

**Student Name:** Madduri sai durga rama prasad

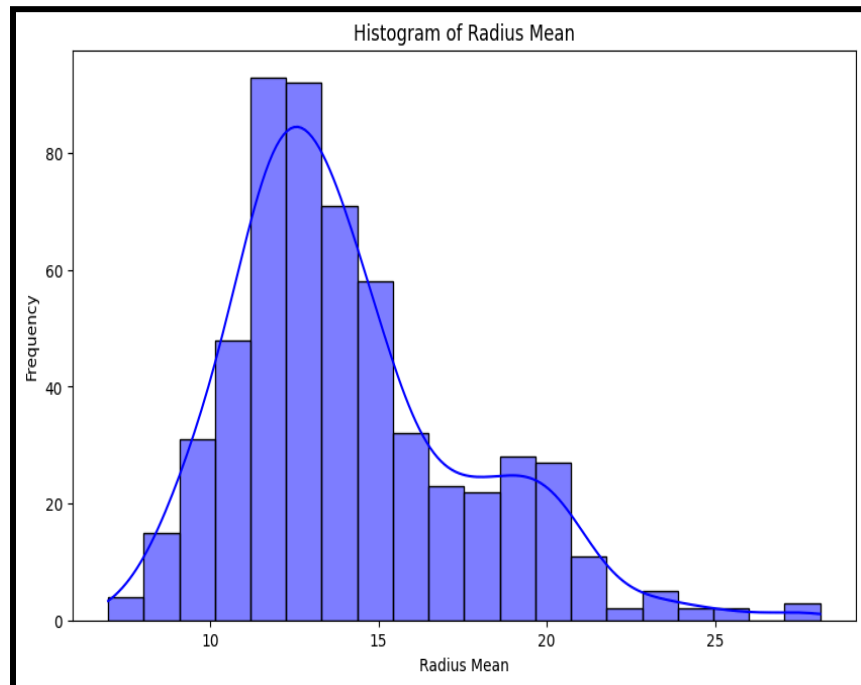
**Student ID:** 23091335

**Git Hub:** <https://github.com/Ramaprasad89/Madduri-sai-durga-rama-prasad-.git>

### Introduction

By using Python, breast cancer predictions depend upon machine learning algorithms analyzing medical data and the patterns associated with the disease. Predictive models classify whether it is a malignant or benign tumor using datasets with patient data such as age, tumor size, biopsy results, etc. Then, it helps in early detection, leading to a much better treatment result and an increase in survival rate among patients.

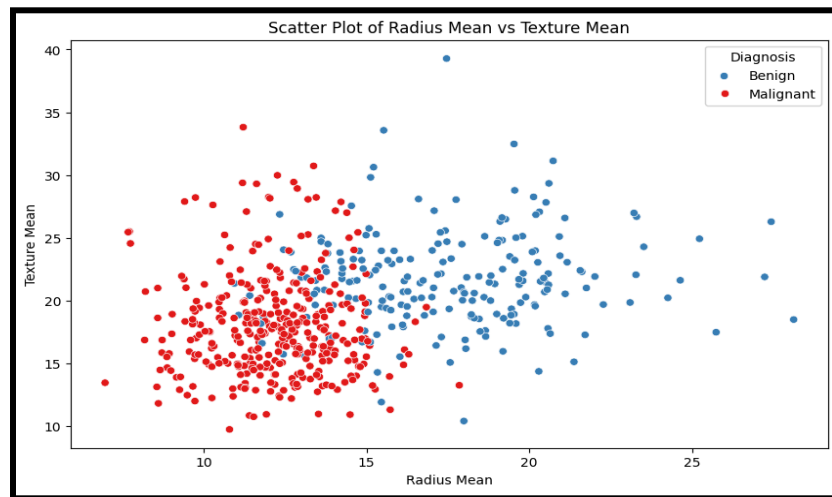
### Histogram of Radius Mean



**Figure 1: Histogram plot**

The above graph represents a histogram for the 'radius\_mean' feature of the data using histplot from Seaborn. It can be used to denote values by distribution with a KDE curve to show the underlying distribution of data. The histogram in this code has 20 bins and blue color axes have been labeled, giving a better insight of the frequency of different radius mean values.

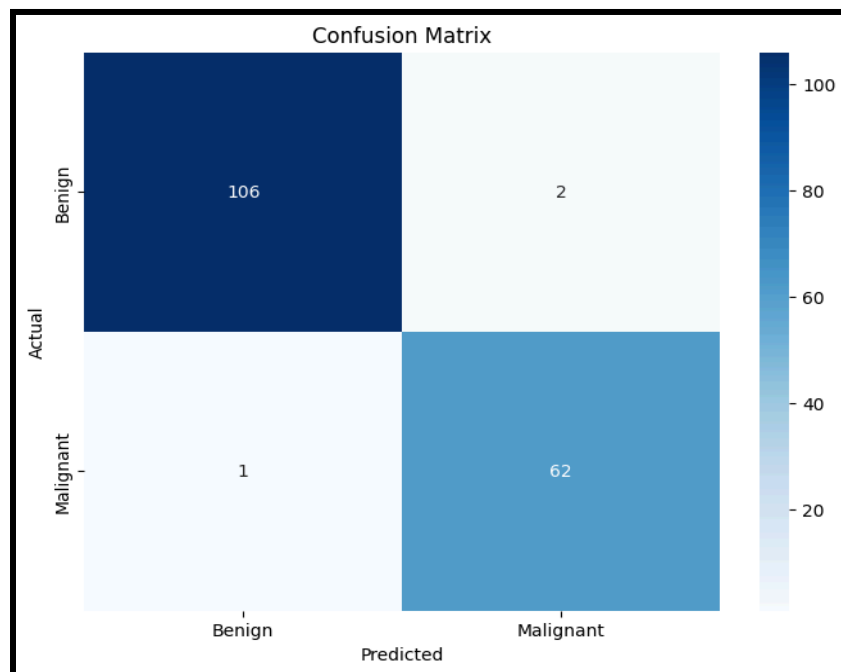
## Scatter Plot of Radius Mean vs Texture Mean



**Figure 2: Scatter plot**

The above graph generates a scatter plot that is used to visualize the relationship between 'radius\_mean' and 'texture\_mean' features. Points are colored according to the 'diagnosis', Benign or Malignant, using Seaborn's scatterplot. The plot includes a legend that describes the diagnosis categories; the axes are labeled with the corresponding feature values so that it can be explored whether these two variables are correlated and their classification.

## Confusion Matrix

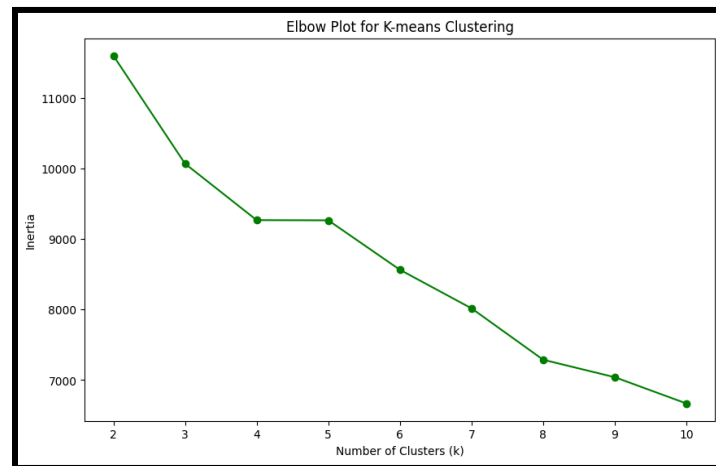


**Figure 3: Application of confusion matrix**

The code computes a confusion matrix using the actual labels, `y_test`, and predicted labels, `y_pred`. A heatmap of the matrix is constructed with annotations showing the counts of the cell using Seaborn. The matrix color scheme is set to use a blue palette, along with axis labels for

'Benign' and 'Malignant'. This can facilitate further evaluation of the classifier's classification performance.

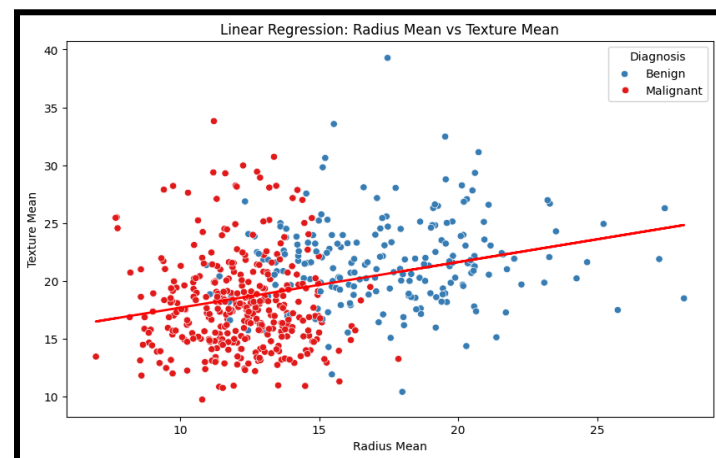
### Elbow Plot for K-means Clustering



**Figure 4: Elbow plot**

The code carries out K-means clustering for the range of cluster numbers 2 to 10 on the scaled data `X_scaled`. It then computes inertia for each run of clustering and saves the results. Finally, the elbow plot is generated based on the number of clusters versus inertia. The "elbow" determines the optimal number of clusters as the point where inertia is decreasing the least.

### Linear Regression: Radius Mean vs Texture Mean (line-fitting)



**Figure 5: Linear regression**

First, a scatter plot is created that relates 'radius\_mean' and 'texture\_mean', while coloring with 'diagnosis'. This step plots a linear regression using the model made by Scikit-learn's `LinearRegression`. It will also train the model using 'radius\_mean' as its predictor and 'texture\_mean' as the target for illustration. A red regression line is plotted in this manner showing the linear relationship of 'texture\_mean'. Included in the plot are categories within the legend of 'diagnosis' to explain how one may discern the trend in this data.