Sakthi Balan Muthiah and Dr. Subrat Kumar Dash

**Team Members:**

Piyush Chowdhary 17ucs111

Jatin Kumar 17ucs186

Tanmay Gwalera 17ucc060

Nisarg Jain 17ucc039

INDEX:

| ***Topic*** | ***Page Number*** |
| --- | --- |
| Motivation | 3 |
| Objective | 4 |
| Preliminary Analysis | 4 |
| Approach | 6 |
| Project Workflow | 10 |
| Inference | 27 |
| Conclusion | 28 |
| Reference | 29 |

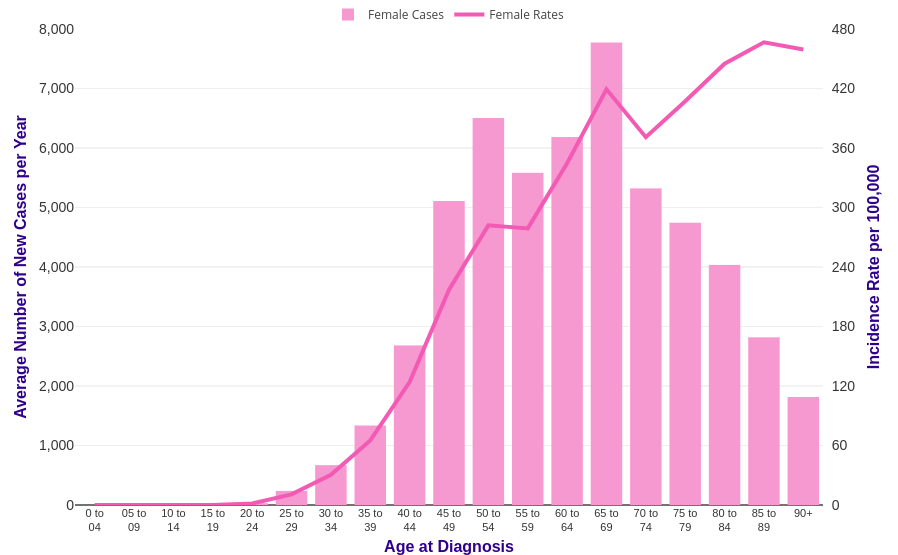
**Motivation**:

Breast cancer is the second most common cancer occurring overall in the world. The major causes include alcohol consumption, greater birth weight, and adulthood height.

Recent studies suggest an increasing trend in the number of people getting

diagnosed with breast cancer. In India, it accounts for about 25% to 32% of all

female cancers in all these cities.



The figure is shown showcases the increasing trends in breast cancer with increasing age in women

In our project, we use the breast cancer data set to understand which attributes that generally lead to this disease and also diagnose a new individual based on the training set so as to save their lives by detecting the disease in early stages.

**Objective:**

To analyze and study the breast cancer data set and use various machine learning algorithms to classify records as **benign** or **malignant**.

**Preliminary Analysis:**

*About the Data Set*:

Characteristics are calculated from a digitized image of a fine needle vacuum (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image. n three-dimensional space is described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

The data set is available on the UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

*Attribute Information****:***

1) ID number

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

3) The rest of the columns 3-32 consists of information about the following features.

Ten real-valued features are computed for each nucleus.

* **Perimeter**
* **Area**
* **Symmetry**
* **Radius**: Mean of the distance from the centre to the perimeter.
* **Texture**: Standard deviation of gray-scale values
* **Smoothness**: Local variation in radius length
* **Compactness** :perimeter^2/area-1
* **Concavity**: Severity of concave portions of the contour
* **Concave Points**: Number of concave portions of the contour
* **Fractal Dimension**: Coastline approximation -1

Each feature contains 3 attributes in the data set: **mean** , **standard error** and **worst**.

*Data Preprocessing***:**

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Dataset contains incomplete data points.They also lack in certain behaviors or trends and contains many errors. Converting these data into a format that the predictor can understand in called pre processing.

**The dimensionality of the feature set**: 569x30.

All feature values are recorded with **four** significant digits.

**Missing attribute values**: none.

**Class distribution**: 357 benign, 212 malignant.

A. We have removed some redundant columns like:

1. **Id** column which does provide any information that helps us classifying the data
2. **Unnamed32** column which only consisted of NaN values
3. We have also separated the **diagnosis** column as it serves as our class label

B. Then we checked that if our data consisted of any NULL values. Luckily for us there were **no null** values in our dataset.

C. Further we have analyzed the mean, standard deviation and other descriptive statistics for the data to get some oversights about the data.

D. We dove further in our analysis by visualizing various kinds of plots:

1. **Count Plots** - To compare the number of each class label ‘Benign’ and ‘Malignant’
2. **Density Plots** - To visualize the distribution of data
3. **Heatmaps** - To determine correlation between different attributes
4. **Pairplots** - To visualize comparison between the pair of attributes which gives us the most separated data (the best margin for classifier)

**Approach:**

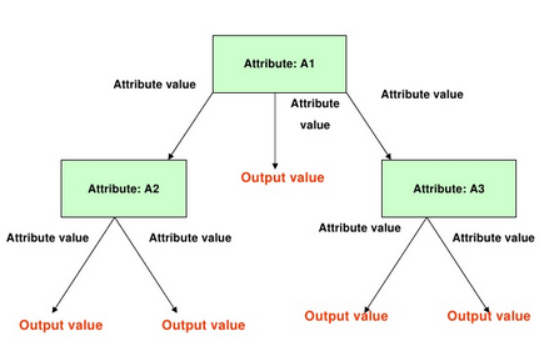
*Decision Tree*:

Decision trees are one of the most practically used methods for supervised learning. They are used for both classification and regression tasks. The basic working ideology is based on predicting the value of a target variable by learning simple decision rules from the training set. They are generally in the form of if-else statements and deeper we go into the tree the rules become more complex and the model becomes fitter.

It’s basically a tree data structure with nodes representing an attribute which causes the split when asked a question. The edges represent answers to questions and leaf nodes represent the output class labels. The entire process is recursive and is repeated for every subtree node.

The attribute from which split occurs depends on measures like GINI Index, Gain, Entropy etc.

The following is a diagrammatic representation of a decision tree:



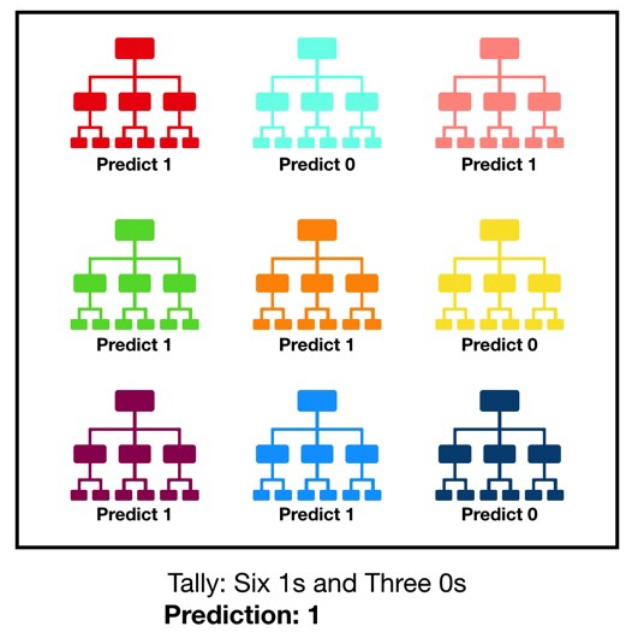
*Random Forests*:

The Random forest classifier that consists of a large number of decision trees that operate as an ensemble. Each decision tree classifies an entity and the final class prediction is done by majority voting.

Requirements for improving accuracy of Random Forest Classifier:

* Features that have some predicting power so that models built do something better than random guessing.
* Trees of the forests and their outcomes are uncorrelated.

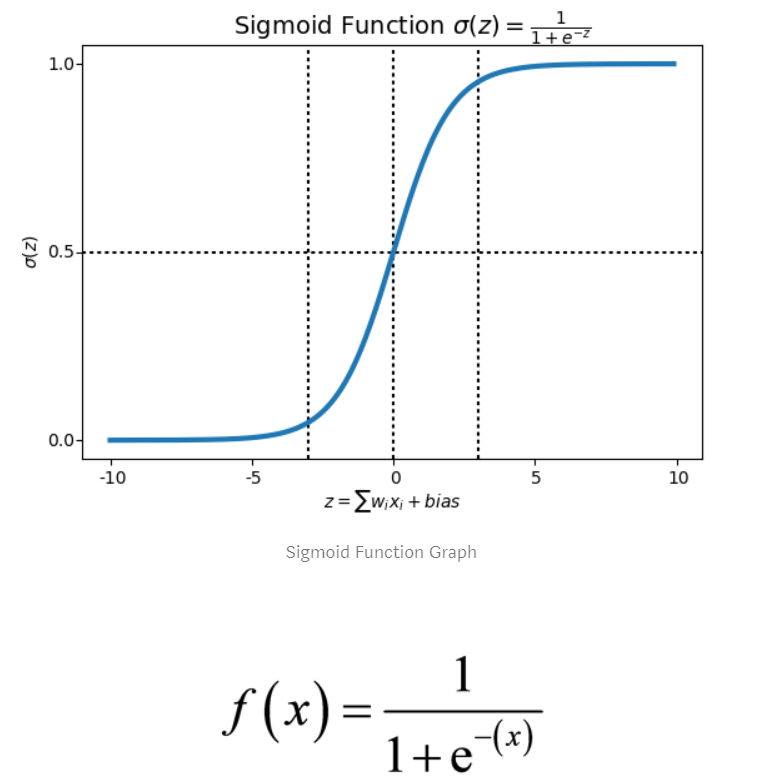
Following diagram shows a random forest example:



*Logistic Regression*:

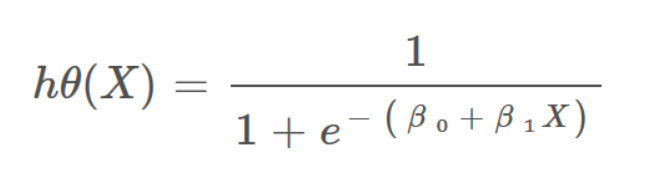
Logistic Regression is a classification algorithm that is used to assign observations to a discrete set of classes. It is basically a linear regression model which uses a complex cost function called the **“Sigmoid function”** , which is used to limit the cost function between 0 and 1.

Plot of sigmoid function:



Generally, for a 2-Class problem, the threshold of the prediction function is 0.5. Based on this, if the prediction function returns a value less than 0.5 then it is classified to class-1 else to class-2.

**Logistic Regression Hypothesis:**



Why we have used the aforementioned algorithms?

* Decision Tree:

It is the most fundamental and widely used algorithm for data classification.It is comparatively less complex and computationally advantageous compared to other algorithms. Despite being simple it gives us respectable accuracy and precision.

* Random Forest:

This one is an ensemble method like AdaBoost but while AdaBoost uses data distribution manipulation, Random forest uses input feature manipulation. Essentially we create various decision trees as our classifiers and then we take weighted average to obtain a classifier with much higher accuracy.It is highly effective for high dimensional and sparse data.

* Logistic Regression:

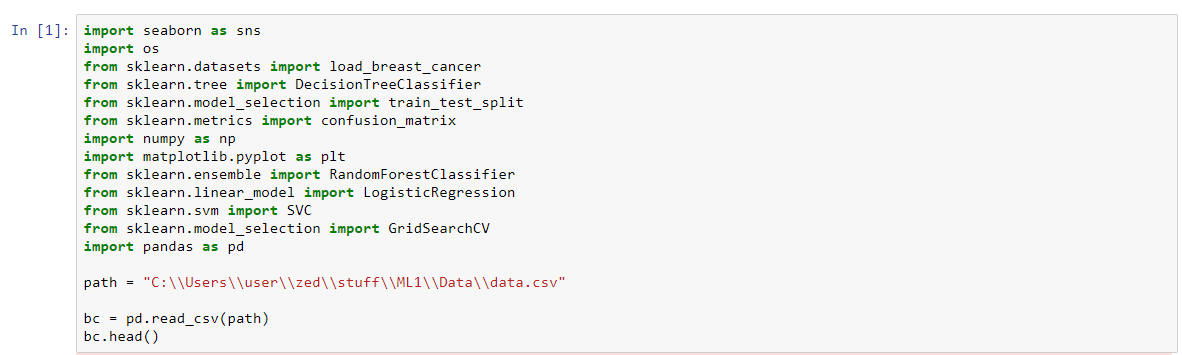
Logistic regression provides us with probabilities of the class labels with the help of logit function. The logit function is then used as an input to a linear regression curve. Since we are using linear regression we can use it for higher dimensionality data such as we currently have.

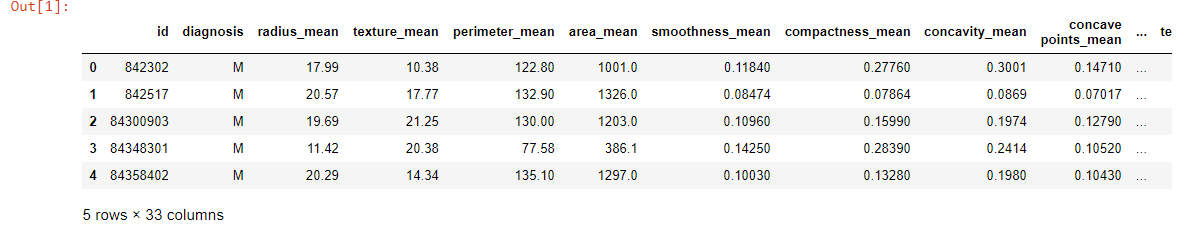
Note:

We could have also used Support Vector Machines but due to the high dimensionality of our data we will not be able to produce a decent classifier.

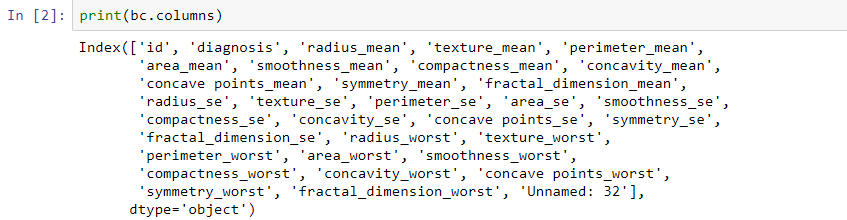
**Project Workflow**:

* Importing the required libraries and first visualization of data:

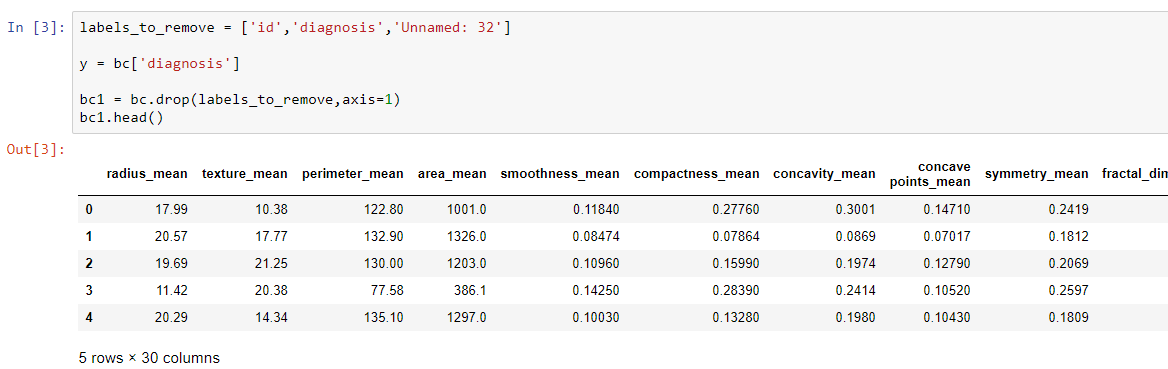


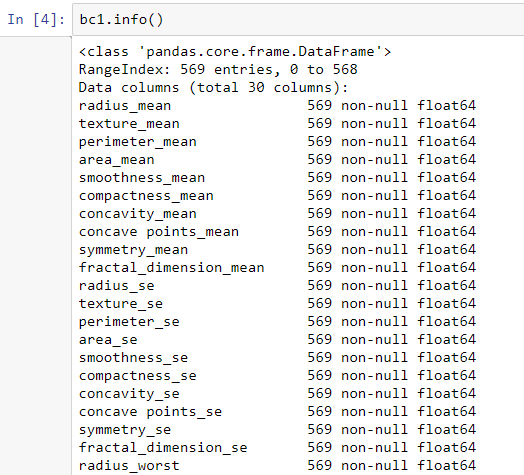


* Seeing the total number of columns:

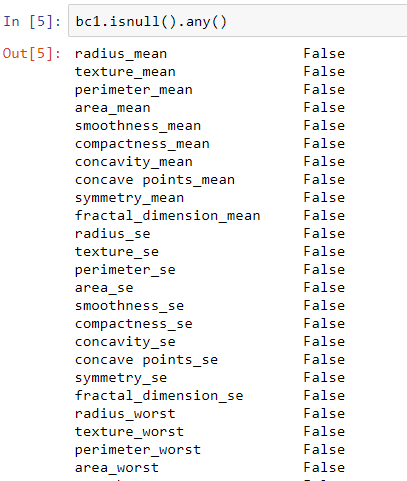


* Removing redundant columns such as ‘Id’ and ‘unnamed 32’, also creating a separate series for the response variable ‘diagnosis’ and finally viewing the cleaned data set :



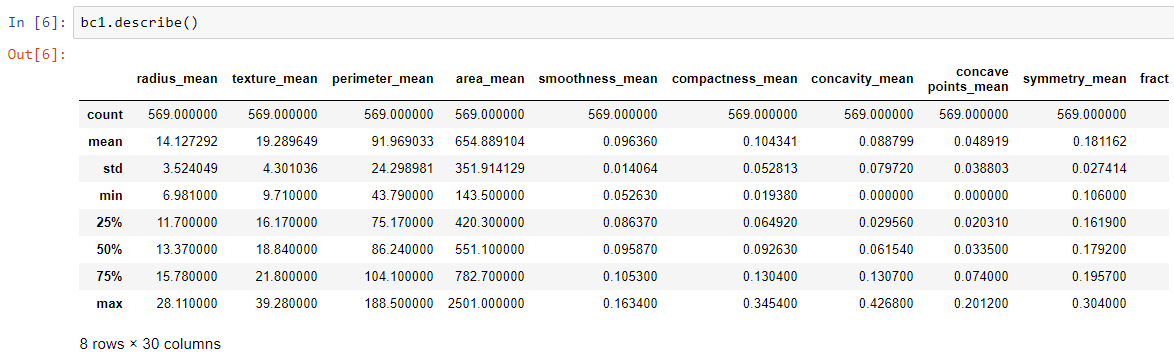


* Determining if any null is present:



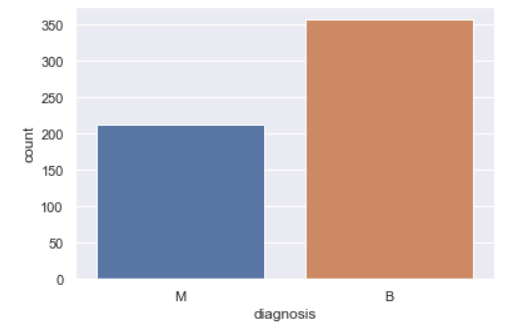
***Wow!! No null values.***

* DataFrame.describe() gives us a summarization of the data frame such as inter-quartile values,min-max ,standard deviation etc.

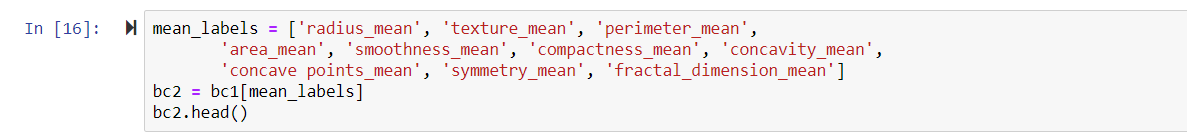


* Here we are using the seaborn library which gives us great plots: As a starter, we are displaying the counts of our class variables

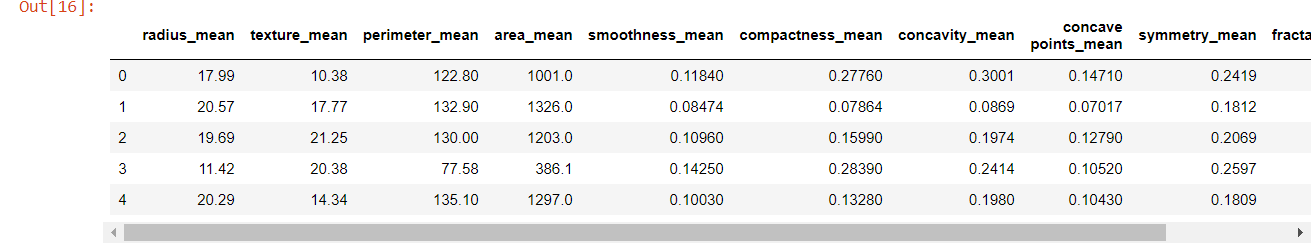




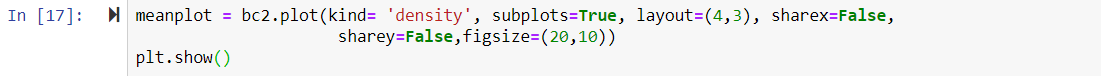
* Now we group our data in three sets viz ‘mean’,’se’ and ‘worst’ , which have about 10 columns each:

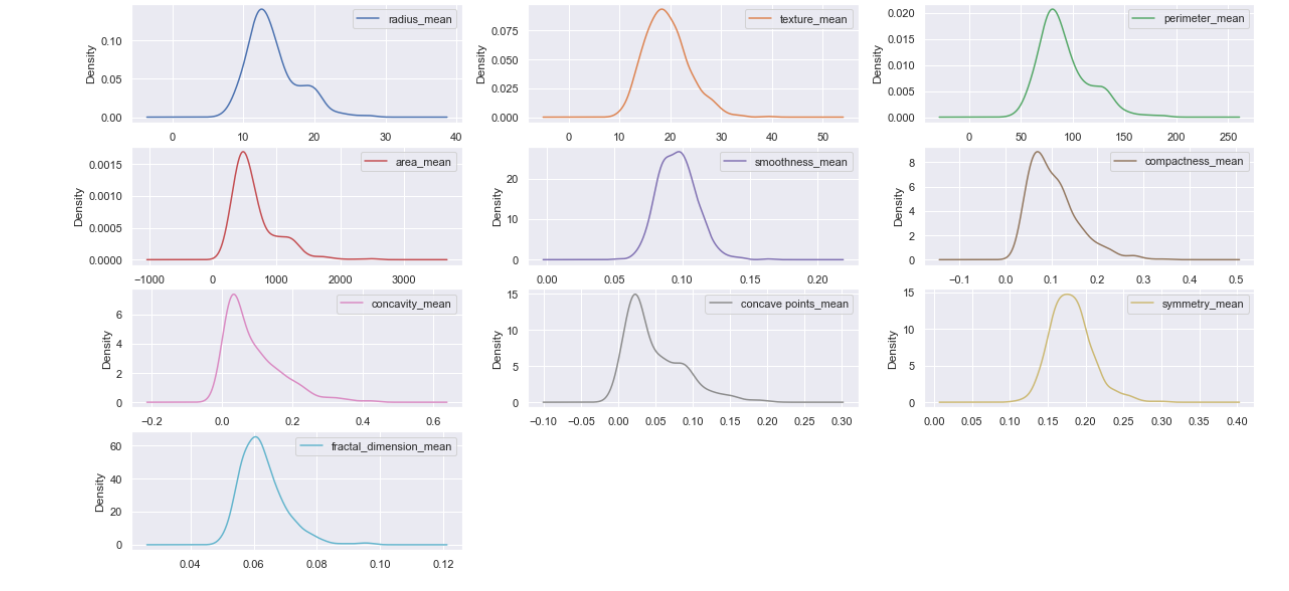


* Displaying the ‘mean’ data frame, capturing the mean of cancer’s attributes:

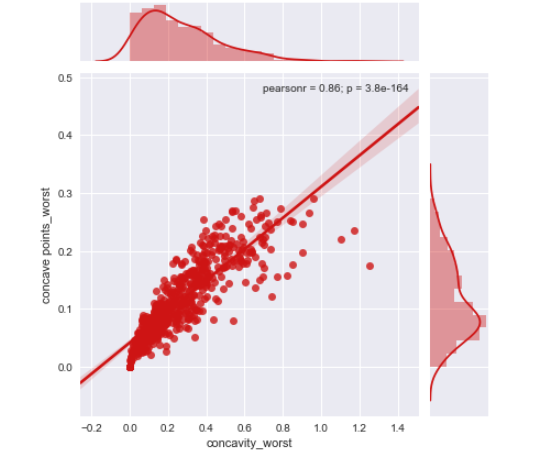


* Subplots of mean attributes:

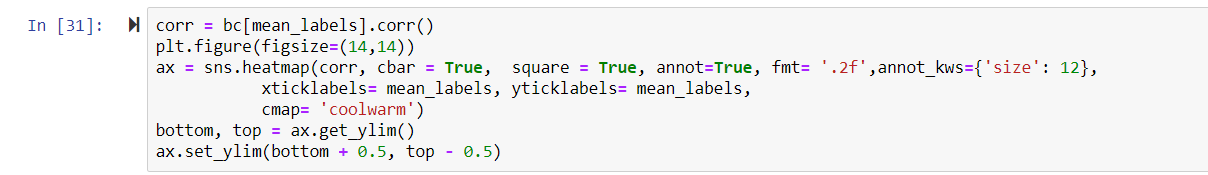






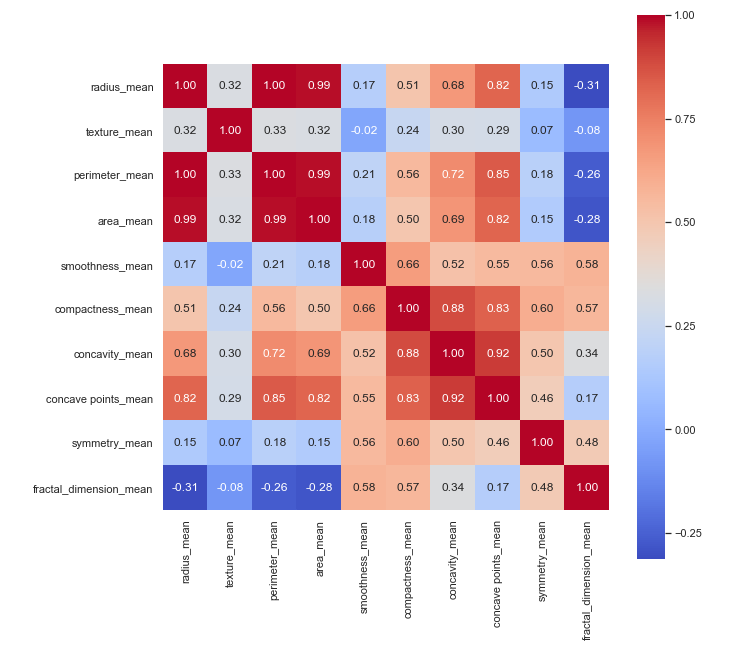


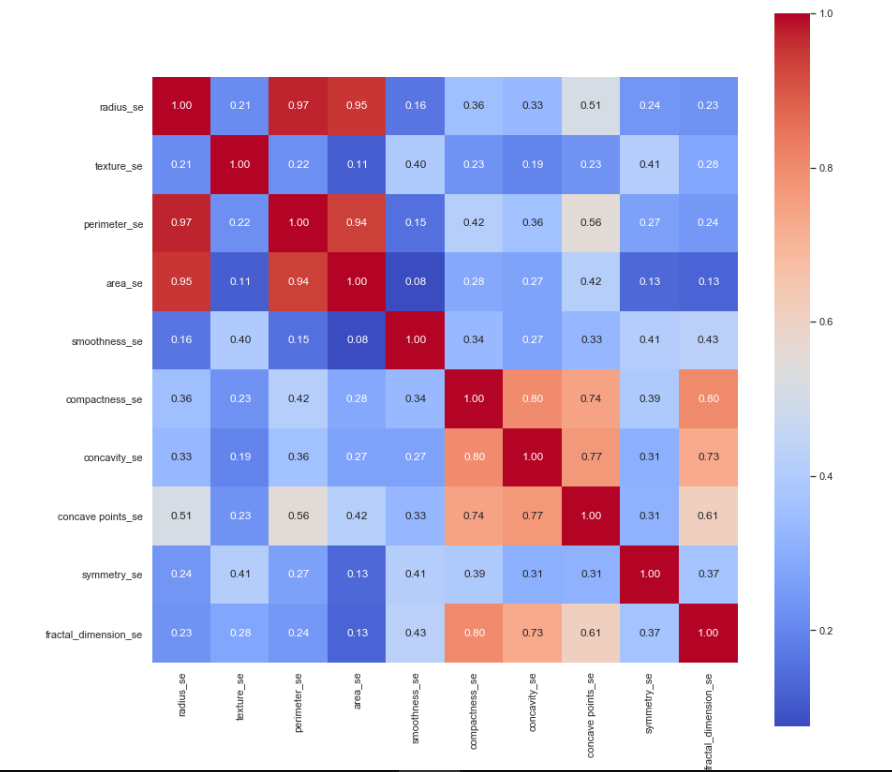
* Now we start to plot the heat map, for that first we calculate the correlation matrix of the ‘mean\_labels’, then using sns.heatmap() to display the plot:

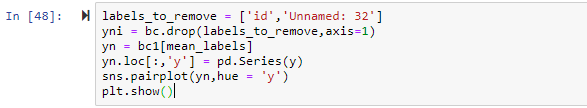


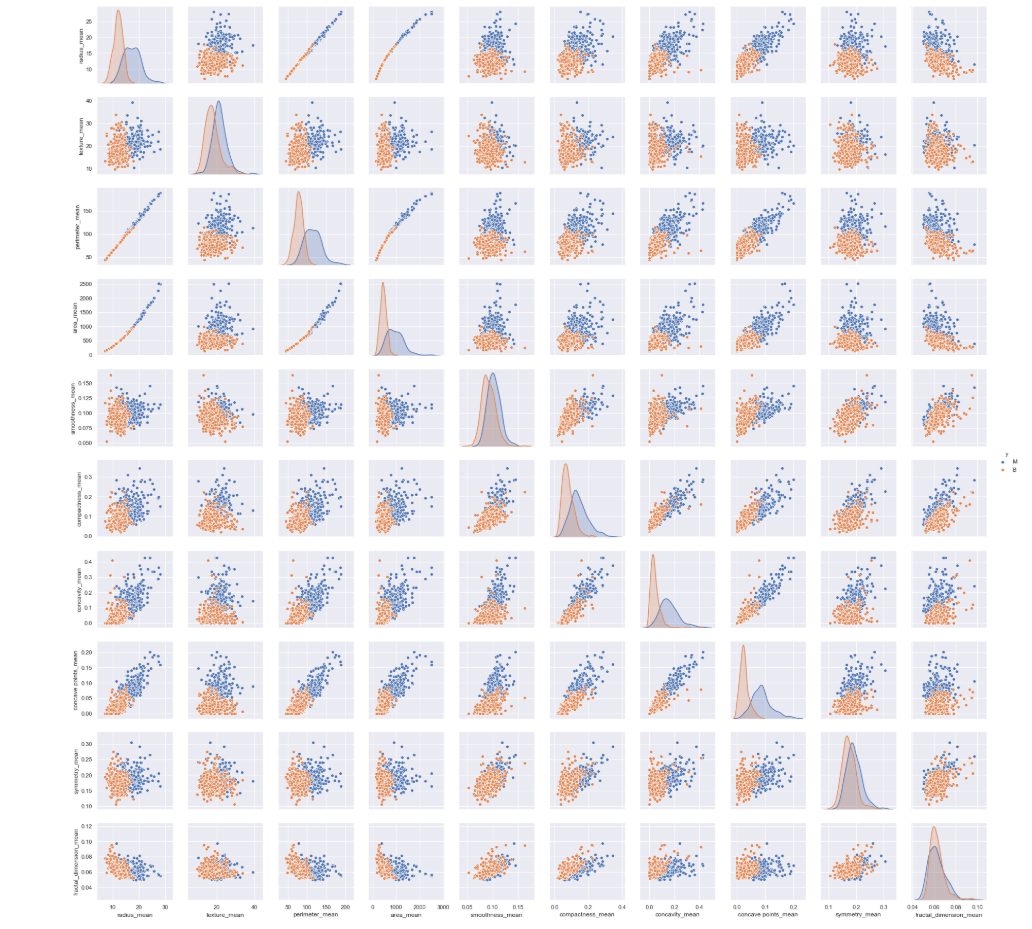
* As we can see, we have some variables that have a very high correlation (as high as1).

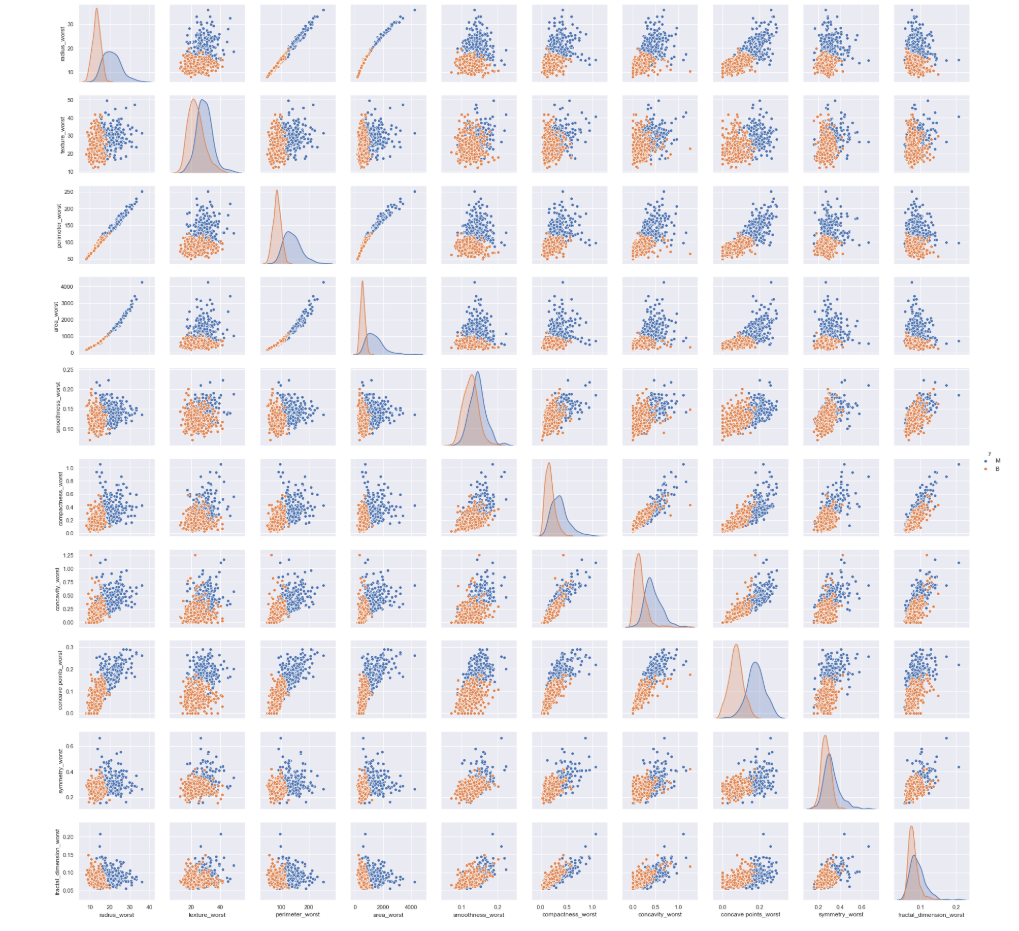
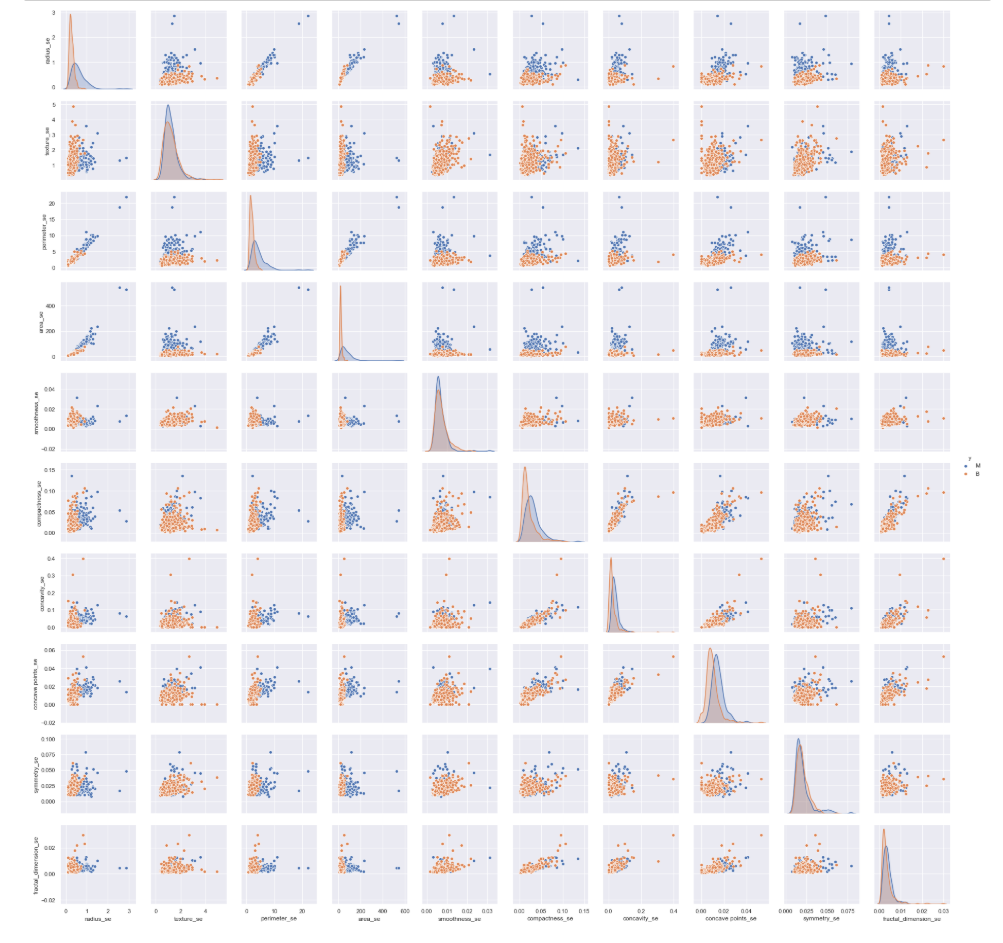
This can also be interpreted theoretically as radius, perimeter and area are all interrelated and will have correlation 1.



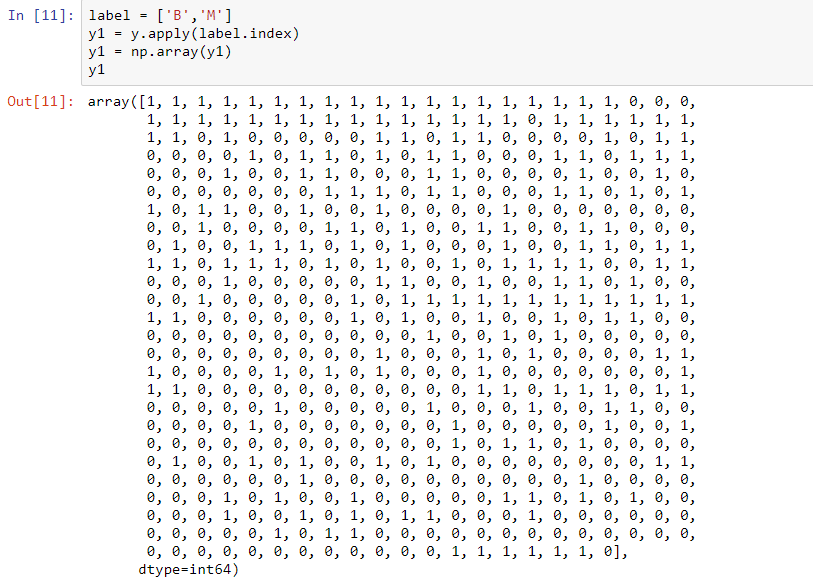
* Repeating the same code as above, we plot the heat map for ‘se’ and ‘worst’ sets:
* Now we use pairplot() to pair plot our attributes:



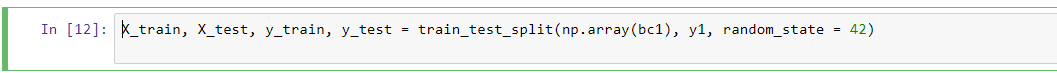
Pair plot of ‘mean’ labels, ‘se’ labels and ‘worst’ labels:



* Converting the Class labels (‘Benign’ and ‘Malignant’) to binary values to assist with classification and accuracy measurements



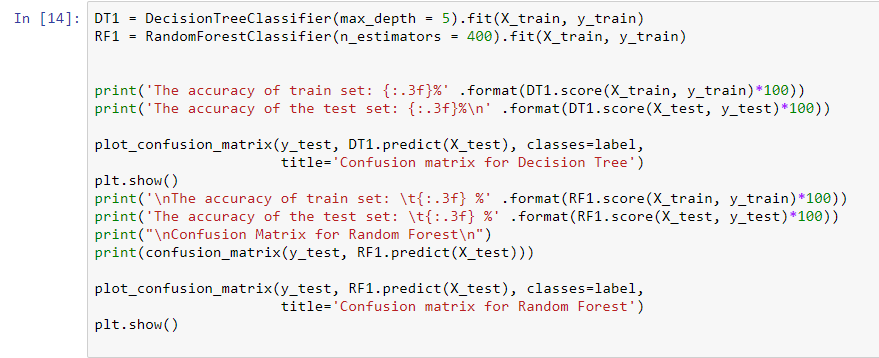
* Dividing the data into Training and Testing sets in the ratio of 7:3 i.e. 70% training samples and 30% test samples.

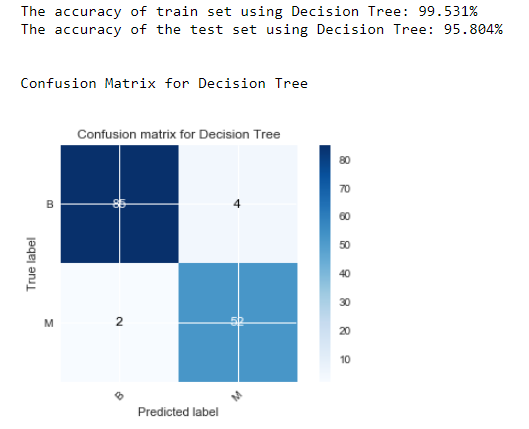


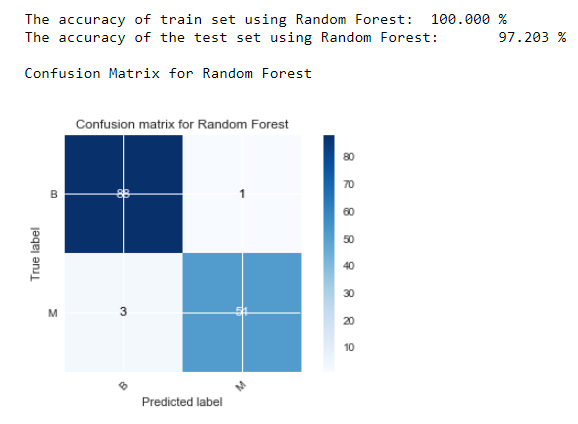
* Defining Confusion Matrix Plot function to visualize the Confusion Matrix



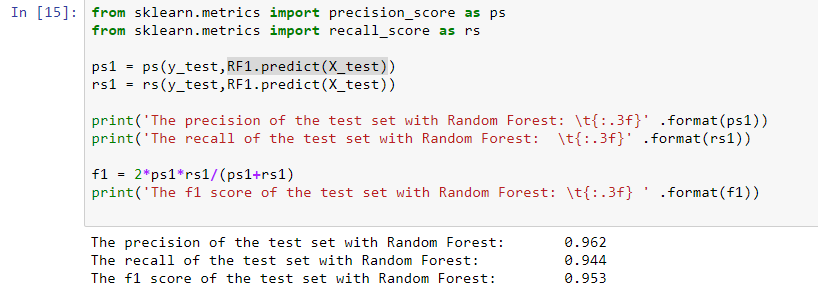
* Applying Decision Tree Classifier and Random Forest Classifier on the data and visualizing the Confusion Matrix for each





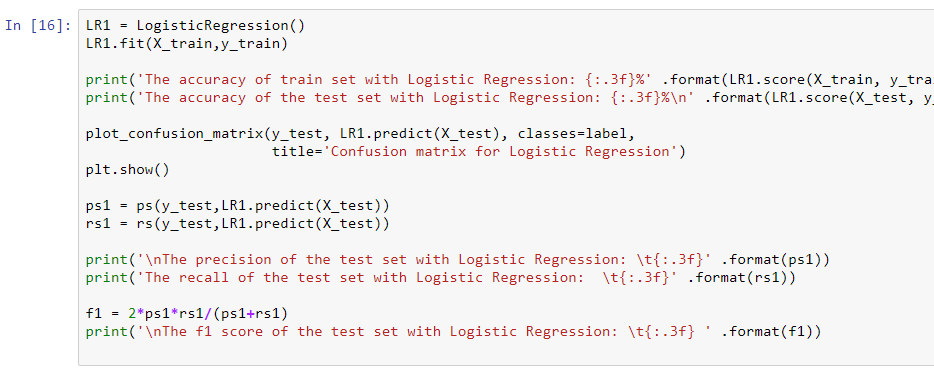


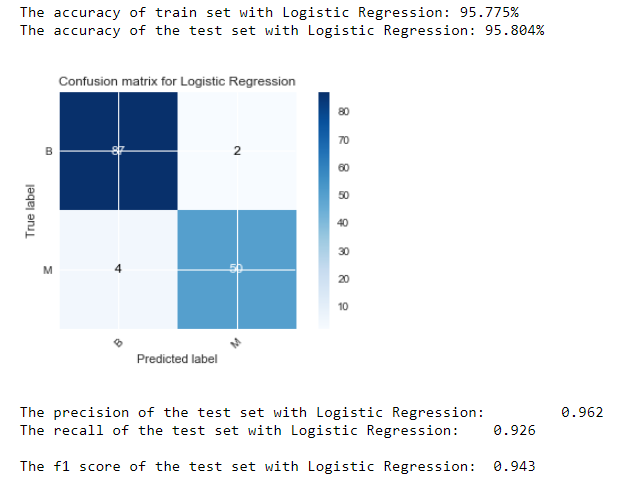
* Calculating the precision, recall and F1 Score for Random Forest Classifier



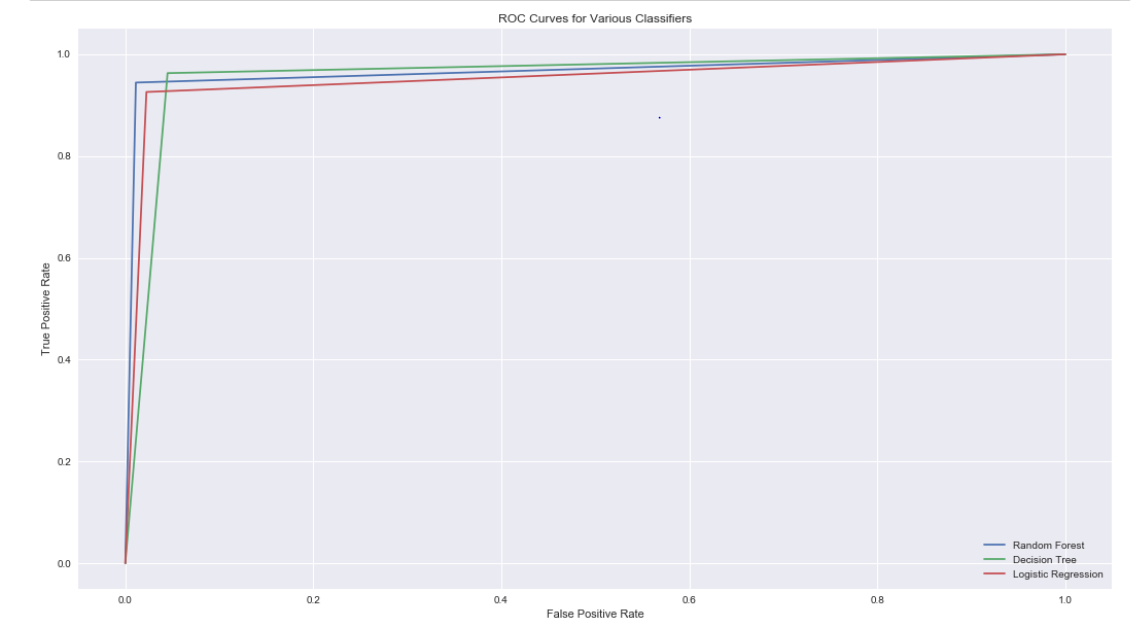
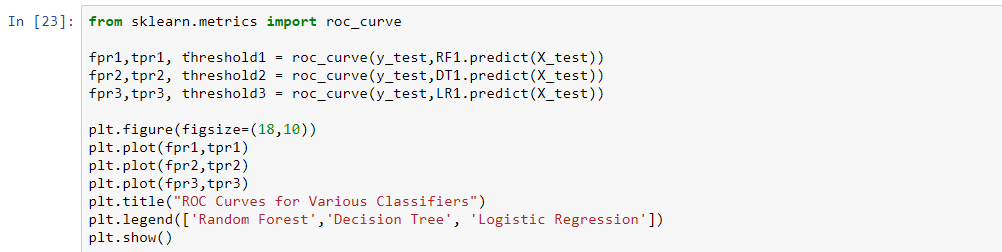
* Applying Logistic Regression on the data and visualizing the Confusion Matrix

And also calculating the precision, recall and F1 Score





* Comparing various classifiers by ROC curves visualizing maximum AUC (area under the curve)

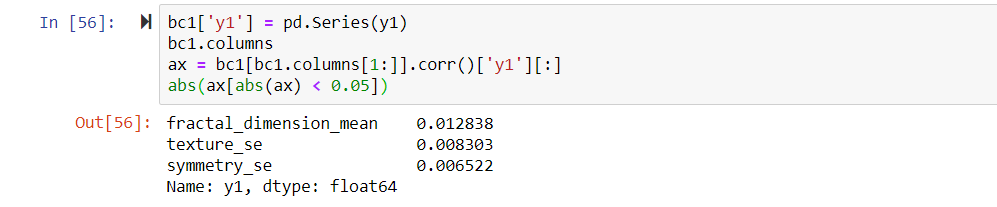


**Inference**:

Here first we have binarized class labels to 0,1 in y1. Then, calculated correlation of

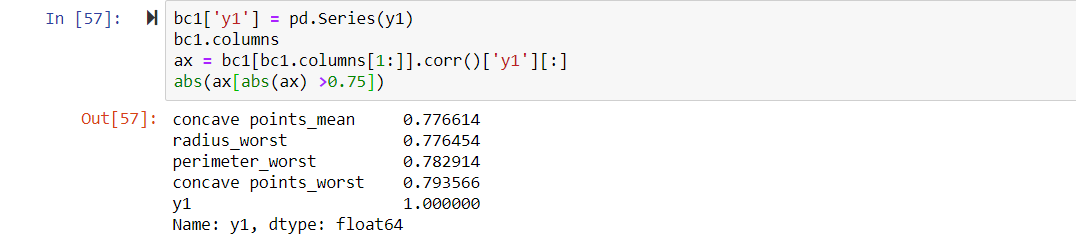
all the attributes to class labels. After filtering, we have found that 3 attributes have

Minimal correlation with the class labels.



Similarly, calculating attributes that have high correlation with class labels, we get,

Concave points\_worst with max correlation.



By comparing the AUC (Area under the Curve) of ROC (Receiver Operating Curve) we can determine which Machine Learning Algorithm will work best for this particular situation.

**Conclusion**:

Being our first Data Science project we tried and successfully built classifiers for the breast cancer dataset using various machine learning algorithms like Decision Tree, Random Forests and Logistic Regression and compared their accuracy, precision , recall and F1 score.

We tried to analyse the attribute relationships using various graphs like count plots, density plots and heat maps etc.

Finally we used ROC curves to compare the various classifiers by maximising the AUC(Area Under the Curve).

The link for this project on github

<https://github.com/TagJones515/First-Repo/blob/master/IDS2.ipynb>

We would like to express our sincere gratitude to Dr. Sakthi Balan M. and Dr.Subrat K. Dash for providing us an opportunity to use our theoretical knowledge to get a practical insight and use them to build our first Data Science project.

**References**:

* <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>
* <https://matplotlib.org/3.1.1/tutorials/index.html> : Documentation for building graphs.
* <https://seaborn.pydata.org/introduction.html> : Built on matplotlib for building graphs.
* [https://numpy.org](https://numpy.org/) : Python package for faster multi-dimensional arrays and matrices
* [https://pandas.pydata.org](https://pandas.pydata.org/) : Python package for data manipulation and analysis.
* [https://scikit-learn.org/stable](https://scikit-learn.org/stable/) : Python package for machine learning algorithms.