

American International University - Bangladesh

Introduction to Data Science [A]

Final-Term Project Report

**Submitted to -**

Tohedul Islam Sir

Dataset – Breast Cancer

**Submitted by –**

FARUQUE, SAKIF AL

20-42308-1

BSc. in CSE

**Dataset Description –**

Breast cancer is the most common cancer among women in the world. It accounts for 25% of all cancer cases and affected more than 2.1 million people in 2015 alone. It begins when breast cells begin to grow out of control. These cells usually form tumors that can be seen on X-rays or felt as lumps in the breast.

The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign(non-cancerous). Here, 'diagnosis' is the target feature/attribute whose value is to be defined as malignant or benign and the rest of the features will be used for classifying that target attribute.

For classification, the KNN algorithm is used. After building the model, accuracy is checked with the dividing and 10-fold cross-validation approaches. Finally, the confusion matrix will be built and precision & recall values will be calculated.

List of feature description is below,

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Expected Value** |
| **Id** | Unique ID | Serial values |
| **Diagnosis** | Target | M - Malignant B - Benign |
| **radius\_mean** | Radius of Lobes | Decimal |
| **texture\_mean** | Mean of Surface Texture | Decimal |
| **perimeter\_mean** | Outer Perimeter of Lobes | Decimal |
| **area\_mean** | Mean Area of Lobes | Decimal |
| **smoothness\_mean** | Mean of Smoothness Levels | Decimal |
| **compactness\_mean** | Mean of Compactness | Decimal |
| **concavity\_mean** | Mean of Concavity | Decimal |
| **concave points\_mean** | Mean of Cocave Points | Decimal |
| **symmetry\_mean** | Mean of Symmetry | Decimal |
| **fractal\_dimension\_mean** | Mean of Fractal Dimension | Decimal |
| **radius\_se** | SE of Radius | Decimal |
| **texture\_se** | SE of Texture | Decimal |
| **perimeter\_se** | Perimeter of SE | Decimal |
| **area\_se** | Are of SE | Decimal |
| **smoothness\_se** | SE of Smoothness | Decimal |
| **compactness\_se** | SE of compactness | Decimal |
| **concavity\_se** | SEE of concavity | Decimal |
| **concave points\_se** | SE of concave points | Decimal |
| **symmetry\_se** | SE of symmetry | Decimal |
| **fractal\_dimension\_se** | SE of Fractal Dimension | Decimal |
| **radius\_worst** | Worst Radius | Decimal |
| **texture\_worst** | Worst Texture | Decimal |
| **perimeter\_worst** | Worst Permimeter | Decimal |
| **area\_worst** | Worst Area | Decimal |
| **smoothness\_worst** | Worst Smoothness | Decimal |
| **compactness\_worst** | Worse Compactness | Decimal |
| **concavity\_worst** | Worst Concavity | Decimal |
| **concave points\_worst** | Worst Concave Points | Decimal |
| **symmetry\_worst** | Worst Symmetry | Decimal |
| **fractal\_dimension\_worst** | Worst Fractal Dimension | Decimal |

The libraries which are used for this project,

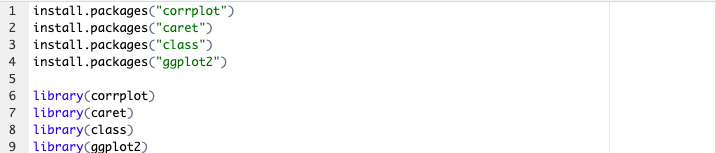


Figure : Libraries & Modules

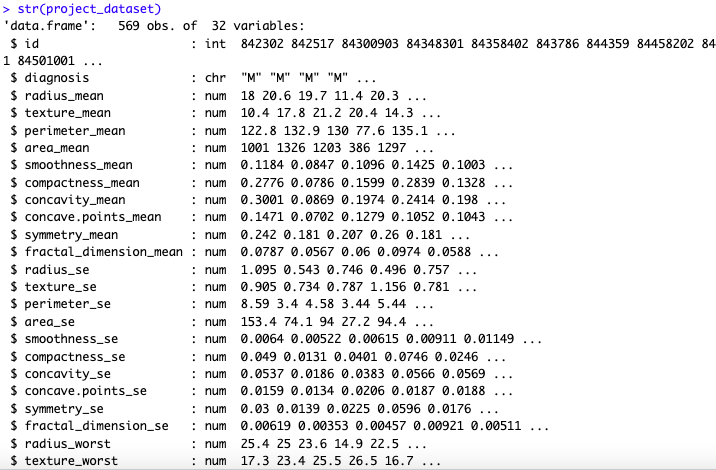
**Reading Dataset –**

The dataset for this project named ‘project\_dataset’ is read from the file ‘breast\_cancer.csv’ using the function below,



Figure : Reading the dataset

The dataset has 569 of instances and 32 attributes. Below, the details about every feature is shown,



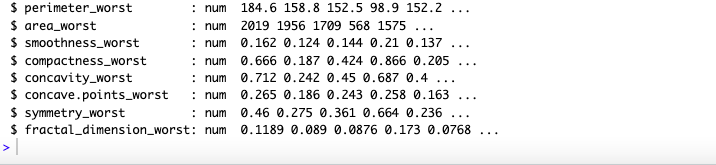
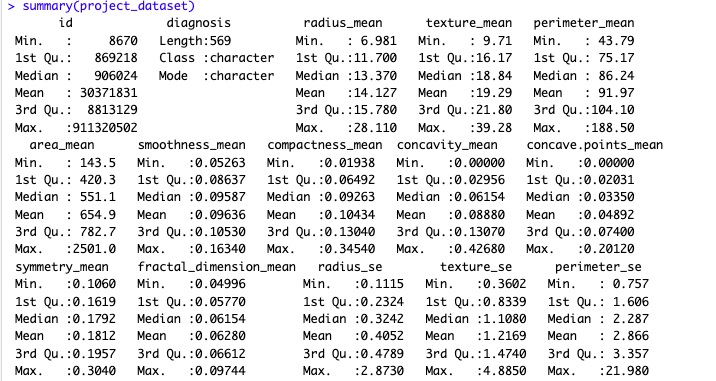


Figure : Details of the dataset

As, the ‘id’ attribute is used for indexing the instances. The summary of the ‘id’ attribute should not be considered. On the other hand, ‘diagnosis’ attribute is categorical. That’s why, only ‘mode value’ of this attribute is considerable. Rest of the features are summarized with their ‘min’, ‘max’, ‘1st quadrate’, ‘3rd quadrate’, ‘mean’, ‘median’ values. The summary of each feature is in below,



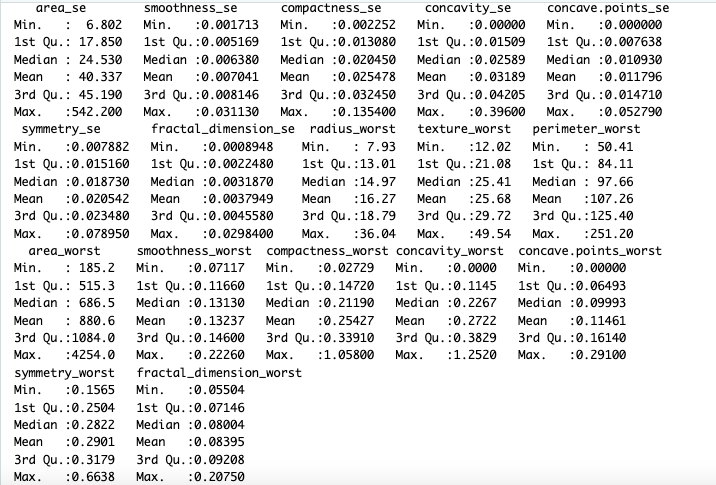


Figure : Summary of the dataset

**Data Transformation –**

As we know, for KNN classification algorithm every feature of the data should be in numerical type. Though, all the features have the numerical data except the ‘diagnosis’ attribute. This ‘diagnosis’ attribute has the value of ‘M’ or ‘B’. Now, this value should have the form of numerical to be operated. That’s why, ‘M’ is transformed into ‘1’ which means ‘cancerous’ and ‘B’ is transform into ‘0’ which means ‘non-cancerous’. In order to do this, below steps are followed,

Step-1: Observing the ‘diagnosis’ attribute first,

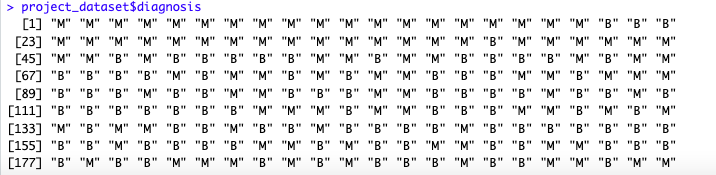


Figure : Raw value of 'diagnosis' attribute

Step-2: Transforming the data,



Figure : Code of processing 'diagnosis' attribute

Step-3: Again, observing for ensuring the transformation,

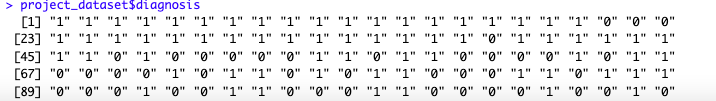


Figure : Transformed 'diagnosis' attribute

Though these data are in factor type but ready for the classification algorithm.

**Handling Missing Values –**

Now, let’s check the recently transformed attribute(‘diagnosis’) for missing values. For doing this, the bar plot has used. In this plot, if any value exists except ‘M’ & ‘B’, another bar will be sketched in it. The plot is in the below,

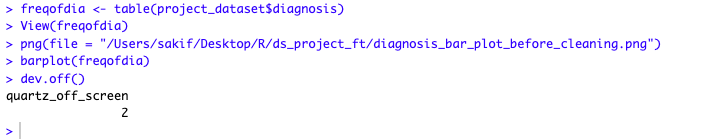


Figure : Code of bar plot

Through this bar plot, it is concluded that, there is no missing value in the ‘diagnosis’ attribute.

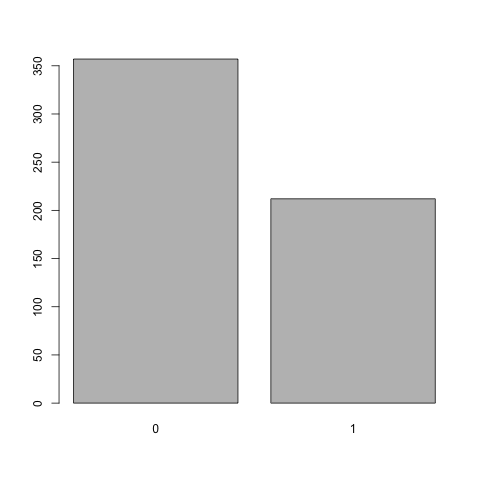


Figure : Applied bar plot in 'diagnosis' attribute

This frequency table emphasizes on, there is no values in the ‘diagnosis’ attribute except ‘0’ and ‘1’.



Figure : Frequency table for 'diagnosis' attribute

After this operation, the whole dataset has been checked for the missing values. Fortunately, there is no missing value at all.

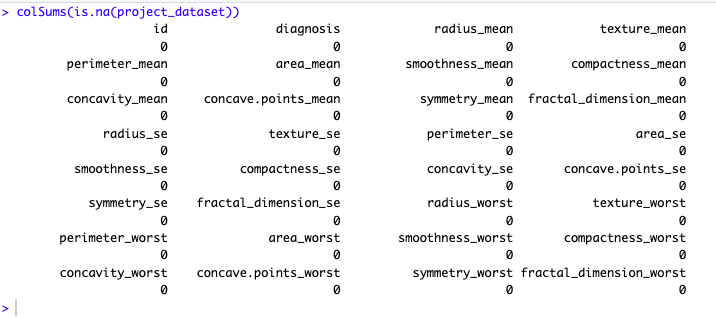


Figure : Code of checking missing values

For the extra level of satisfaction, the process of missing value elimination was performed in the whole dataset.



Figure : Code of omitting missing values

**Correlation & Feature Selection –**

Correlation is one of the best strategy to select the appropriate features related to the target attribute and remove the attributes whose values are considered as duplicated or unnecessary for the classification or other model building processes. Here, the Pearson correlation coefficient approach has been used for defining the correlation among the whole dataset.

As we know, correlation can only be established among the numerical data and from the before process, the ‘diagnosis’ attribute was transformed into ‘factor’ datatype. So, this feature is converted into ‘integer’ first.



Figure : Code of converting 'diagnosis' attribute factor to integer

Now the correlation defining operation can be performed through the ‘cor’ function and the calculation is in below,

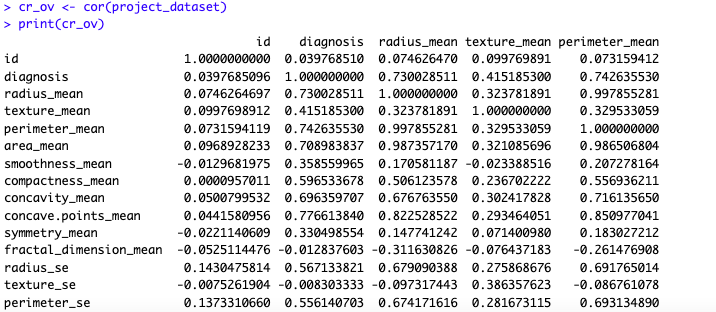


Figure : View of correlation coefficients among the attributes

As we can see the above calculation is a little bit messy. So, in convenience to understand the correlation, the ‘corrplot’ function has been used to have a correlation plot diagram. In the plot diagram, common points represent the correlation between two attributes and the color intensities represent the measurement of the correlation coefficient. Below is the diagram,



Figure : Plotting the correlation plot among the dataset

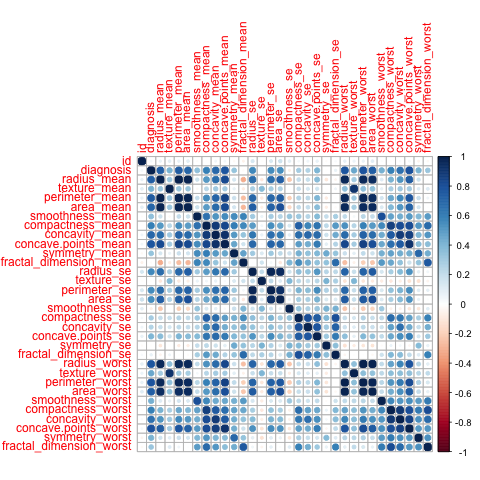


Figure : Correlation diagram of whole dataset

From the correlation diagram there are some observations such as,

Observation-1: the ‘id’ attribute is not correlated with any of the features as expected because it is only useful for indexing.

Observation-2: The features ‘fractal\_dimension\_mean’, ‘texture\_se’, ‘smoothness\_se’, ‘symmetry\_se’, ‘fractal\_dimension\_se’ are not correlated with target attribute ‘diagnosis’.

Observation-3: There are some independent attributes which are highly correlated with other independent attributes, such as

|  |  |
| --- | --- |
| **Independent attributes** | **Highly correlated independent attributes** |
| Radius\_mean | perimeter\_mean, area\_mean, radius\_worst, perimeter\_worst, area\_worst |
| Texture\_mean | texture\_worst |
| Compactness\_mean | concavity\_worst, concave.points\_worst |

Now, these observations will be used in omitting the unnecessary features and the rest of the features will be counted as the selected features. To select the feature, operations should be performed in steps regarding the observation,

Handling observation-1: The ‘id’ attribute is omitted.



Figure : Eliminating the 'id' attribute

Handling observation-2: The observed attributes which are not correlated should be omitted through below code,



Figure : Eliminating the attributes which have no correlation with target attribute

Let’s check the correlation plot after omitting,

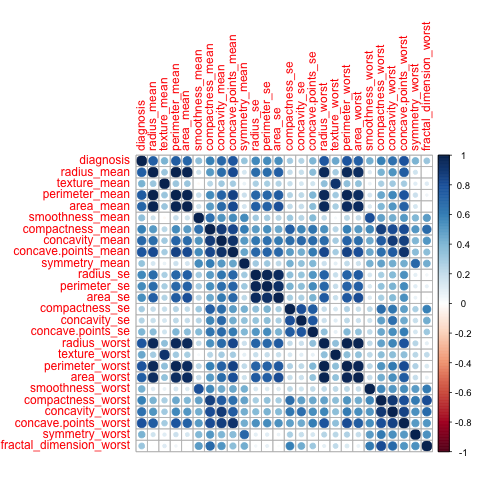
****

Figure : Correlation plot after handling 2nd observation

Handling observation-3: After handling the 2nd observation, the correlation plot says there are some independent features which are highly correlated with other independent features. Now, these highly correlated features should be omitted through below code,



Figure : Eliminating the highly correlated independent attributes

Now, let’s check the final correlation plot,

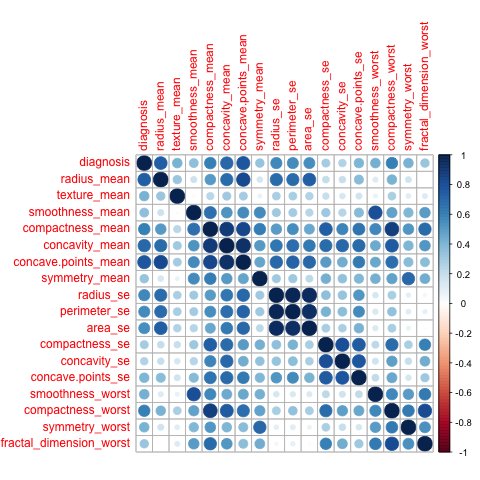


Figure : Correlation plot after final elimination

Though there are some highly correlated features which are present, the number of those features is few. Again, if these features get omitted that will make impact on the model accuracy.

After the elimination of features regarding the correlation, the selected features are in below,

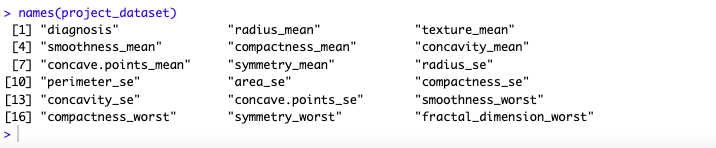


Figure : Final selected features for building the model

**Normalization –**

For the normalization, the min-max approach is applied to scale the values of those attribute in range of 0 to 1 here. As, the value of the ‘diagnosis’ attribute is categorical, normalization should not be performed to this attribute. That’s why the features which have to be normalized are differentiated in below,

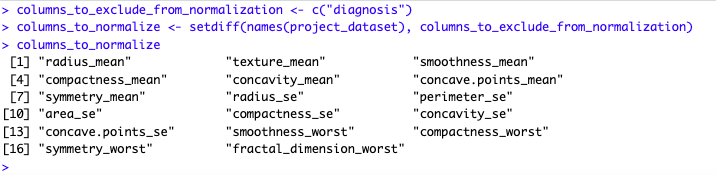


Figure : Select features to normalize

Now the min-max functionality is applied to the selected features for normalization and some data to show in below,

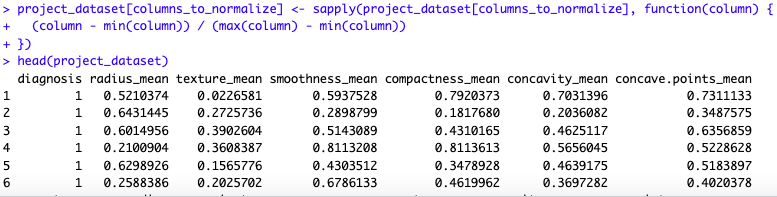


Figure : Applying the normalization and view of some normalized data

**Model Building –**

At this part of the project, the whole dataset is ready for developing the model. In order to do it, some steps must be followed,

Step-1: Splitting the data instances into train data and test data. Train data will get the 80% of the instances and the test data will get 20% of the instances. Using ‘createDataPartition’ function from the module ‘caret’ the data has been divided through this process,

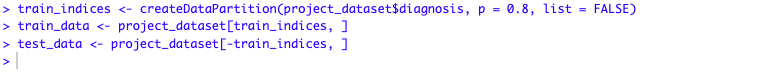


Figure : Code of data splitting into 8:2 ratio.

Step-2: Separating the target attribute and rest of the features instances like this,

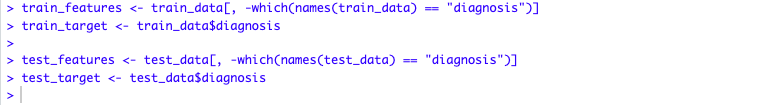


Figure : Code of separating data into train and target features

Step-3: Calculating the optimized value of K. The optimized value of K is the square root of the total instances.



Figure : Calculating the optimized value of K for the KNN classification

Step-4: Developing the model through ‘knn’ function of ‘class’ module. This function returns the predicted instances of ‘target’ attribute against the instances of ‘test\_features’ attributes.

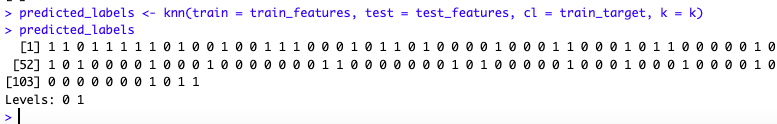


Figure : KNN classification is applied and the predicted labels are shown

Another approach of developing model will be followed in the Accuracy Measurement section.

**Accuracy Measurement –**

For calculating accuracy, two approaches are followed, first one is division and the second one is k-fold cross validation.

Approach-1: In accuracy of division, firstly the total number of correct predictions is calculated. Then the total number of correct predictions is divided by the total instances of ‘test\_target’ attribute. The final result is considered as the accuracy of the model.

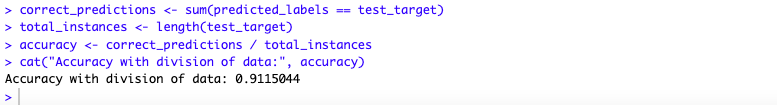


Figure : Code of calculating the accuracy in dividing method

Approach-2: In this k-fold cross validation, the k value is 10, that’s why this approach is named as 10-fold cross validation.

The ‘trainControl’ and ‘train’ functions from ‘class’ module are used for developing the model. The ‘trainControl’ function is setting the number of folds which is 10. The ‘train’ function is building the model. As the target attribute ‘diagnosis’ was converted to ‘integer’ for the correlation before, this ‘train\_target’ attribute is converted to ‘factor’ again. Because the ‘train’ function only accepts the ‘factor’ type ‘train\_target’. After that the property ‘$results$Accuracy’ is used to show the accuracy.

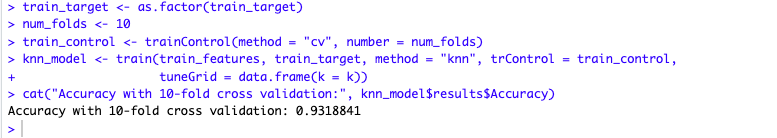


Figure : Code of calculating the accuracy in 10-fold cross validation method

**Confusion Matrix –**

For the confusion matrix is defined through the ‘confusionMatrix’ function from the ‘caret’ module. It uses two arguments ‘predicted\_labels’ from the model building approach-1 and the ‘test\_target’ which needs to be ‘factor’ datatype. As the target attribute ‘diagnosis’ was converted to ‘integer’ for the correlation before, this ‘test\_target’ attribute is converted to ‘factor’ again. Finally, the confusion matrix is generated like this,



Figure : Confusion matrix

Now the values of these properties ‘Pos Pred Value’ & ‘Sensitivity’ which came from the object of the ‘confusionMatrix’ function represent the ‘precision’ & ‘recall’ values repectfully as like as this,

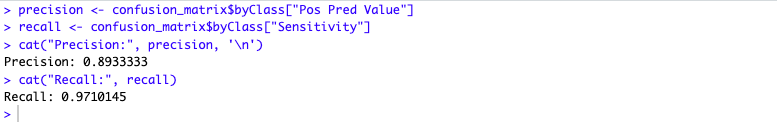


Figure : The values of precision & recall