Ranking of Various ML Classifier Algorithms Based on Their Accuracy, Score, and Test Score on Lung Cancer Prediction

Deepesh Jami, Srinivas Likhith Raichur and Lokesh Varma
School of Computer Science and Engineering(SCOPE)
Department of Computer Science - VIT

ABSTRACT

Lung cancer is the prime cause of cancer related deaths for both men and women. Lung cancer is the exponential growth of malicious cells in one or both Lung.Lung cancer is a type of cancer that starts in the lungs. Your lungs are two sponge-shaped organs in your chest that absorb oxygen when you inhale and exhale as you exhale. Lung cancer is the leading cause of cancer deaths in the United States, both men and women. Lung cancer kills more lives each year than colon, prostate, ovarian and breast cancer combined.

The information related to patients is collected from the standard dataset then it is preprocessed. The noisy, irrelevant, missing data is eliminated. Then we are going to use classification algorithms like K Nearest Neighbour, Naïve Bayes, and Support Vector Machine algorithms to build a cancer risk prediction system is proposed here which predicts cancers and is also user friendly, time and cost saving.

Introduction

Lung cancer is the most common cause of cancer worldwide. If the lung cancer has spread, a person may feel symptoms in other parts of the body. The lung cancer symptom is used to predict risk level of disease. We are analysing very popular algorithms and stimulating them by training with an approved dataset which is the dataset of people who are diagnosed. When the same data set is trained with different algorithms we can identify which one of the algorithms are giving out better results.

Problem Statement

There are various type of the cancer like lung, breast, prostrate, carnival etc. Each type of cancer has specific symptoms. Based on the symptoms the type of the cancer is predicted. In this project we are mainly focusing on the lung cancer prediction as 1 in 4 cancer deaths are from lung cancer.

Objective

We analyse the lung cancer prediction using classification algorithm such as Naive Bayes, Decision tree and SVM (Support Vector Machine). Cancer and non-cancer patient’s data is gathered from the dataset, pre-processing will be done and will be analysed using a classification algorithm for predicting lung cancer. The dataset that we have collected has 1000 instances and 25 attributes.

Literature Survey

A Review of System that predicts lung cancer using Data Mining Techniques and Self Organizing Map (SOM):

In this paper the author has developed a prototype lung cancer disease prediction system using data mining classification techniques. Where this system extracts the data from a historical lung cancer disease database. As they have mentioned in the paper, the most effective model to predict patients with Lung cancer disease appears to be SOM algorithm. As if they mentioned the self-organizing map (SOM) is a very good tool in exploratory phase of data mining. It projects input space on low dimensional regular grid prototypes that we can utilize to visualize and explore properties of the data. If number of SOM units are found high, to facilitate quantitative analysis of the data and the map, then we have to group the similar. In this paper, they analyze different approaches to clustering of the SOM are considered. When compared to the direct clustering of the data it is found that two stage procedure is performing well that is first produce the prototypes using SOM and the clustered in the second stage.

RECENT LUNG CANCER DETECTION TECHNIQUES:

In this paper the author has given details about the approaches of Artificial Neural Networks. An artificial neural network (Ann) is a graphical decision-making structure with processing nodes that simulates the biological neural structure in the brain. Determining the number of hidden nodes and layers and the activation function is crucial to the ANN's design. Feed forward network is a form where they are staggered like a feed- forward topology where the node layers are staggered like a feed forward topology. Multi-layer perceptron is another popular form of neural network models. To generate the output, the weighted sum from the inputs along with the bias term are forwarded to the antinational level. Radial basis function is another form of supervised based neural network. This supports single stage training in comparison to the repetitive training used by mlp. Principal component analysis is a statistical technique which generates linearly non correlated variable set known as the principal components. The results of a PCA are expressed as set of transformed variable values in relation to a specific data value.

Prediction of Lung Cancer Using Machine Learning Classifier (19 July 2020):

In this paper they have shown that with RBF classifier the accuracy is found to be 81.25% on lung cancer data. As they have mentioned the accuracy can be further improved with suitable feature selection and integrated approach with other supervised learning process and modified functional

Detection and Prediction of Lung Cancer:

By using a method of image classification, we take the images and then input into the MATLAB software and from there we do preprocessing to the dataset. And from MATLAB we use the preprocessing toolbox that is available this approach will do the classification fast but this is not a very viable approach as this might not be very good when we want to customize the results. This approach might be very viable when the computation power is low on the system that we are using to build the model. OBJECTIVES: · Inbuilt methods are examined and the results are analyzed

Description of proposed methodologies

Data collection

There are various types of the cancer like lung,breast, prostrate, carnival etc. Each type of cancer has specific symptoms. Based on the symptoms the type of the cancer is predicted.
Dataset used should be more precise and accurate in order to improve the predictive accuracy of machine learning algorithms. We have taken our datasets from Kaggle (Cancer Patients Dataset) This dataset contains the attributes that are taken into consideration for lung cancer prediction.

Attributes:

Symptoms are considered as the attributes and diagnosis is done using Machine Learning techniques. Here we consider 25 attributes with 1000 instances or 1000 patients.
The attributes taken into consideration are:

- Age
- Dust Allergy
- Genetic Risk
- Fatigue
- wheezing
- Obesity
- chest pain
- Dry Cough
- Gender
- Alcohol use
- Chronic Lung Disease
- weight loss
- swallowing difficulty
- Smoking
- clubbing of finger nails
- Snoring.
- Air Pollution
- Occupational Hazards
- coughing of blood
- shortness of breath
- Balanced Diet
- passive smoker
- Frequent Cold

Data Analysis and Processing:

We have used various modules and libraries in order to make this process easy:
- scipy
- numpy
- matplotlib
- pandas
- sklearn
- seaborn

And then we performed Data Visualization in order to classify patients with respect to the attributes we have considered for lung cancer prediction.

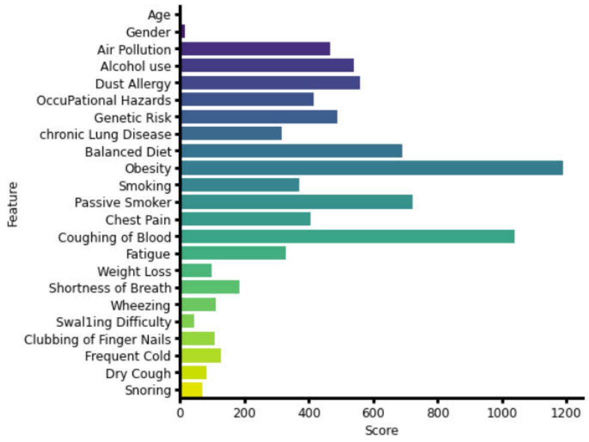
Feature Selection:

Though there are a lot of variables to look at we can just find the most important ones by using the SelectKBest Algorithm with ANOVA F-ratio statistics

Feature Selection is a method through which we can generate the F-ratio scores of all features and we can determine which ones to use for machine learning.

Hence using feature selection, we have confined our dataset to certain attributes which obtained from the results of this process

Feature selection Graph:



Model Selection:

Since this is a Classification problem various Supervised Machine learning algorithm can be used. For this project the algorithms which we chose were Linear Regression, Decision Tree, Random Forest, K Nearest Neighbors, and Artificial Neural Network.

• *K-NN Algorithm*: it takes similarities between new case / data and available cases and puts the new case in the category that closely resembles the available categories. The K-NN algorithm stores all available data and separates the new data point accordingly. This means that where new data comes from it can be easily categorized into a compatible category using the K-NN algorithm. The K-NN algorithm can be used for deceleration and partitioning but mainly for partition problems.

• *SVM:* In machine learning, vector support systems (SVMs, and vector support networks) supervised learning models have compatible learning algorithms that analyze the data used to classify and re-analyze. Given a set of training examples, each marked as one or two of the two categories, the SVM training algorithm creates a model that provides new examples in one category or another, making limited class distinctions for binary (although methods such as
Platt rating is available to use SVM in possible configuration settings)

• *Naïve Bayes:* In machine learning, the Naïve Bayes dividers are a family of simple "actionable" objects based on using the Bayes' theorem with strong (naïve) independent ideas between the elements. They are among the simplest forms of the Basesian network. It is not a single algorithm but a family of algorithms in which all share the same goal, which means that all components are independent of each other.

Evaluation of Models

In the project every model is implemented using Python's Scikit-Learn library. All the models are evaluated in the following procedure:

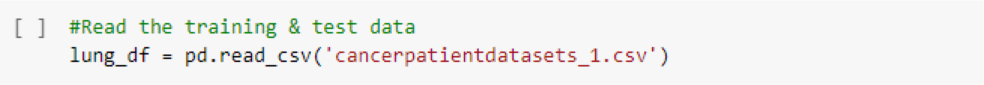
• *Importing all the libraries necessary for models*
• *Importing the Dataset*
• *Data preprocessing:*
This step involves the following two steps:
i) Dividing the data into attributes and labels
ii) Dividing the data into training and testing sets.
We have train_test_split methodin the Scikit-Learn library which allows us to comfortably divide data into training and test sets.
• *Training the Algorithm:*
Once we have divided the data into training and testing sets, its time to train our models on the training sets. The Scikit- Learn has a certain libraries like svm, sklearn.neighbors. These libraries have certain built-in classes to make our tasks easy.

• *Making predictions:*
We now finally come to the end of the process where we make prediction. This becomes possible using predict method from the sklearn libraries.
• *Evaluating the Algorithm:*
For evaluating the algorithms, we obtain a confusion matrix. The Scikit-Learn's metrics has certain methods like classification_report and confusion_matrix using which we can obtain the values of these metrics.

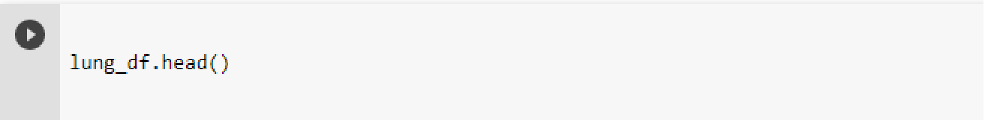
Implementation

Here we will be showing the data visualisation and feature selection steps. The final dataset used in the project for training and testing was obtained through this process

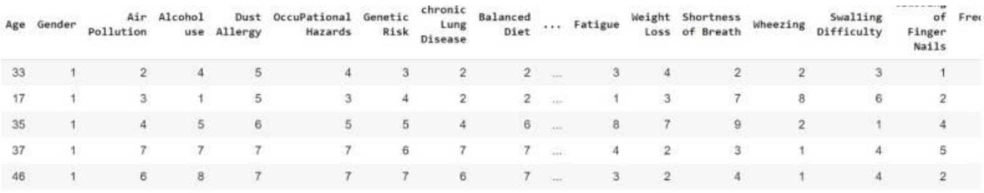
Dataset (Pandas head() method used to return first 5 rows of dataset)



```
[ ]  #Read the training & test data
     lung_df = pd.read_csv('cancerpatientdatasets_1.csv')
```



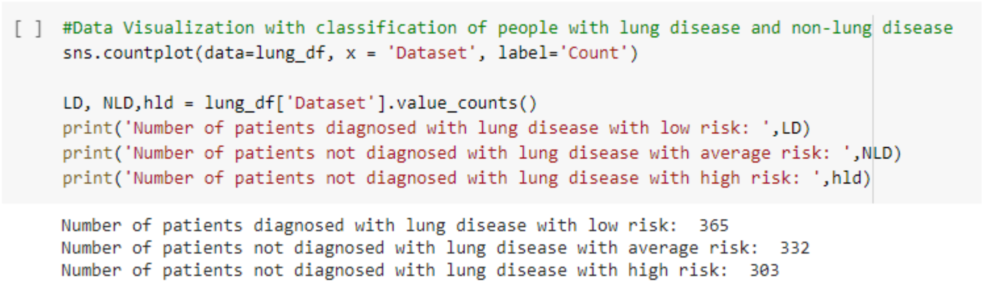
```
lung_df.head()
```



Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	...	Fatigue	Weight Loss	Shortness of Breath	Wheezing	Swalling Difficulty	of Finger Nails	Fre
33	1	2	4	5	4	3	2	2	...	3	4	2	2	3	1	
17	1	3	1	5	3	4	2	2	...	1	3	7	8	6	2	
35	1	4	5	6	5	5	4	6	...	8	7	9	2	1	4	
37	1	7	7	7	7	6	7	7	...	4	2	3	1	4	5	
46	1	6	8	7	7	7	6	7	...	3	2	4	1	4	2	

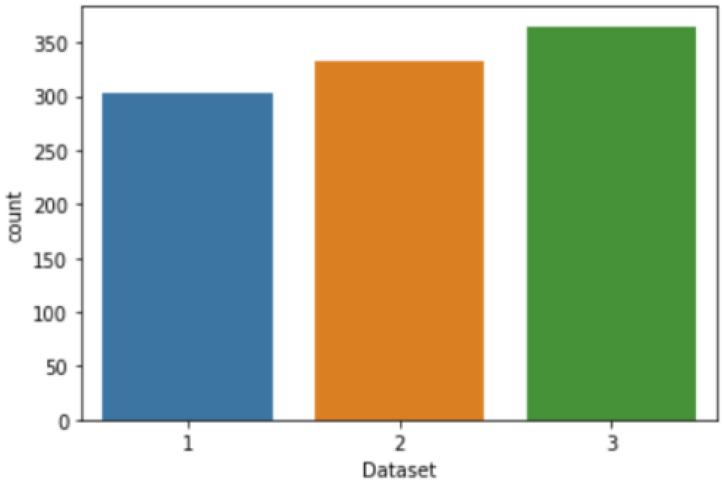
Data Visualization:

Visualization with classification of people with lung disease and non-lung disease



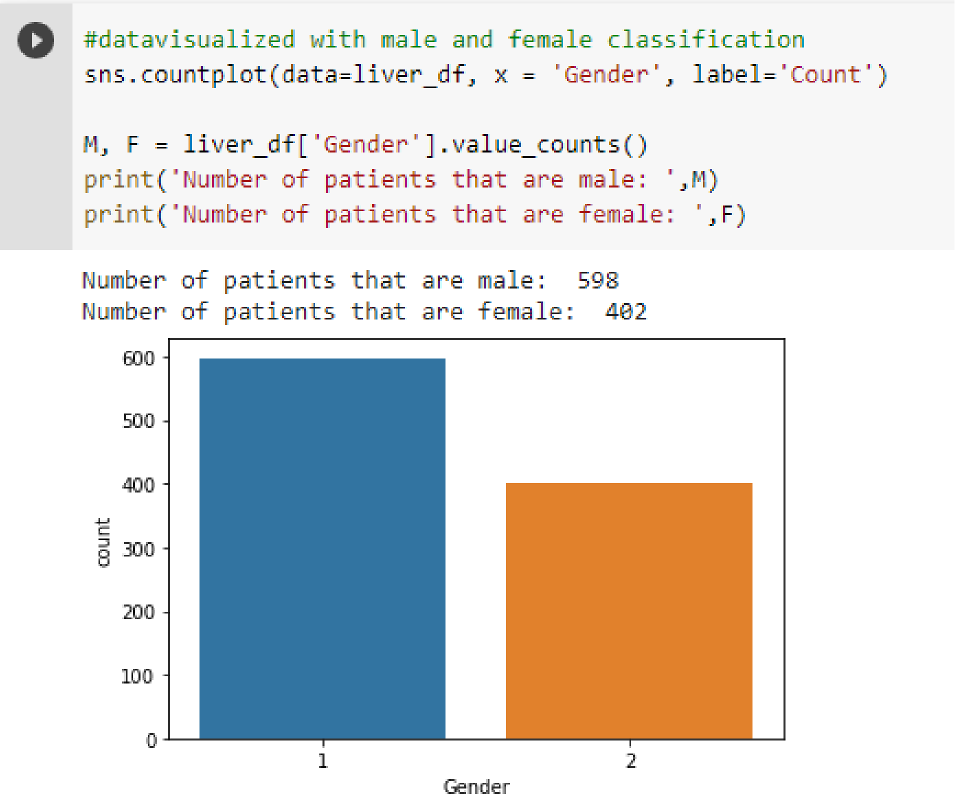
```
[ ]  #Data Visualization with classification of people with lung disease and non-lung disease
     sns.countplot(data=lung_df, x = 'Dataset', label='Count')

     LD, NLD,hld = lung_df['Dataset'].value_counts()
     print('Number of patients diagnosed with lung disease with low risk: ',LD)
     print('Number of patients not diagnosed with lung disease with average risk: ',NLD)
     print('Number of patients not diagnosed with lung disease with high risk: ',hld)
```

```
Number of patients diagnosed with lung disease with low risk:  365
Number of patients not diagnosed with lung disease with average risk:  332
Number of patients not diagnosed with lung disease with high risk:  303
```



Lung Disease vs Non-Lung Disease

Data Visualization with male and female classification

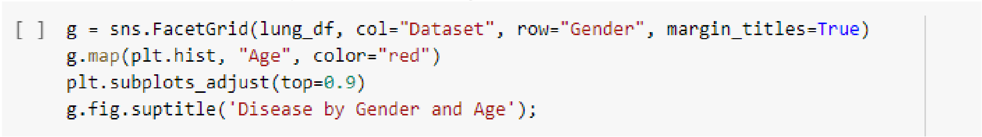


```
#datavisualized with male and female classification
sns.countplot(data=liver_df, x = 'Gender', label='Count')

M, F = liver_df['Gender'].value_counts()
print('Number of patients that are male: ',M)
print('Number of patients that are female: ',F)
```

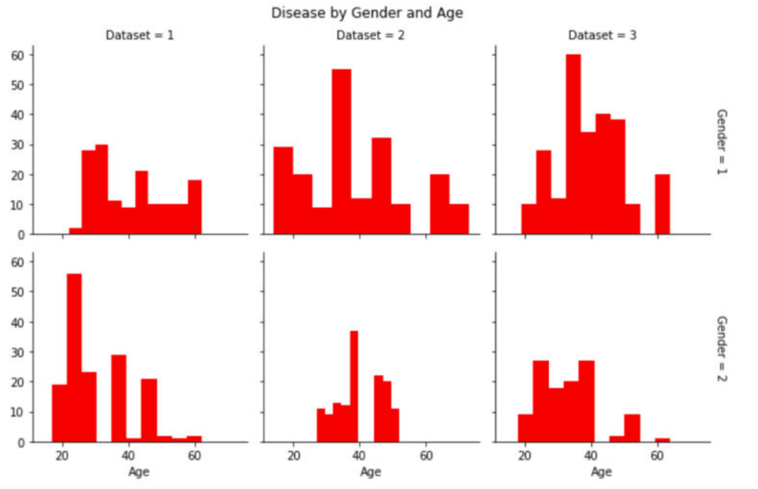
```
Number of patients that are male:  598
Number of patients that are female:  402
```

Male vs Female

Classification of disease by Gender and age:

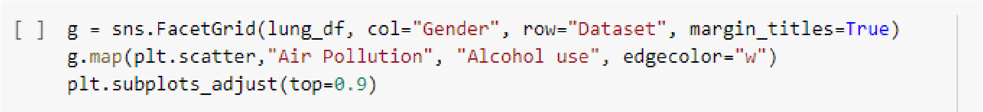


```
[ ]  g = sns.FacetGrid(lung_df, col="Dataset", row="Gender", margin_titles=True)
     g.map(plt.hist, "Age", color="red")
     plt.subplots_adjust(top=0.9)
     g.fig.suptitle('Disease by Gender and Age');
```

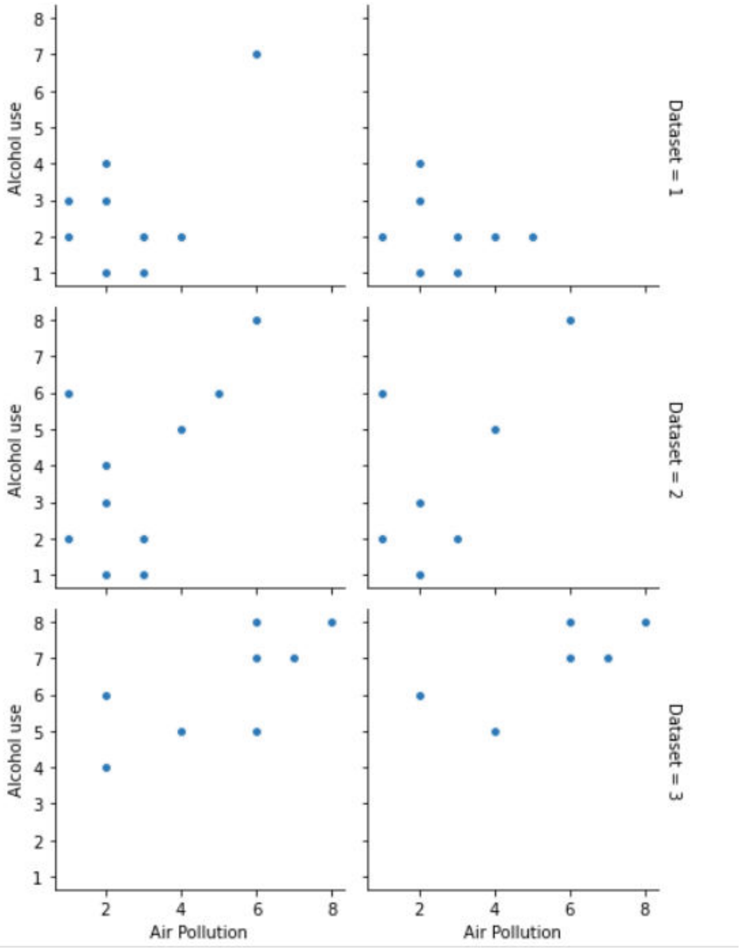


Gender vs Age

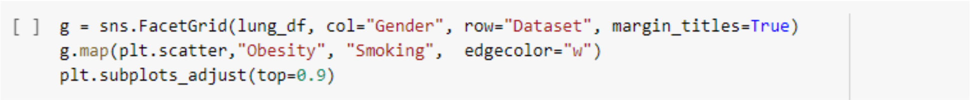
Classification of disease using Alcohol use and Air Pollution:



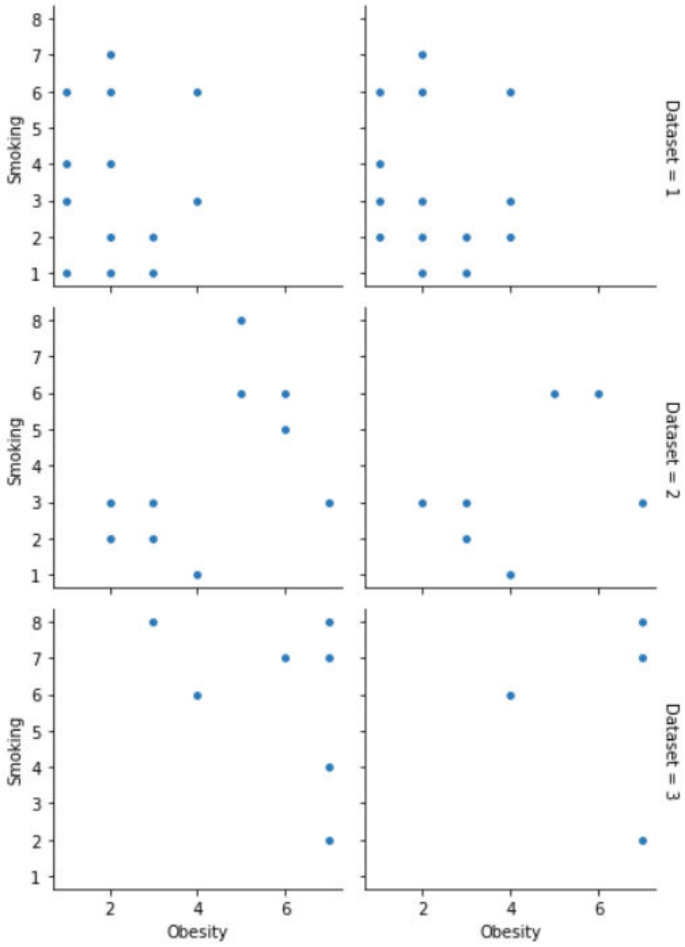
```
[ ]  g = sns.FacetGrid(lung_df, col="Gender", row="Dataset", margin_titles=True)
     g.map(plt.scatter,"Air Pollution", "Alcohol use", edgecolor="w")
     plt.subplots_adjust(top=0.9)
```



Classification of disease using Smoking and Obesity:



```
[ ]  g = sns.FacetGrid(lung_df, col="Gender", row="Dataset", margin_titles=True)
     g.map(plt.scatter,"Obesity", "Smoking",  edgecolor="w")
     plt.subplots_adjust(top=0.9)
```



Major Observation (Feature Selection)

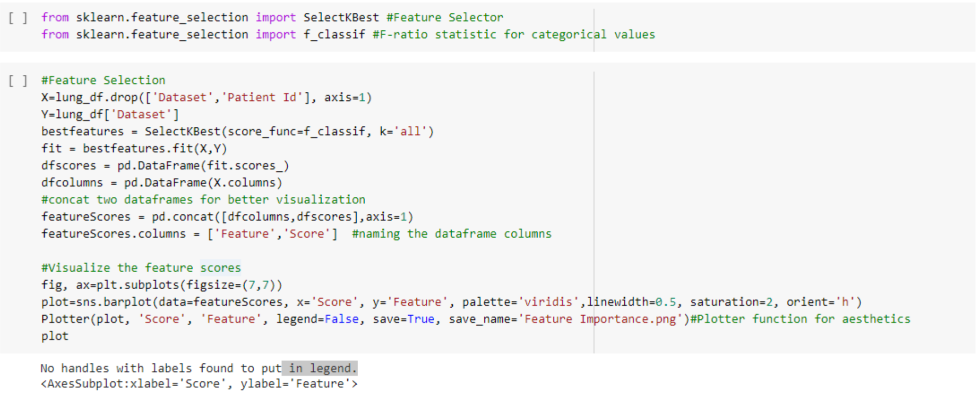
Though there are a lot of variables to look at we can just find the most important ones by using the SelectKBest Algorithm with ANOVA Fratio statistics.

Feature Selection is a method through which we can generate the Fratio scores of all features and we can determine which ones to use for machine learning.

Feature Selection Steps:

- Visualize the feature scores
- We will take all the features that scored more than 200 as they show the least redundancy
- Generate the features into a list
- Add the Level String to be used to make the new dataframe
- Generate new dataframe with the selected features

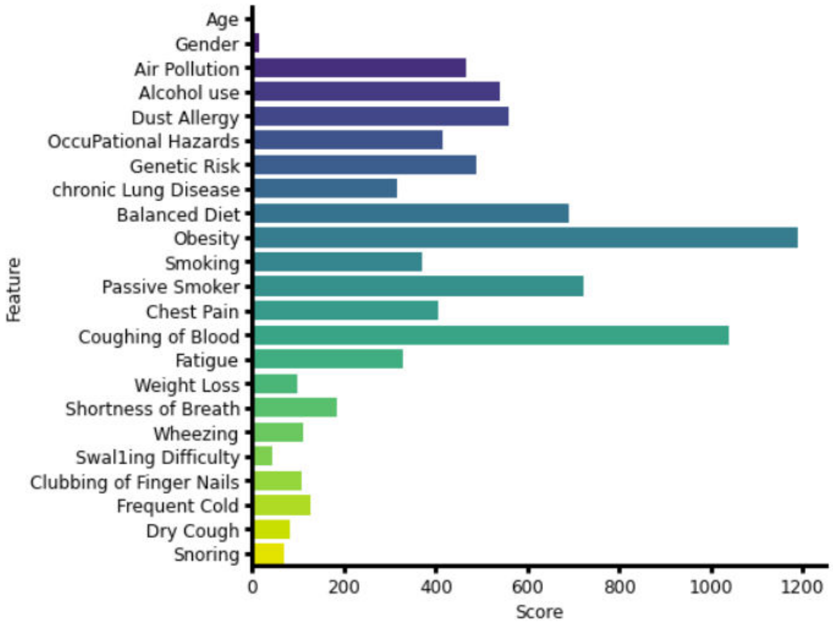
Feature Selection Code:



```
[ ] from sklearn.feature_selection import SelectKBest #Feature Selector
    from sklearn.feature_selection import f_classif #F-ratio statistic for categorical values
```

```
[ ] #Feature Selection
    X=lung_df.drop(['Dataset','Patient Id'], axis=1)
    Y=lung_df['Dataset']
    bestfeatures = SelectKBest(score_func=f_classif, k='all')
    fit = bestfeatures.fit(X,Y)
    dfscores = pd.DataFrame(fit.scores_)
    dfcolumns = pd.DataFrame(X.columns)
    #concat two dataframes for better visualization
    featureScores = pd.concat([dfcolumns,dfscores],axis=1)
    featureScores.columns = ['Feature','Score']  #naming the dataframe columns

    #Visualize the feature scores
    fig, ax=plt.subplots(figsize=(7,7))
    plot=sns.barplot(data=featureScores, x='Score', y='Feature', palette='viridis',linewidth=0.5, saturation=2, orient='h')
    Plotter(plot, 'Score', 'Feature', legend=False, save=True, save_name='Feature Importance.png')#Plotter function for aesthetics
    plot
```

```
No handles with labels found to put in legend.
<AxesSubplot:xlabel='Score', ylabel='Feature'>
```

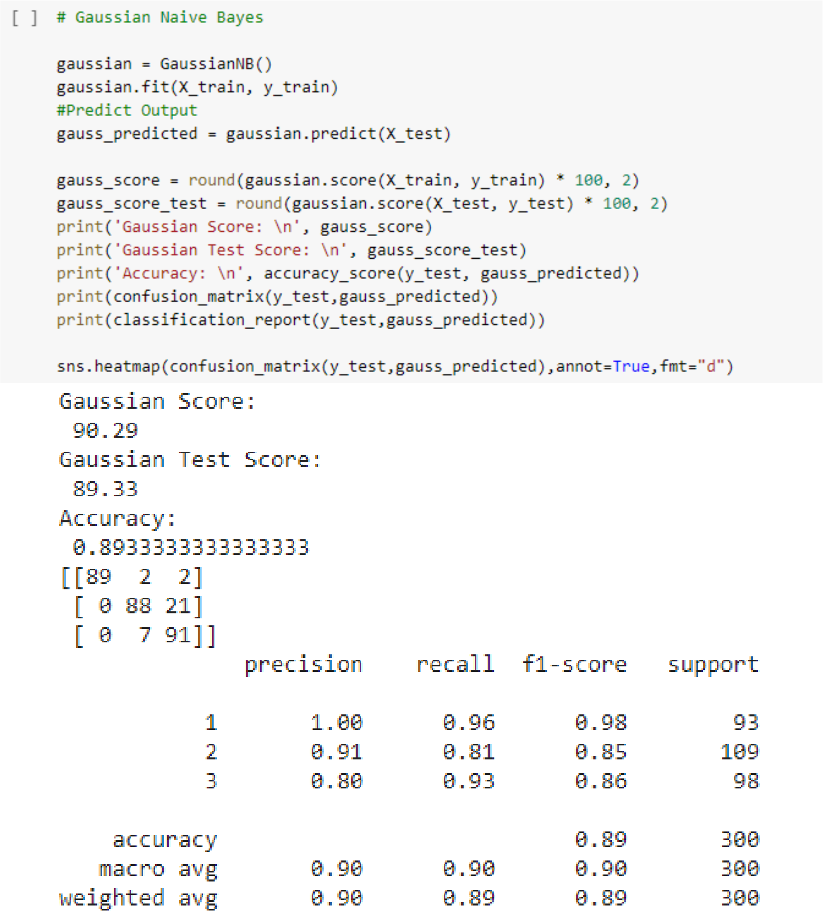


Feature Selection Graph

Results and Conclusion

As mentioned above we have imported various Scikit-Learn modules for training our algorithms After performing the Evaluation steps we have obtained the following results

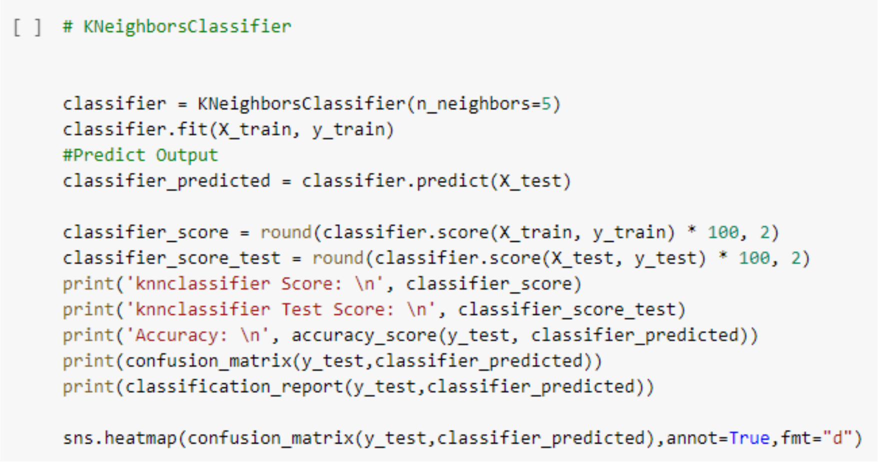
1) Gaussian Naïve Bayes:



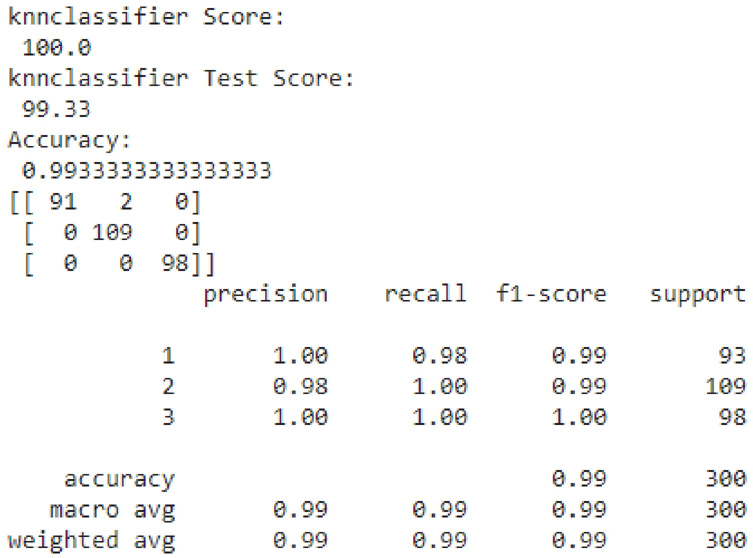
```
[ ]  # Gaussian Naive Bayes

     gaussian = GaussianNB()
     gaussian.fit(X_train, y_train)
     #Predict Output
     gauss_predicted = gaussian.predict(X_test)

     gauss_score = round(gaussian.score(X_train, y_train) * 100, 2)
     gauss_score_test = round(gaussian.score(X_test, y_test) * 100, 2)
     print('Gaussian Score: \n', gauss_score)
     print('Gaussian Test Score: \n', gauss_score_test)
     print('Accuracy: \n', accuracy_score(y_test, gauss_predicted))
     print(confusion_matrix(y_test,gauss_predicted))
     print(classification_report(y_test,gauss_predicted))

     sns.heatmap(confusion_matrix(y_test,gauss_predicted),annot=True,fmt="d")
```

```
Gaussian Score: 
 90.29
Gaussian Test Score: 
 89.33
Accuracy: 
 0.8933333333333333
[[89  2  2]
 [ 0 88 21]
 [ 0  7 91]]
              precision    recall  f1-score   support

           1       1.00      0.96      0.98        93
           2       0.91      0.81      0.85       109
           3       0.80      0.93      0.86        98

    accuracy                           0.89       300
   macro avg       0.90      0.90      0.90       300
weighted avg       0.90      0.89      0.89       300
```

2) K Nearest Neighbours Classifier:

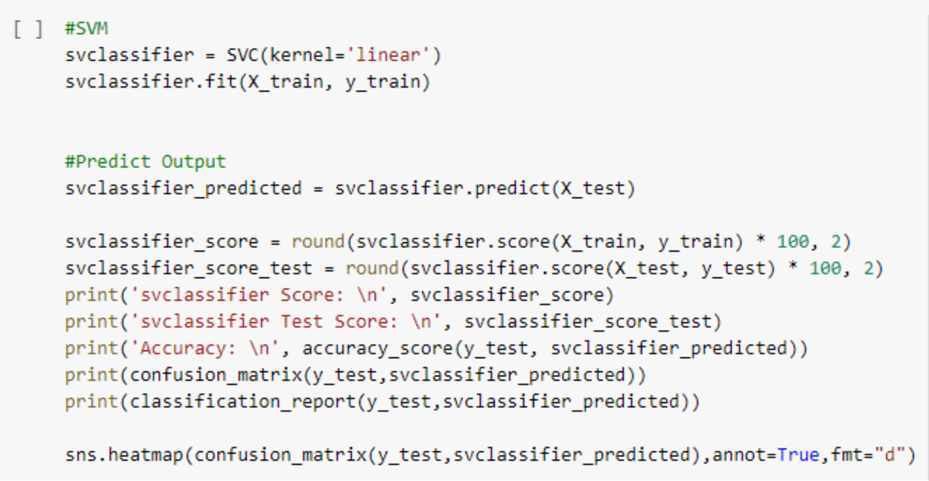


```
[ ]  # KNeighborsClassifier


     classifier = KNeighborsClassifier(n_neighbors=5)
     classifier.fit(X_train, y_train)
     #Predict Output
     classifier_predicted = classifier.predict(X_test)

     classifier_score = round(classifier.score(X_train, y_train) * 100, 2)
     classifier_score_test = round(classifier.score(X_test, y_test) * 100, 2)
     print('knnclassifier Score: \n', classifier_score)
     print('knnclassifier Test Score: \n', classifier_score_test)
     print('Accuracy: \n', accuracy_score(y_test, classifier_predicted))
     print(confusion_matrix(y_test,classifier_predicted))
     print(classification_report(y_test,classifier_predicted))

     sns.heatmap(confusion_matrix(y_test,classifier_predicted),annot=True,fmt="d")
```

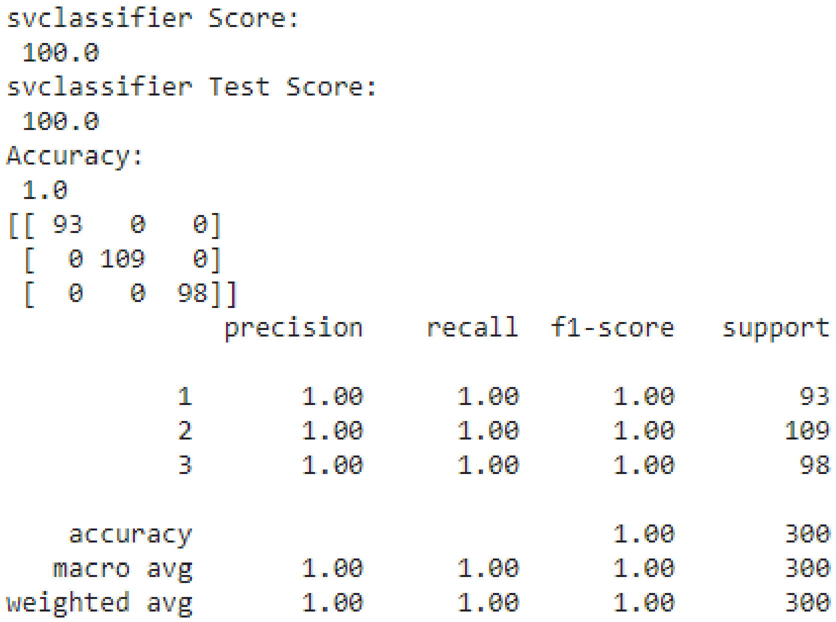


```
knnclassifier Score:
 100.0
knnclassifier Test Score:
 99.33
Accuracy:
 0.9933333333333333
[[ 91   2   0]
 [  0 109   0]
 [  0   0  98]]
              precision    recall  f1-score   support

           1       1.00      0.98      0.99        93
           2       0.98      1.00      0.99       109
           3       1.00      1.00      1.00        98

    accuracy                           0.99       300
   macro avg       0.99      0.99      0.99       300
weighted avg       0.99      0.99      0.99       300
```

3) Support Vector Machine:



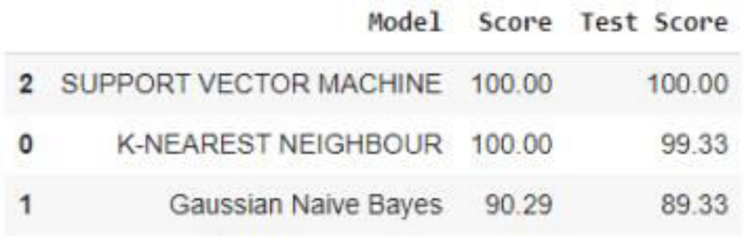
```
[ ] #SVM
    svclassifier = SVC(kernel='linear')
    svclassifier.fit(X_train, y_train)


    #Predict Output
    svclassifier_predicted = svclassifier.predict(X_test)

    svclassifier_score = round(svclassifier.score(X_train, y_train) * 100, 2)
    svclassifier_score_test = round(svclassifier.score(X_test, y_test) * 100, 2)
    print('svclassifier Score: \n', svclassifier_score)
    print('svclassifier Test Score: \n', svclassifier_score_test)
    print('Accuracy: \n', accuracy_score(y_test, svclassifier_predicted))
    print(confusion_matrix(y_test,svclassifier_predicted))
    print(classification_report(y_test,svclassifier_predicted))

    sns.heatmap(confusion_matrix(y_test,svclassifier_predicted),annot=True,fmt="d")
```



```
svclassifier Score:
 100.0
svclassifier Test Score:
 100.0
Accuracy:
 1.0
[[ 93   0   0]
 [  0 109   0]
 [  0   0  98]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        93
           2       1.00      1.00      1.00       109
           3       1.00      1.00      1.00        98

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

Final Observation and Accuracy

After Evaluating all the Models we can now rank all the models to choose the best one for our problem.



	Model	Score	Test Score
2	SUPPORT VECTOR MACHINE	100.00	100.00
0	K-NEAREST NEIGHBOUR	100.00	99.33
1	Gaussian Naive Bayes	90.29	89.33

We can finally conclude that Support Vector Machine suits best for our project as it gives the most accurate results.

References

1) Ada, R.K.: Early detection and prediction of lung cancer survival usingneural network classifier (2013).

2) https://www.ijitee.org/wpcontent/uploads/papers/v9i6/F3652049620.pdf

3) https://ieeexplore.ieee.org/abstract/document/9074947/

4) https://link.springer.com/article/10.1007/s00330-020-07141-9

5) https://ieeexplore.ieee.org/abstract/document/8479499/

6) https://ieeexplore.ieee.org/document/9074947

7) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6037965/

8) https://link.springer.com/chapter/10.1007%2F978-981-15-6648-6_11.

9) https://link.springer.com/article/10.1007/s10916-018-1139-7 Cancer: Prevention and Detection, Columbia Electron.Encycl., http:// www.infoplease.com/encyclopedia/science/cancer-medicineprevention-detection.html; 2012 [accessed August 11, 2016].

10) I M Nasser, S S Abu-Naser Predicting Tumor Category Using ArtificialNeural Networks International Journal of Academic Health and Medical Research (IJAHMR), volume 3, issue 2 Posted: 2019

11) Besbes, Ahmed, and Nikos Paragios. "Landmark-based segmentationof lungs while handling partial correspondences using sparse graphbased priors." In Biomedical Imaging: From Nano to Macro, International Symposium on, pp. 989-995.

12) Agrawal, A., Misra, S., Narayanan, R., Polepeddi, L., & Choudhary, A.(2011). A lung cancer outcome calculator using ensemble data miningon SEER data (pp. 19). Association for Computing Machinery (ACM).

13) https://ieeexplore.ieee.org/document/9074947

14) https://www.hindawi.com/journals/tswj/2015/786013/ https:// link.springer.com/chapter/10.1007%2F978-981-15-6648-6_11.