**Breast Cancer Classification**

**Data Mining Project**

**Dr Walaa Gad**

Omar Gamal AbdelMageed AbdelGawad 20201701634 G2

Omar EmadEldeen Anwar Metwaly Gebal 20201701635 G2

Mariam Emad Roushdy Habashy Issak 20201701654 G3

Pola Sameh Magdi Gergis 20201701670 G2

Martina Ashraf Shawky Awad 20201701641 G1

**Table of Contents**

**Introduction 2**

Objective, Purposes and Methodology 2

Report Structure 2

**Data Cleaning and Preprocessing 3**

**Describing Our Dataset 5**

1. Radius 5
2. Texture 5
3. Perimeter 6
4. Area 6
5. Smoothness 7
6. Compactness 7
7. Concavity 8
8. Concave Points 8
9. Symmetry 9
10. Fractal Dimension 9
11. Diagnosis 10

**Balancing Data 11**

**Splitting Data Frame 11**

**Without Features Selection 12**

1. KNN Classification 12
2. Naïve Bayesian Classification 13
3. Decision Tree Classification 14
4. Support Vector Machine 16

**With Features Selection 18**

Feature Selection 18

1. KNN Classification 22
2. Naïve Bayesian Classification 23
3. Decision Tree Classification 24
4. Support Vector Machine 25

**Neural Network Classification 27**

**Conclusion 31**

**Introduction**

This report summarizes the statistical modeling and analysis results associated with a Breast Cancer Diagnosis dataset (1). This dataset is originally from Wisconsin. The Breast Cancer Diagnosis dataset features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

**Objective, Purposes and Methodology**

The main purpose of this report is to compare classification models in predicting whether the tumor in Benign or Malignant. The classification models used are KNN Classification, Naïve Bayesian Classification, Decision Tree Classification, Support Vector Machines and Neural Network Classification. Along the way the features of our dataset will be described.

**Report Structure**

The structure of the report is listed in the next few lines. Firstly, we will start with the description of the features, then we will dive into comparing the classification models without feature selection then with feature selection then the neural network as per the order above in the Objective, Purposes and Methodology.

**Body**

**Data Cleaning and Preprocessing**

We started off by cleaning our dataset.



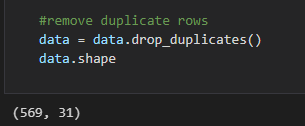
Firstly, we checked if there are any null values and found none.



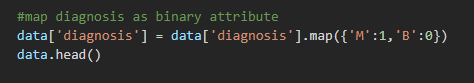
Then we dropped the id column which was unnecessary.



Then, we checked if there are any duplicates but we found none.



Lastly, we mapped the only categorical attribute – diagnosis – into zeros (Benign) and ones (Malignant).



Data cleaning and preprocessing resulted in no reduction of rows showing us that the data is already mostly clean. Total number of rows 569.

**Describing Our Dataset**

Now that our dataset has 569 rows, we will start describing our variables.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. n the 3-dimensional space is that 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].

* In all our histograms we used 4 classes.
* All of the continuous features have three variations in the dataset. The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image.
* We use the following libraries for describing and visualizing.



|  |
| --- |
| **1. Radius** |
| The first feature in our data and it describes mean of distances from center to points on the perimeter. |
| It is quantitative continuous data; we used a histogram to visualize it.C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\141A3F21.tmp |

|  |
| --- |
| **2. Texture** |
| The second feature in our data and it describes the gray-scale values. |
| C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7A54EED7.tmp |

|  |
| --- |
| **3. Perimeter** |
| The third feature in our data. |
| C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D80855FD.tmp |
| **4. Area** |
| The fourth feature in our data. |
| C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\26D91613.tmp |
| **5. Smoothness** |
| The fifth feature in our data and it describes local variation in radius lengths. |
| It is quantitative continuous data; we used a histogram to visualize it. |
| **C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\327FCC99.tmp** |

|  |
| --- |
| **6. Compactness** |
| The sixth feature in our data and it describes (perimeter^2 / area - 1.0). |
| It is quantitative continuous data; we used a histogram to visualize it. |
| **C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\C476730F.tmp** |
| **7. Concavity** |
| The seventh feature in our data and it describes severity of concave portions of the contour. |
| It is quantitative continuous data; we used a histogram to visualize it. |
| **C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\C7ECBEF5.tmp** |

|  |
| --- |
| **8. Concave Points** |
| The eighth feature in our data and it describes number of concave portions of the contour. |
| It is quantitative continuous data; we used a histogram to visualize it. |
| **C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D89B81CB.tmp** |

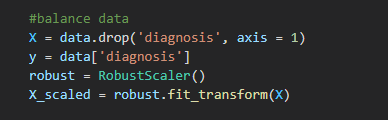
|  |
| --- |
| **9. Symmetry** |
| The ninth feature in our data. |
| It is quantitative continuous data; we used a histogram to visualize it. |
| C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\82CF0911.tmp |
| **10. Fractal Dimension** |
| The tenth feature in our data. |
| It is quantitative continuous data; we used a histogram to visualize it. |
| **C:\Users\Mariam\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\44C07E47.tmp** |

|  |
| --- |
| **11. Diagnosis** |
| The last and most important feature in our data is diagnosis. It describes the type of tumor. (M = malignant, B = benign) |
| It is quantitative continuous data; we used a bar chart and pie chart to visualize it. |
|  |
| As we can see data is imbalanced. |

**Balancing Data**

Since data according to diagnosis is unbalanced we will try to balance it by Robust Scalar.



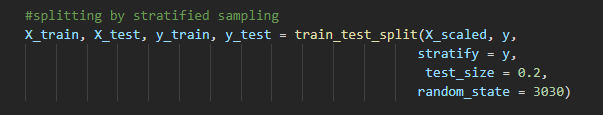


**Splitting Data Frame**

**To be able to test our methods we split our data to training data and testing data**

We split 80% of the data for the training, and 20% is for testing the accuracy





Since our goal is to predict whether the breast tumor is malignant (cancerous) or malignant (not cancerous) we will use 5 methods to classify our inputs. Then we will compare the accuracy of each to determine the best way to accurately classify the tumors

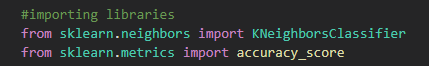
1. KNN Classification
2. Bayesian Classification
3. Decision Tree Classification
4. Support Vector Machines
5. Classification using a neural network

**Now we will apply the classification models without feature selection then one more time with feature selection to check which has the best accuracy score.**

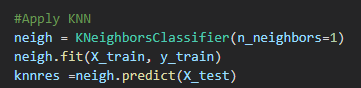
**Without Feature Selection**

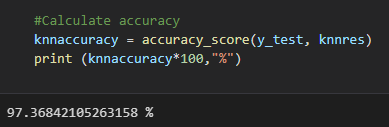
**Classification Method 1 – KNN (K Nearest Neighbor) Model**

We begin by importing the KNeighborsClassifier and accuracy\_score from sklearn:



We then apply the KNN

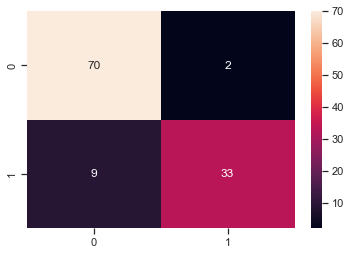


After using the test sample to test the algorithm we got an accuracy of **97.37%**

Confusion matrix plot

Text

Description automatically generated



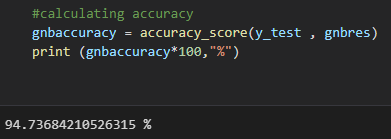
**Classification Method 2 – Naïve Bayesian Classification**

We begin by importing the GaussianNB and accuracy\_score from sklearn:



We then apply the Naïve Bayesian Classification



After using the test sample to test the algorithm we got an accuracy of **94.74%**

Confusion matrix plot

**Text

Description automatically generated**

Chart, square

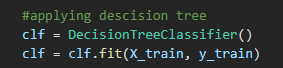
Description automatically generated with medium confidence

**Classification Method 3 – Decision Tree Classification**

We begin by importing the DecisionTreeClassifer and accuracy\_score from sklearn:

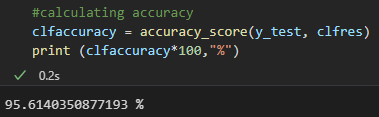


Then we apply the Decision Tree Classifier

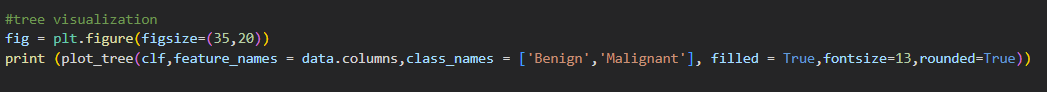


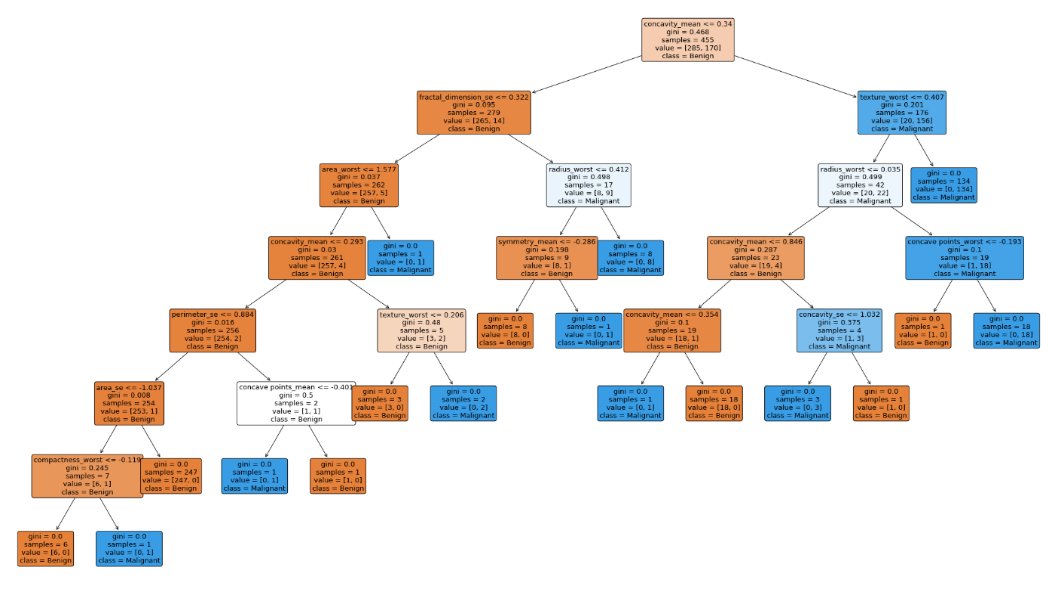
Then we predict the test sample



After using the test sample to test the algorithm we got an accuracy of **95.61%**

**Decision tree plot**

****

****

**Confusion matrix plot**

**Text

Description automatically generated**

A picture containing chart

Description automatically generated

**Classification Method 4 – Support Vector Machines**

We begin by importing svm from sklearn:

A screenshot of a computer

Description automatically generated with medium confidence

Then we apply the model

Text

Description automatically generated

Then we test our model

Text

Description automatically generated

Then we calculate the accuracy and get accuracy of **95.614%**

Text

Description automatically generated

Plotting the confusion matrix  
Text

Description automatically generated

A picture containing chart

Description automatically generated

**Now that we have tried all the classification models without feature selection, we conclude that KNN is the best method out of all:**

|  |  |
| --- | --- |
|  | Accuracy |
| KNN | 97.37% |
| Naïve Bayesian | 94.74% |
| Decision Tree | 95.61% |
| SVM | 95.614% |

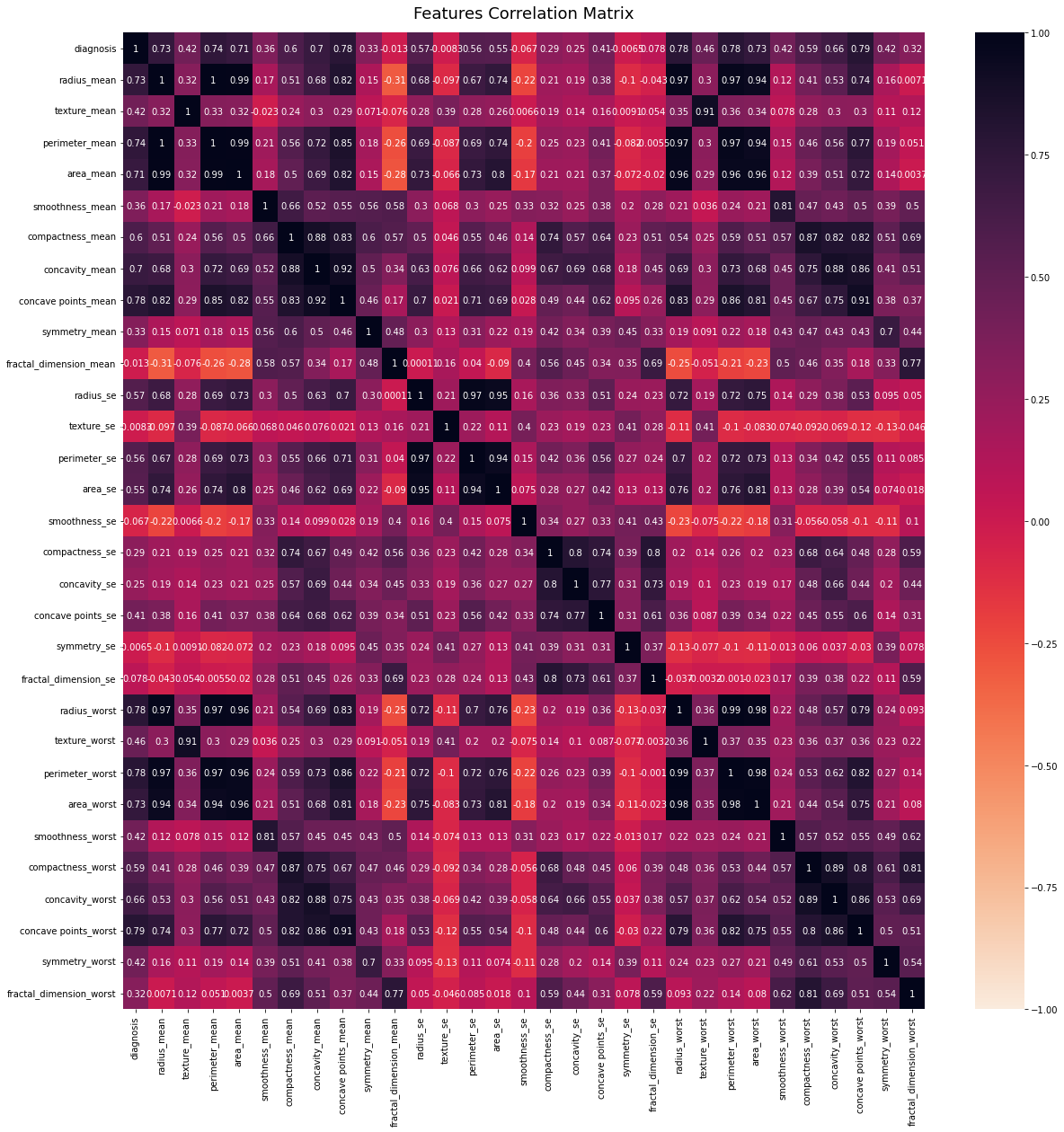
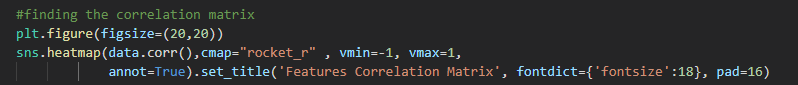
Chart, bar chart

Description automatically generated

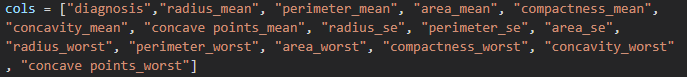
**With Feature Selection**

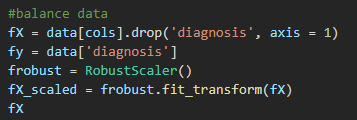
**Feature Selection**

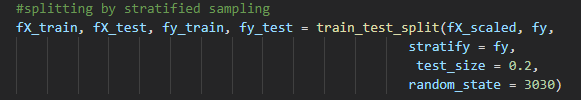
To find the most prominent features we compute the correlation matrix.

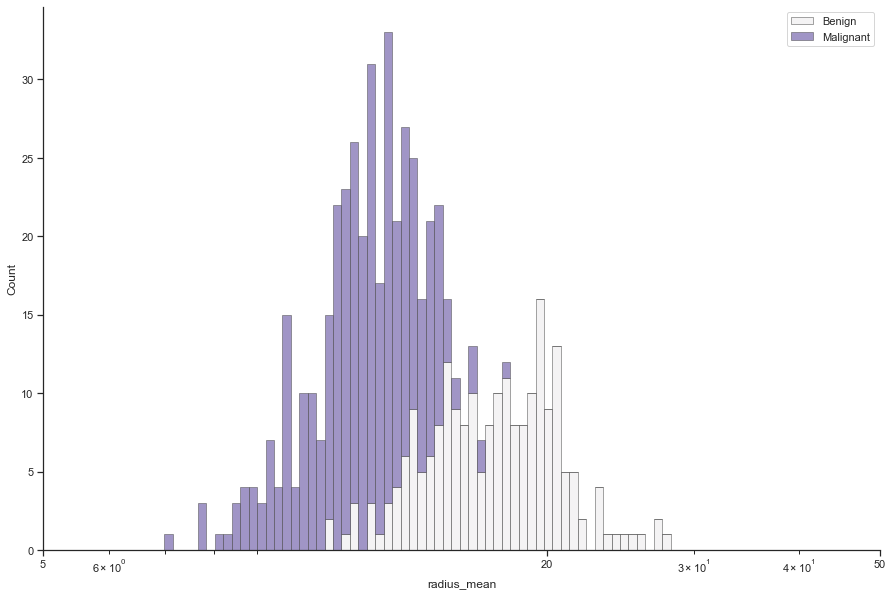


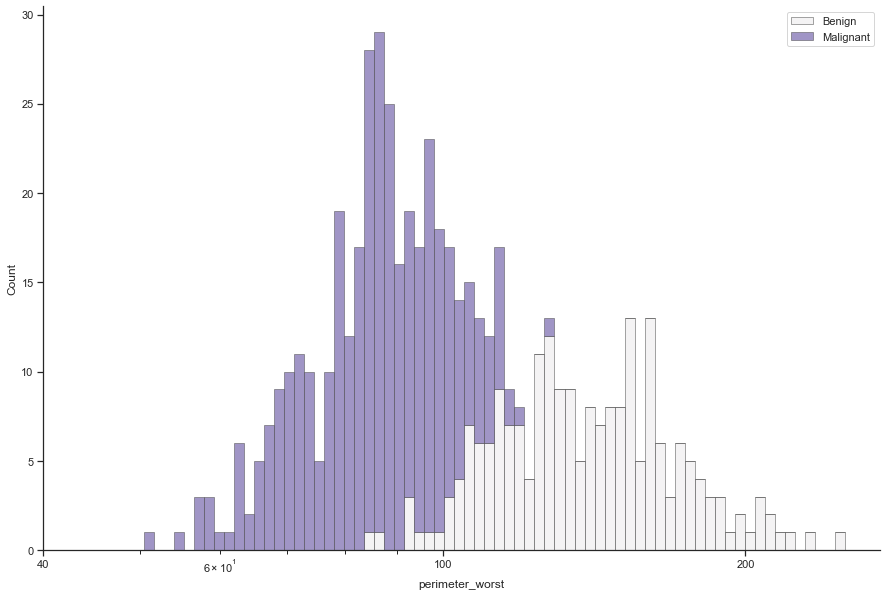
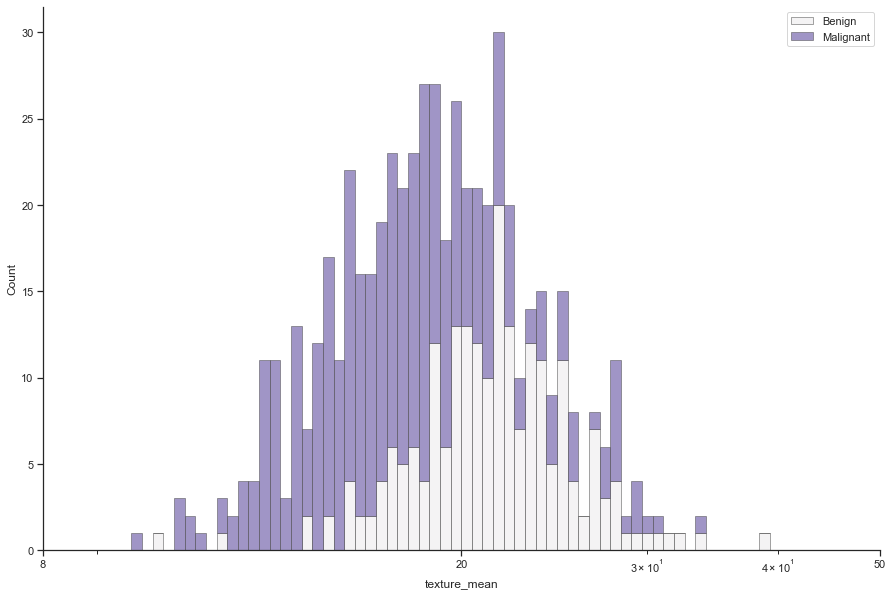
Now to view the correlation directly with diagnosis in sorted order we run the following code: 

We see that radius\_mean, perimeter\_mean, area\_mean, compactness\_mean, concavity\_mean, concave points\_mean, radius\_se,perimeter\_se, area\_se, radius\_worst, perimeter\_worst, area\_worst, compactness\_worst, concavity\_worst and concave points\_worst are highly correlated with diagnosis.

Then we balance and split the data again after choosing the most correlated features to the diagnosis.

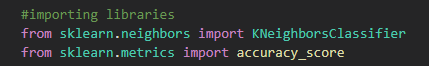


These plots show the ratio of malignant to benign tumors in different ranges of tumor radius, texture and perimeter respectively. 

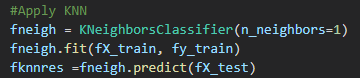


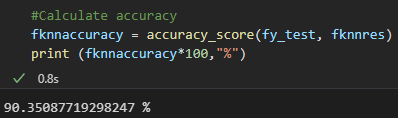
**Classification Method 1 – KNN (K Nearest Neighbor) Model**

We begin by importing the KNeighborsClassifier and accuracy\_score from sklearn:



We then apply the KNN



After using the test sample to test the algorithm we got an accuracy of **90.35%**

Now we plot the confusion matrix

Text

Description automatically generated

A picture containing chart

Description automatically generated

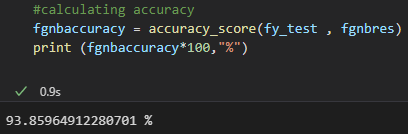
**Classification Method 2 – Naïve Bayesian Classification**

We begin by importing the GaussianNB and accuracy\_score from sklearn:



We then apply the Naïve Bayesian Classification



After using the test sample to test the algorithm we got an accuracy of **93.86%**

Now we plot the confusion matrix

Text

Description automatically generated

Square

Description automatically generated with low confidence

**Classification Method 3 – Decision Tree Classification**

We begin by importing the DecisionTreeClassifer and accuracy\_score from sklearn:

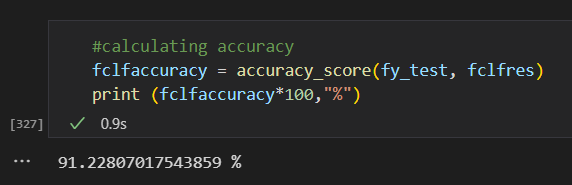


Then we apply the Decision Tree Classifier



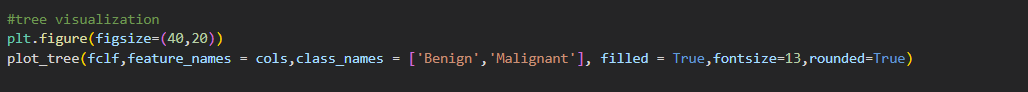
Then we predict the test sample



After using the test sample to test the algorithm we got an accuracy of **91.22%**

The decision tree plot

Timeline, treemap chart

Description automatically generated

Now we will plot confusion matrix

**Text

Description automatically generated**

A picture containing chart

Description automatically generated

**Classification Method 4 – Support Vector Machines**

We begin by importing svm from sklearn:

A screenshot of a computer

Description automatically generated with medium confidence

Then we apply the model

Text

Description automatically generated

Then we test our model

Text

Description automatically generated

Then we calculate the accuracy and get accuracy of **92.982%**

Text

Description automatically generated

Plotting the confusion matrix  
Text

Description automatically generated

Chart

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | Accuracy |
| KNN | 90.35% |
| Naïve Bayesian | 93.86% |
| Decision Tree | 91.22% |
| SVM | 92.982% |

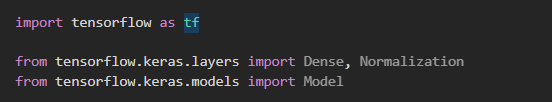
**Now that we have tried all the classification models with feature selection, we conclude that the Naïve Bayesian is the best, however the models are generally more accurate when we don’t use feature selection:**

Chart, bar chart

Description automatically generated

**Classification Using Neural Networks**

First, we import the needed libraries.



It is good practice to normalize features that have different ranges.

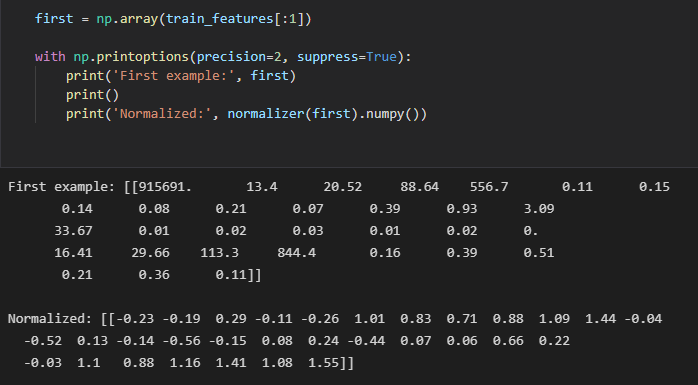
If we skip normalizing the features, the model will still converge (ie: reach the optimum loss. i.e. become able to predict and perform regression as best as it can), However, normalization makes training much more stable and faster in performance.

The tf.keras.layers.Normalization is a clean and simple way to add feature normalization into the model.

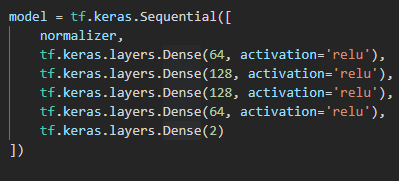
The first step is to create the layer:

Then, fit the state of the preprocessing layer to the data by calling Normalization.adapt:

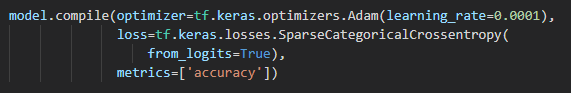
When the layer is called, it returns the input data, with each feature independently normalized:



**Building the Neural Network Model**

We first declare the layers of the neural network and their activation functions.

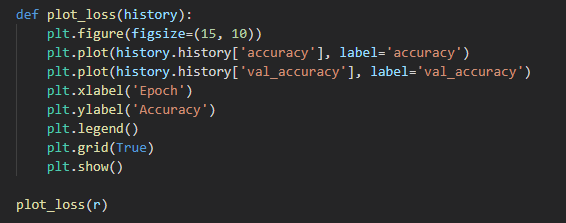
We then config the model with losses and metrics that we want using model.compile()

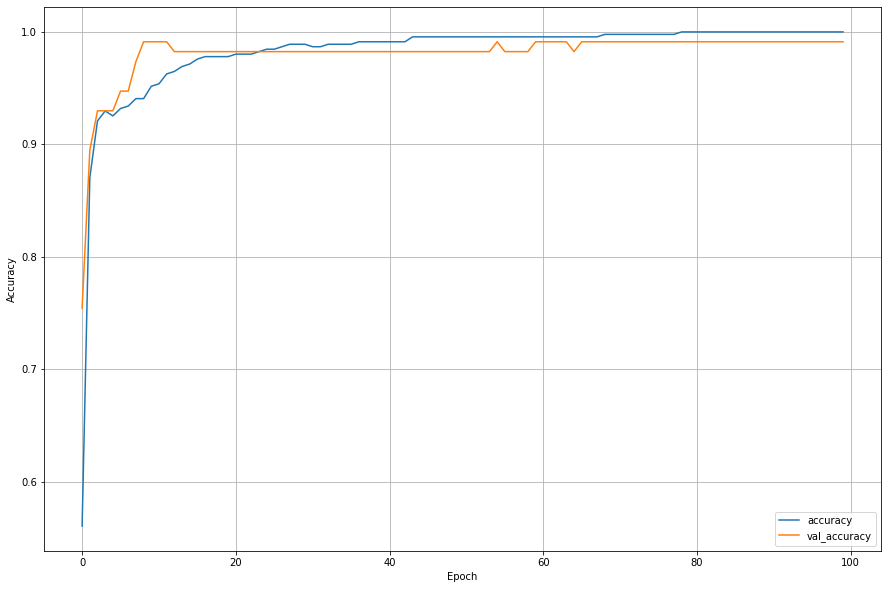


We then train the model on the training data for a fixed number of epochs (iterations).



**Model Evaluation**

model.fit returns a history object which contains the loss and accuracy of the model at each epoch. So now we can use this object to plot accuracy across time so as to get a better understanding of the model's performance.



**Make Predictions**

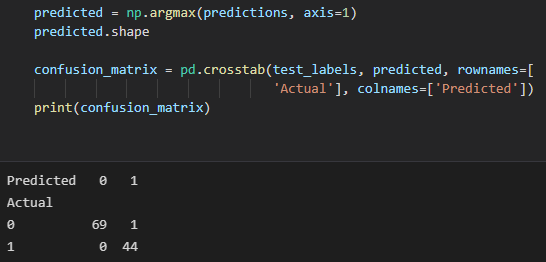
With the model trained, we now use it to make predictions about the type of the tumor. We attach a softmax layer to convert the model's output to probabilities, which are easier to interpret.

We let the model predict the clarity of the test set:

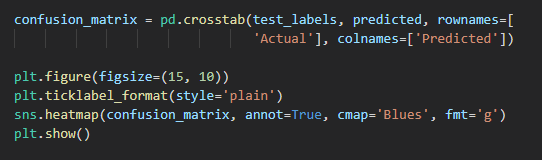


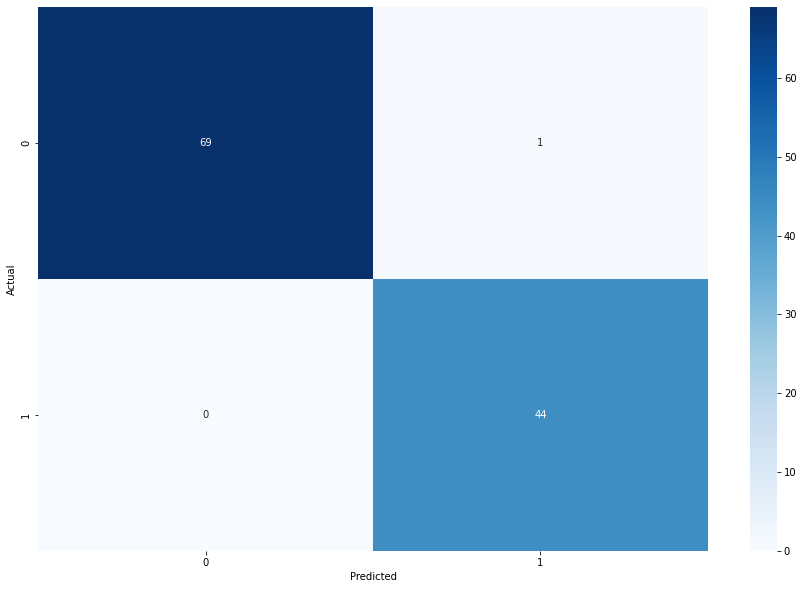
**Review Performance**

To get an overview of the model's performance, we print the confusion matrix of the predicted labels. This shows us how many labels did the model classify correctly and how many did it miss.



Now let's visualize the confusion matrix:





We built a classifying deep neural network model to classify the tumor type using TensorFlow library. The model reached top accuracy of **99.12%**. In the end we visualized the model's performance using the confusion matrix.



**Conclusion**

Let’s recap, first we gave a brief description on every feature of our dataset. Then we calculated accuracy for each model from (KNN, Naïve Bayesian, Decision tree) with and without feature selection. The results show that without feature selection is better and that’s due to having more information to train on. That shows us that even if the correlation is small, it still helps in training our model in some cases. Lastly we tried using a neural network to predict whether the tumor is cancerous (malignant) or not (benign) and it gave us the best accuracy out of all the classification models **(99.12% accuracy**).