# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 4

Name: Ravi Varma Kumar Bevara

### **PART I: Model and Dataset Description**

Ensemble Learning

All the fundamental algorithms that we have performed in earlier assignments have their limitations because of their simplicity. In practice, deep neural networks require significant amount of data which we might not have. So, another approach to boost the performance of simple learning algorithms is ensemble learning. Ensemble learning is a learning paradigm that, instead of trying to learn on one super-accurate model, focuses on training large number of low-accuracy models and then combining the predictions given by those weak models to obtain a high-accuracy meta-model. The weak learner, which is used for Random Forest is 'Decision Tree' which is trained by splitting the training set after just few iterations.

Random Forest - It is a type of bagging that uses a modifies tree learning algorithm that inspects, at each split in the learning process, a random subset of the features.

* It is so effective because it is using multiple samples of the original dataset, so thereby reducing the variance.
* Low variances mean low overfitting.
* If the training set was samples, sometimes it may include the unavoidable artifacts such as noise, outliers etc.
* But, by creating multiple random samples with replacement of our training set, we reduce the effect of these artifacts.

The date is it used for this assignment is about the features computed from a digitize images in the process of fine needle aspirate of a breast mass. All the features describe the characteristics of the nucleus cells present in the blood which will help in detecting the tumor or cancer. The data set in total consists of 33 columns which includes an ID number indicating the clinical trial, the final diagnosis indicating the teamer presence, 10 real valued features which describes the cell nucleus in various dimensions.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Radius | Mean of distances from center to points of perimeter. |
| Texture | Standard deviation of gray scale values of images |
| Perimeter | Core size of the tumor causing cancer. |
| Area | Area of tumor |
| Smoothness | Local variation of radius lengths of tumor |
| Compactness | Perimeter^2 / Area -1 |
| Concavity | Severity of concave points of the contour |
| Concave points | Number of concave portions of the contour |
| Symmetry | Position of the contour to the edges |
| Fractal Dimension | Coastline approximation of the contour - 1 |

The data set includes three variants of the above mentioned 10 features such as mean, standard error and worst of these features when computed for each image. So, they result in 30 features in which we have a class in balance of 357 benign, and 212 malignant.

The bigger picture in this assignment is predict the classification of breast cancer presence based on the fine needle aspirate digitized images of patients.

### **PART II: Exploratory Data Analysis (Data Pre-Processing, Data Visualizations)**

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

A picture containing application

Description automatically generated

To better understand the variables, we need to look how the data is distributed and what’s the shape of the data. I continued with exploring the columns of the data. In this process, I recognized values of the target variable ‘quality’ are not balanced or ordered. The modules NumPy and pandas has various functions such as .head() to check the structure of the data. Also .columns will result all the columns of the dataset.



Table

Description automatically generated

After looking at the bird’s eye view of training and testing data, I decided to observe an exploring statistical measure such as standard deviation quartile distribution for each variable inside training and testing data.

Table

Description automatically generated

Now, we will go through the shape of the data using .shape function and we will check for the null values using isnull() function and also using the seaborn visualization.

Graphical user interface, text, application

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Also, we can see that there is a column named ‘Unamed 32’ which is all null values and an ID column which represents the clinical trial. So, they can be removed as they do not have any impact in model training and prediction.

Table

Description automatically generated

Then, I double checked the shape of the data and the information about each column or feature to justify that there are no null values in the data.

Table

Description automatically generated

Soon, after that I realized that the target variable is in categorical format, I decide to use that column into 0,1 values. Also, I decided to check for the class imbalance problem as it can cause potential difference in the stage of model training and evaluation.

Chart, pie chart

Description automatically generated

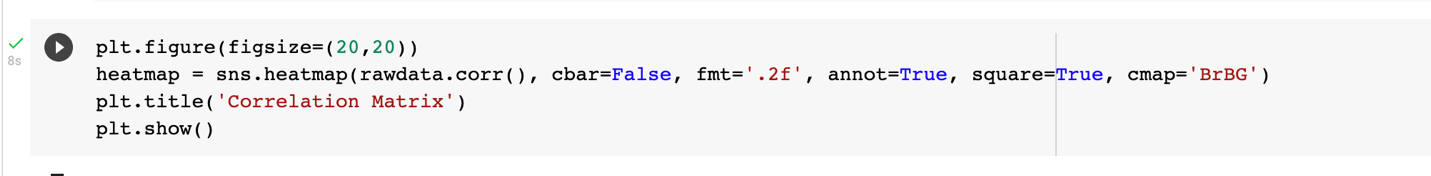
Table

Description automatically generated

Based on the above screenshot, we can clearly see that the given raw form of data is too much variant for each variable. So, we may need to use the scaling property to get all the data points into a same range, which will eventually gain an advantage in model training and testing.

Once, I came to know about the variables, then I wanted to get a better understanding of each variable in the dataset separately. I think the correlation plot would be a better choice to examine the relationships between target and other variables, along with histograms of each variable.

First step is to check how all the features are behaving with other attributes, using Pearson correlation coefficients.

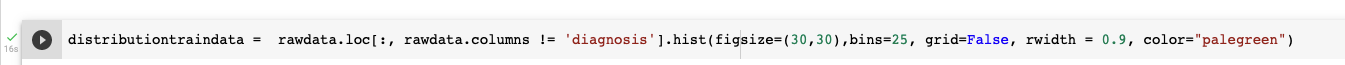


A screenshot of a computer

Description automatically generated with low confidence

So from the correlation matrix, we can see that there are so many positively correlated attributes and also it may cause the issue ‘Multicollinearity’. Mean columns of radius, perimeter, area are having a high correlation with the malignant tumor.

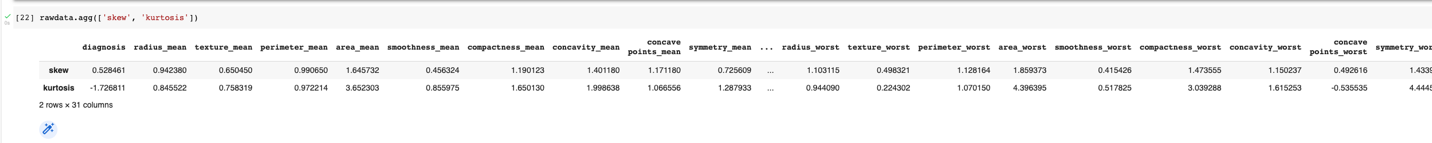
Moving forward, decided to observe the distributions of each feature in the data using histogram plot.



A picture containing window, green, surrounded, several

Description automatically generated

Although maximum of the features are right skewed, as we are performing the random forest classifier which uses the best algorithm of decision tree that can handle the variations and artifacts in the input data. This boosted my opinion for not performing any transformations on the features to convert them into uniform distributions. Then, I checked for the skewness and kurtosis of the raw data to conclude that the features are not normally distributed and from the below image we can see that the keratosis and skewness are significantly variable in the set of columns.



Strong linear relationship between the area perimeter and radius I decided to plot a pairplot from seaborn to disclose the relationship between the mean variables. From the results we can see that almost perfectly linear patterns between radius, very mature area attributes are hinting the presence of multicollinearity between these variables which means they are highly linearly related and that could possibly impact new model training. Another set of variables that possibly imply multicollinearity are concavity, concave points and compactness.

Graphical user interface, text, application, email

Description automatically generated

Multicollinearity is a problem as it undermines the significance of independent variables, and we fix it by removing the highly correlated predictors from the model. This can be done by using various methods such as partial least square regression or principal component analysis which will cut down the number of independent features that can be used to train the model. But here I choose to double check the multicollinearity problem using mask correlation matrix and manually remove or drop all the features from the data.

Calendar, whiteboard

Description automatically generated

We can verify the presence of multicollinearity between some of the variables. For instance, the mean of radius column has a correlation of 1 and 0.99 with perimeter mean an area mean columns respectively. This is because the three columns essentially contain the same information, which is the physical size of the observation. Therefore, we should only pick one of the three columns when we go into the further analysis.

A picture containing timeline

Description automatically generated

A picture containing outdoor

Description automatically generated

After looking into the correlation matrix result, we can list the variables with strong positive correlation, moderate positive correlation, and low positive correlation. Also, the same cases with the negative correlations. But due to the unclear image of the correlation matrix, I decided through all these strong positively correlated values which can cause the problem of multicollinearity.

I decided to drop all the worst columns for all 10 features because they are already involved in the calculations of mean and standard errors which are the other features used to train the model. So it might cause the duplication which will lead to overfitting of training the model.

A picture containing timeline

Description automatically generated

Chart, histogram

Description automatically generated

Once after removing the columns which are causing multicollinearity problem, I decided to evaluate using the same mask correlation matrix method.

Table

Description automatically generated

Even though I removed all the attributes which are causing multicollinearity, the statistical distribution of remaining attributes is also variant and ranging in different scales. So to overcome this problem we might need to use standardization before getting into the training as that will bring all the values to the same scale in the range (0,1)

Also, the class imbalance problem is still unsure as we can see from the below image that even in the clean data, we have a class imbalance.

Chart, pie chart

Description automatically generated

### **PART III: Model Training (Splitting the Data, Applying the Model)**

After a deep study in the packages available for model training in python, I decide to use sklearn also known as Scikit-Learn to train my random forest algorithms. After looking at the data in the preprocessing phase, I have decided to build 2 models and compare the best of the models by using different metrics and selection methods. To start off with, I will split the data into 80,20 proportions respectively for training and testing phases of the model. Also, I will use standard scaler feature to standardize the training and testing data.

Graphical user interface, text, application, email

Description automatically generated

Then I will train a base model using random forest classifier with no real setting of the parameter that will affect the model performance.

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a cell phone

Description automatically generated with low confidence

That’s absurd the base model has achieved an accuracy of more than 90%, we will not look into the classification report which will give us more details of metrics used in the classification such as precision recall an f1 score.

Table

Description automatically generated

I'm building random forest model now it is the time to kill the model using various hyper paramedics because random forest is one of the end symbol algorithms which will build a number of training models on the various random samples. Random forest comes with a numerous hyper parameter that can make the user to weak in the knees. But in this assignment, we will consider some bass hyper parrot meters of random forest which will have made it affect in model training those are…

* max\_depth
* bootstrap
* max\_features
* criterion

Maximum depth of a tree in random forest is defined as the longest path between the root note and lift note which means we are limiting until what death every tree in random Forest has to grow. What's the maximum depth value increases the performance over the test set increases initially but after a certain point it starts to decrease rapidly.

Bootstrap is a bully and variable that defines whether the sampling method used in building the trees is replacement sampling or not a bootstrap sampling.

Maximum features parameter represents the number of features to consider when the tree is looking for a best split, the search for a split does not stop until at least one valid partition of the note samples his phone, even if it requires effectively inspect more than the parameter set.

Criterion is a function to measure the quality of a split, which supports to values representing gini impurity and entropy for the information gain.

Graphical user interface

Description automatically generated with medium confidence

We didn’t consider the N estimators hyperparameters for a random forest algorithm because it is consuming most of the model training time, it describes the number of trees that has to be formulated in each iteration. Choosing a large number of estimators in the random forest is not the best idea because it will increase the computational complexity which happens in my case by extinguishing the CPU capacity.

Using the best parameters formulated by the grid search CV method, we will train a new team in the model to achieve better accuracy.

Graphical user interface, text

Description automatically generated with medium confidence

Chart

Description automatically generated with low confidence

As we see the accuracy of the tune model has increased from 90% to 93% by improving the recall rate.

For the curiosity, I decided to plot one of the trees in the estimators using some python modules.



Diagram

Description automatically generated

Now is the time to check the features importance using the feature importance capability of the random forest classifier and plotting the importance in ascending order.

Graphical user interface, text, application, email

Description automatically generated

For the construction of a random forest model, feature importance is a critical requirement. We can estimate variable relevance after the model has been trained. The interpretability provided by a single tree is lost when employing ensemble methods with decision trees. A single tree can show us critical node splits and variables that were significant at each split.

Chart, bar chart

Description automatically generated

**PART IV: Model Evaluation Metrics (Prediction Results, Test Scores, and Metrics)**

Machine learning model performance assessment is just like assessing this course, how are used to evaluate our schools in high schools and colleges for the meeting the eligibility criteria for getting the best courses or getting selected into the campus interviews for companies, etc. so apparently the good school recognizes the fact that the candidate is always good. The same is being expected in the machine learning model and that should achieve the expected result in predictions or forecasting or any other required automation in the problem statements.

Accuracy is just a number, for getting a better understanding of a prediction-based problem that corrects the predictions which are made by the model built by the team with the available number of records. So, we need to train the model across different combinations of data to get better accuracy.

Even though I’ve plotted the confusion matrix which is used for assessing model performance as it contains the true positive, false positive, true negative, and false negative it will depict that particular instance of the training phase as it may not be preferred when the model is overfitting.

So, the evaluation method that I choose in this assignment to evaluate the random forest classification algorithm is K Fold cross-validation. It means in each set of training and testing that would be performed precisely once during the entire process of building one decision tree we will be dividing the whole data set into K samples in which one sample will behold for testing that particular phase of training.

Graphical user interface, text, application, email

Description automatically generated

From the above picture, we can see that the model has been trained with 10 possible samplings of training data and the testing data and the overall average of all the models is almost the same as 93% which we achieved in tuning the hyperparameters. This is the fact that the random forest classifier algorithm which is being hyper tuned is not overfeeding the data set.

AUC-ROC curve is your performance measurement for declassification problems at various threshold settings. ROC is the probability go and AOC represents the degree or measure of separate ability. He tells how much the model is capable of distinguishing between classes. However, the UC, the better the model is a predicting or differentiating the various classes. By analogy, the higher the AOC, the better the model at distinguishing between patients with the disease and no disease.

Graphical user interface, chart

Description automatically generated

Even though the model accuracy is 93%, the AUCs code attained from the above picture is 97% which means the model has better accuracy in differentiating the patients with the disease and no disease.

### **PART V: Summary and Conclusion**

To conclude, we have trained two models one with base parameters which is achieving 90% accuracy, and the other model which is hyperparameter tuned using the GridSearchCV method using 4 parameters. So, we can also consider tuning other parameters for more information or to better the recall rate of the model. The top features indicate that radius, compactness, texture are the most important values that should be considered in deciding whether the patient is diseased or not. So, in my opinion, the clinical trials can conduct with different needle sizes in performing the FNA to achieve more accurate results.

We have now gone through a number of metrics to assess the capabilities of our random forest, but there is still much that can be done using background information from the data set. Feature in your knitting would be a powerful tool for extracting information and moving forward into the research phase and would help define key metrics to utilize when optimizing model parameters. They have been advancements with image classification in the past decade that make it possible to use images instead of extracted features from those images, but this data set is a great resource for making use of machine learning processes and concepts without the image data.

One of the greatest advantages of using random for us in the classification problem is that it is robust to outliers because of the random sampling method, and it handles both continuous and discrete variables equally well. Moreover, a random forest is insensitive to the scaling of features as well as any other monotonic transformation due to the random subspace selection. However, the trees are very sensitive to the noise in the input data call my drawback of using Random Forest or decision trees is that we need to avoid or fit by pruning, setting the minimum number of samples need to leave, or defining the maximum depth of the tree. In this assignment, we performed setting the maximum depth of the tree.

The final model is achieving the accuracy of 93% at a recall rate of 96% for no disease and 88% for diseased patients. But the better thing is the AUC score that defines the differentiability of the model in distinguishing the patients with the disease and without disease which is at 97%.