BREAST CANCER ANALYSIS

-By Shubhra Mahey

Introduction

Breast Cancer is one of the most commonly diagnosed cancer among the American women, other than skin cancer. It was also estimated that by 2019, that about 30% of newly diagnosed cancers in women will be breast cancers. Breast Cancer occurs as a result of abnormal growth of cells in the breast tissue, known as a "Tumor". A tumor does not mean cancer - tumors can be Benign (B, not cancerous), pre-malignant (pre-cancerous), or Malignant (M, cancerous). The dataset is showing some factors that might influence breast cancer. The dataset has historic medical records of 569 patients and 32 variables. It contains 569 samples of malignant and benign tumor cells.

Project Motivation and Goal

The labels in the data being discrete, the predication would fall into two categories, Malignant or Benign, making this is a classification problem. The goal, therefore, would be to classify whether the breast cancer is benign or malignant and predict the recurrence and non-recurrence of malignant cases after a certain period. To achieve this, I used certain machine learning classification methods that can predict the discrete class of a new input.

Data Cleaning

The data had one column “Unnamed: 32” which had NaNs and rest of the other variables, apart from id would be useful for the predictions. The patient “ids” might be a variable that would be needed while displaying the results, so we keep it. Other than this, none of the variables had NaNs, missing values or duplicates. Every data point is a unique value, with different patient ids. There are some skewed and correlated variables which are taken care of in the next steps.

Exploratory Data Analysis (EDA)

Of the 569 patients, there are 357 patients in the data that do not have cancer cells and 212 patients that show the presence of cancer cells. Using three different visualizations for plotting the data variables, different deductions are made. Plots like **Histograms** showed some Gaussian and Exponential trends, which are the variables that played an important role in the analysis later as Machine Learning techniques use these distributions on the input variables. The data is also skewed when it comes to variables like perimeter_se, radius_se, area_se, concavity_se, fractal_dimension_se, etc. This is observed through other plots like **box plot**, and **scatter plot** as well. It was better not to correct, standardize or remove the variables at the point because they might lose their meaning, or could play an important role in the prediction in the later stages. In **Box Plots**, there were a few outliers (dots above the clustered dots) above the notch for some variables which would ideally mean these variables have some problematic data under them which is incorrectly entered or measured while taking the patient’s data and should be removed. These outliers affect both results and assumptions but removing them at this stage might affect the predictions later. The **Scatter Plots** also show similar results about the outliers. There are plots that overlap or some that can be used in the classification of Malignant cancer cells from Benign. Outliers are not important if they capture inaccurate information, or if they carry little weight in the analysis. That cannot be said here by just looking at the data. So again, I kept all the variables because of no noticeable large outliers that show any strong distinction.

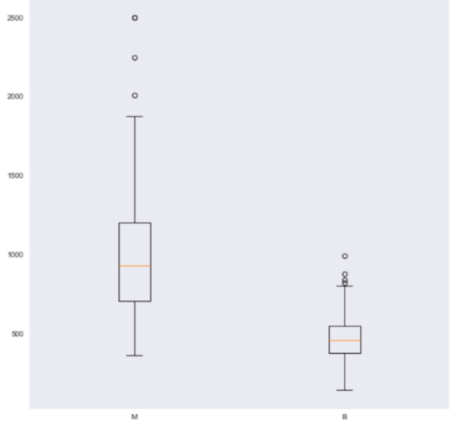


Image1

Next, the Correlation Plot between the variables of this data set show a strong correlation between a few variables like: strong positive correlation between mean_area and mean_perimeter, negative correlation of fractal_dimention_mean with mean values of radius, texture, and perimeter, etc. This can be clearly seen in the next section where these variables are reduced. Keeping both the variables that are correlated would not necessarily make much difference in the predictions. Furthermore, removing such variables would help reduce the number of variables to be dealt with and make an accurate prediction for Malignant or Benign cancer cells.

Feature Selection from Correlation Matrix:

The correlation heat maps below show all the correlated variables on a scale of -0.25 to 1. Variables like radius_mean, perimeter_mean and area_mean being highly correlated with each other (correlation>0.9, i.e. 90% correlated), tow of them can be dropped to reduce the number of variables as their presence would not make much difference in the predictions later. Similarly, considering all the correlated variables and keeping only one from each group having a correlation greater than 90% or 0.90 here, we get the heatmap of the final variables the predictions will be performed on. Now, the dataset consists of 16 independent variables, 1 dependent variable, i.e. "diagnosis" and 569 data points (patients).

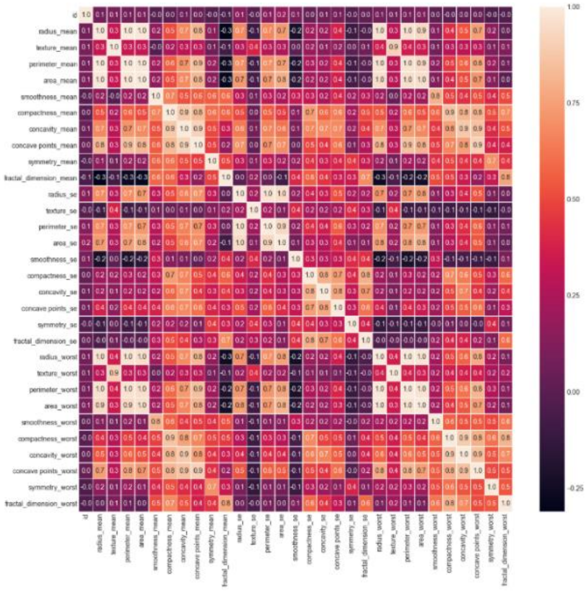


Image 2

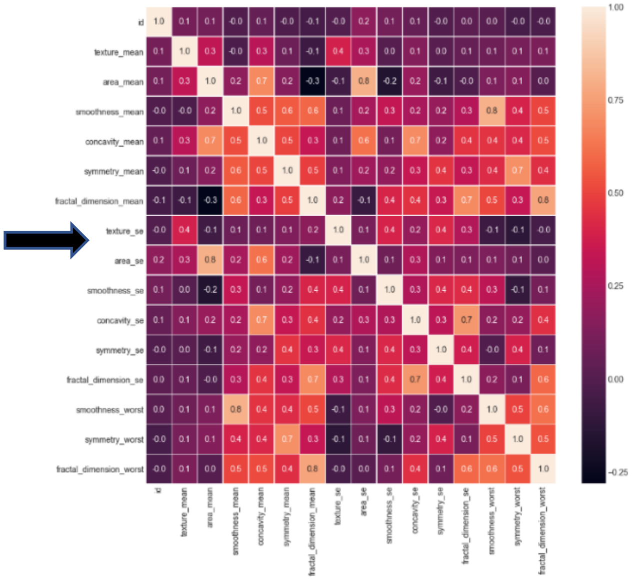


Image 3

After this, the next steps included encoding the class labels(diagnosis): Malignant tumors (M) now represented as class 1(i.e. cancer cells present) and the Benign tumors (B) are represented as class 0 (i.e. cancer cells absent). This is done just to make all data numeric for the predictive models.

PREDICTIVE MODELS

The data now split into training (80%) and testing (20%) set in order to assess the model accuracy. The model algorithms are trained on the first set and then predictions are made on the second test set, i.e. unseen data (to validate the model built). Data points in the training set are excluded from the test set. Using feature scaling, the data points that may be of varying magnitudes, range or units, are brought to the same scale, which is basically transforming the data to fit a specific scale of 0-1 or 0-100.

Model Selection

Different Machine Learning algorithms can be classified in two groups: supervised learning and unsupervised learning. For this dataset, since this is a classification dataset, I have used supervised learning algorithms for classification of the data. The process includes creating 6 base line models and get model performance for each of them, including their accuracy and other evaluation metrics like precision, recall, f1-score and support. These tell us the quality of the machine learning models. These models are then run again with the best parameters, which we get through Hyper-Parameter Tuning and GridSearch Cross Validation. Finally, these models, when ran on the best parameters, give the optimal model performance and give us the best predictors (variables) to classify the cell as Malignant or Benign. I also performed cross-validation to check if the model is unbiased towards the data. The Hyper-Parameter Tuning helps training the models better in order to increase their performance and accuracy, along with other factors like Precision and Recall. After that, all the models are combined to get the best accuracy for the dataset. In the end, the models are ensembled by allotting weighs to them, and multiplying the prediction probability with the wights of each model and adding them. This is shown in the end of all the analysis.

Hyperparameter tuning involves evaluating each model with their best parameter values, by trying different combinations that evaluate model performance. Evaluating only on the training set can lead overfitting, which is a very common problem in machine learning. If the model is optimized for the training data, then our model will give best scores on the training set but will not be able to generalize to the new unseen data, such as the test set. This is called **overfitting**, or in other words, creating a model that knows the training set very well but cannot be applied to real problems. Therefore, hyperparameter optimization takes care of overfitting through cross validation as well.

Table 1 below shows model metrics for 3 models: Random Forest, XGBoost, and Logistic Regression before and after performing hyper-parameter tuning and running the models with their best parameters. This ultimately improved the model performance. Base line models include models: Logistic Regression, Naïve Bayes, SVM (Support Vector Machine), KNN (K-Nearest Neighbors), XGBoost (eXtreme Gradient Boosting), and Random Forest. I selected these algorithms because the dataset is a binary classification data (M and B) and for such a problem, these algorithms topped the list.

Selecting Random Forest and XGBoost is basically because these are tree-based algorithms and thus, if there are more trees, it would not allow overfitting trees in the model. These methods increase the predictive power of the algorithm, avoid overfitting, and are amongst the most used algorithms in the industry. They also have very pruned results in the real-world applications. As for logistic regression, it is the best algorithm intended for binary (two-class) classification problems.

Table 1

	Model Evaluation Metrics (for M (1))	**Random Forest**	**XGBoost**	**Logistic Regression**
Without Hyper Parameter Tuning (Base Models)	**Accuracy**	0.9649	0.9824	0.9649
	Precision	0.98	0.98	0.93
	Recall	0.93	0.98	0.98
	F1-Score	0.95	0.98	0.95
With Hyper Parameter Tuning	**Accuracy**	0.9736	0.9824	0.9736
	Precision	1.00	0.98	0.95
	Recall	0.93	0.98	0.98
	F1-Score	0.96	0.98	0.97
Improvement in Accuracy after Hyper Parameter Tuning		0.91%	0.00%	0.91%

As we can see from the results above, the accuracy, precision and recall improved in the case of Random Forest and Logistic Regression but not in XGBoost. The accuracy is improved by 0.91%.

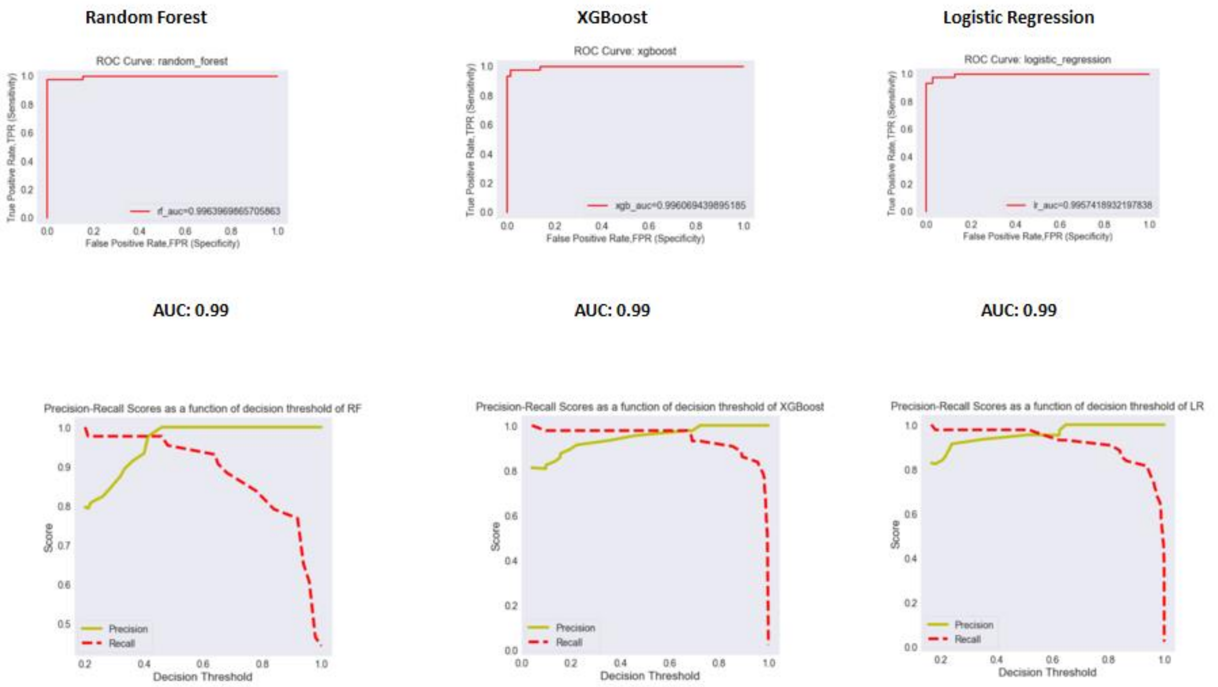


Image 4: ROC curve and AUC plots for the 3 models

Above are the ROC (receiver operating characteristic) curves and the AUC (Area Under the Curve) plots for all the 3 models I used. This is a very ideal case where the results are perfect as the data is not noisy. Overfitting has been taken care by cross-validation.

Feature Importance (optimal case, after hyper-parameter tuning)

Top 5 predictors for Malignant tumors according to Random Forest are:
area_mean, **concavity_mean**, **area_se**, **symmetry_worst**, and **texture_mean**.
According to XGBoost, the top 5 predictors come out to be:
area_mean, **concavity_mean**, **area_se**, **fractal_dimension_se**, and **smoothness_worst**.

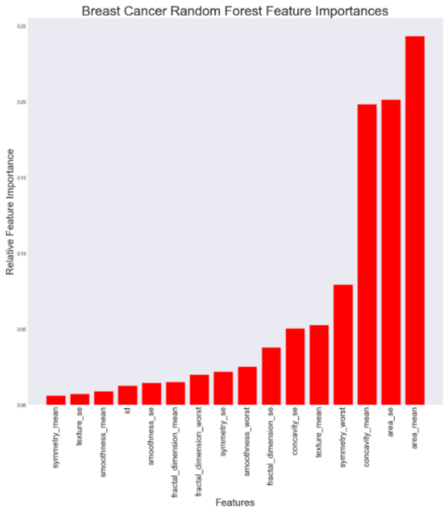


Image 5

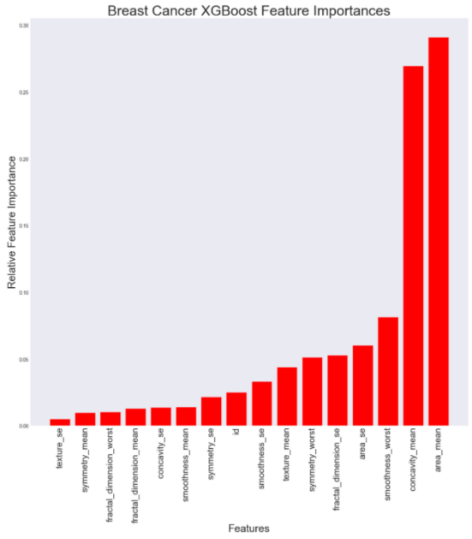


Image 6

The variable importance is calculated based on the **Mean Decrease Impurity (MDI) / Gini Importance** of the variables here. In decision trees (random forest is a decision tree), every node is a condition of how to split values in a single feature, so that similar values of the dependent variable end up in the same set after the split. The condition is based on impurity, which in case of classification problems is Gini impurity/information gain (entropy), while for regression trees its variance. So, when training a tree, we can compute how much each feature contributes to decreasing the weighted impurity. In the case of Random Forest, we are talking about averaging the decrease in impurity over trees.

Ensembling the 3 Models

Ensemble methods help to improve the machine learning results by combining the decisions from multiple models to improve the overall performance of the data prediction, as compared to a single model. There are various ensembling techniques like: Max Voting, Weighted Average, Stacking, Bagging, Boosting, etc. I used one of the methods, Max Voting. This method is generally used for classification problems. Here, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote', and the ones which we get from the majority of the models are used as the final prediction. The final ensembled accuracy from the 3 models come out to be 98%.

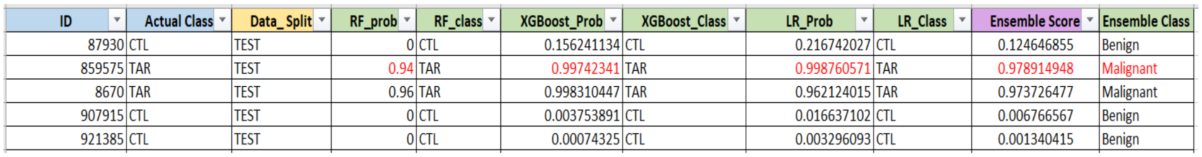
I also got the prediction classes and probabilities from each model and exported them to an excel sheet for making a dashboard that has results and conclusions.

Dashboard Summarization of Results

I categorized the "diagnosis" dependent variable as TAR (target patients, i.e. patients that have a cancerous tumor cell, M (1)) and CTL (control patients, i.e. patients that don't have cancer, B (0)). The data contains 114 patients (test set) on which the predictions are performed.

1) Raw Data

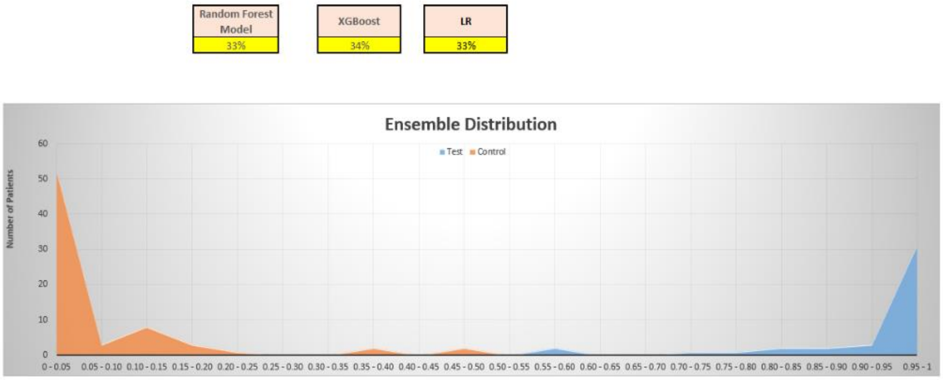
Table 2



ID	Actual Class	Data_Split	RF_prob	RF_class	XGBoost_Prob	XGBoost_Class	LR_Prob	LR_Class	Ensemble Score	Ensemble Class
87930	CTL	TEST	0	CTL	0.156241134	CTL	0.216742027	CTL	0.124646855	Benign
859575	TAR	TEST	0.94	TAR	0.99742341	TAR	0.998760571	TAR	0.978914948	Malignant
8670	TAR	TEST	0.96	TAR	0.998310447	TAR	0.962124015	TAR	0.973726477	Malignant
907915	CTL	TEST	0	CTL	0.003753891	CTL	0.016637102	CTL	0.006766567	Benign
921385	CTL	TEST	0	CTL	0.00074325	CTL	0.003296093	CTL	0.001340415	Benign

The Raw Data has the imported patient ids; the actual class of the data/patient (Malignant, TAR or Benign, CTL); the data split (TEST/TRAIN, in this case TEST); the Probabilities and Class for all the 3 classifying models: RF (Random Forest), XGBoost, and LR (Logistic Regression); and the calculated/resulting Ensembled Score and ensembled class (Malignant/Benign) for the Test set patient data.

2) Ensembler Tool



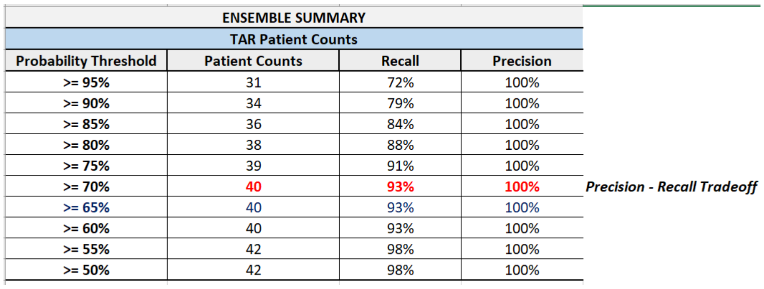
Based on the performance metrics, equal weihts have been given to all three models.

The ensembled score is calculated based on weighted average ensembling method. For instance, taking a data point from Table 2 above (2nd observation color coded: red), we have rf_prob (probability of random forest for that patient), xgb_prob (probability from xgboost) and lr_prob (probability from logistic regression) as 0.94, 0.99742341 and 0.998760571 respectively. The score is calculated by multiplying individual probabilities by the respective model weight assigned and adding them, i.e.

0.94 (0.33) + 0.99742341 (0.33) + 0.998760571 (0.34) = 0.978914948,

which is closer to 1, hence Malignant (1) Ensembled class. This is how the prediction ensembled probability and class is found out for each patient, and so we get the above **Ensembled Probability Distribution.**

3) Ensemble Summary

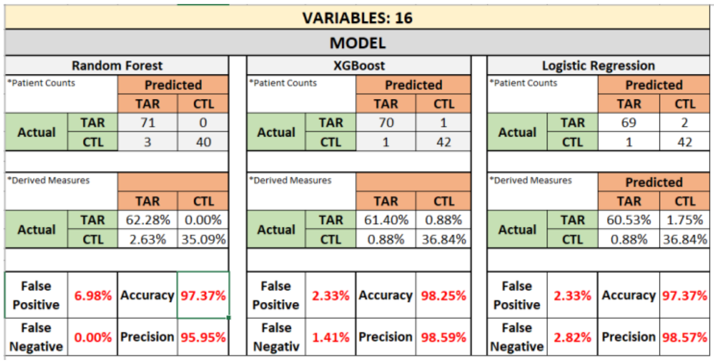


ENSEMBLE SUMMARY			
TAR Patient Counts			
Probability Threshold	Patient Counts	Recall	Precision
>= 95%	31	72%	100%
>= 90%	34	79%	100%
>= 85%	36	84%	100%
>= 80%	38	88%	100%
>= 75%	39	91%	100%
>= 70%	40	93%	100%
>= 65%	40	93%	100%
>= 60%	40	93%	100%
>= 55%	42	98%	100%
>= 50%	42	98%	100%

Precision - Recall Tradeoff

Ensemble Summary shows the Precision-Recall trade-off for the data. I came to an optimal number of patients (which usually depends on the client, if they want a greater number of patients while sacrificing on the recall/precision or need a balanced precision-recall as well as patient counts). Here, I selected 40 patients with probability threshold >=70% for classifying the data as M or B, and with a recall of 93% and a 100% precision (which is highly ideal, but the model was well-trained, data was not noisy and the overfitting has been taken care by cross-validation).

4) Confusion Matrix of the 3 models



VARIABLES: 16

MODEL

Random Forest

*Patient Counts		Predicted	
		TAR	CTL
Actual	TAR	71	0
	CTL	3	40

*Derived Measures			
		TAR	CTL
Actual	TAR	62.28%	0.00%
	CTL	2.63%	35.09%

False Positive	6.98%	Accuracy	97.37%
False Negative	0.00%	Precision	95.95%

XGBoost

*Patient Counts		Predicted	
		TAR	CTL
Actual	TAR	70	1
	CTL	1	42

*Derived Measures			
		TAR	CTL
Actual	TAR	61.40%	0.88%
	CTL	0.88%	36.84%

False Positive	2.33%	Accuracy	98.25%
False Negativ	1.41%	Precision	98.59%

Logistic Regression

*Patient Counts		Predicted	
		TAR	CTL
Actual	TAR	69	2
	CTL	1	42

*Derived Measures		Predicted	
		TAR	CTL
Actual	TAR	60.53%	1.75%
	CTL	0.88%	36.84%

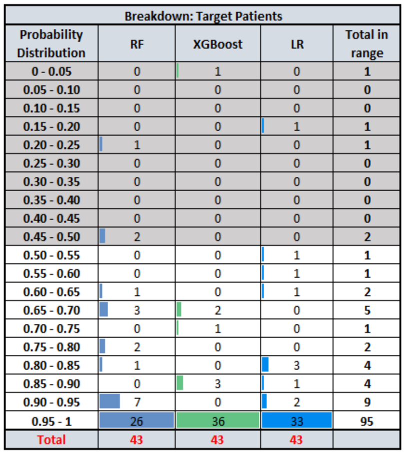
False Positive	2.33%	Accuracy	97.37%
False Negative	2.82%	Precision	98.57%

This shows the confusion matrix of the 3 main models I used for Cancer prediction namely, Random Forest, XGBoost, and Logistic Regression and their derived measures along with Accuracy and Precision.

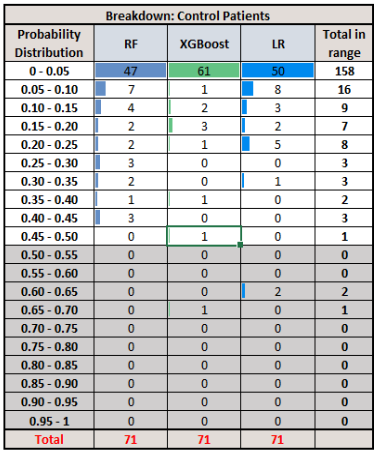
5) Probability Distribution

The probability Distribution shows the distribution of the Target and Control Patients by the 3 models in the test dataset with total of 43 patients as Malignant and 71 patients as Benign.

As we see, for the Target patients, 26 patients are predicted as Malignant having probability between 0.95-1 by Random Forest, 36 by XGBoost and 33 by Logistic Regression.



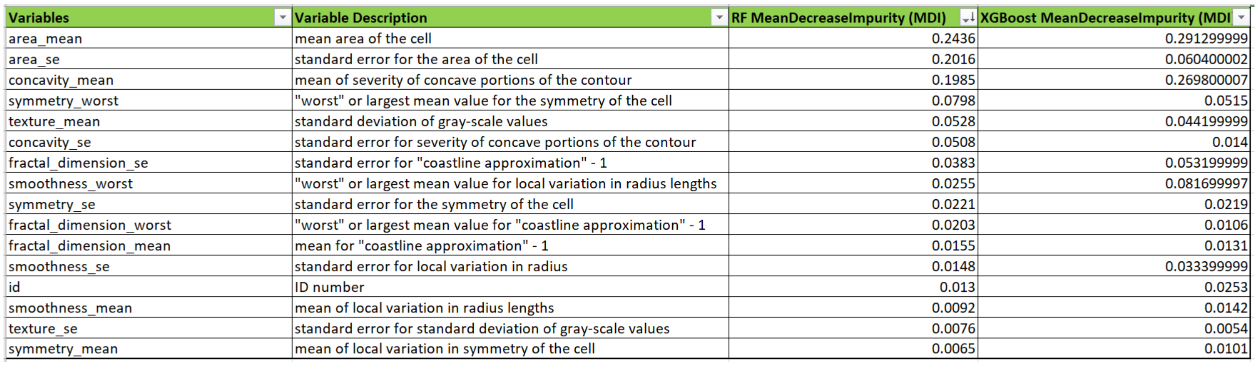
Breakdown: Target Patients				
Probability Distribution	RF	XGBoost	LR	Total in range
0 - 0.05	0	1	0	1
0.05 - 0.10	0	0	0	0
0.10 - 0.15	0	0	0	0
0.15 - 0.20	0	0	1	1
0.20 - 0.25	1	0	0	1
0.25 - 0.30	0	0	0	0
0.30 - 0.35	0	0	0	0
0.35 - 0.40	0	0	0	0
0.40 - 0.45	0	0	0	0
0.45 - 0.50	2	0	0	2
0.50 - 0.55	0	0	1	1
0.55 - 0.60	0	0	1	1
0.60 - 0.65	1	0	1	2
0.65 - 0.70	3	2	0	5
0.70 - 0.75	0	1	0	1
0.75 - 0.80	2	0	0	2
0.80 - 0.85	1	0	3	4
0.85 - 0.90	0	3	1	4
0.90 - 0.95	7	0	2	9
0.95 - 1	26	36	33	95
Total	43	43	43	



Breakdown: Control Patients				
Probability Distribution	RF	XGBoost	LR	Total in range
0 - 0.05	47	61	50	158
0.05 - 0.10	7	1	8	16
0.10 - 0.15	4	2	3	9
0.15 - 0.20	2	3	2	7
0.20 - 0.25	2	1	5	8
0.25 - 0.30	3	0	0	3
0.30 - 0.35	2	0	1	3
0.35 - 0.40	1	1	0	2
0.40 - 0.45	3	0	0	3
0.45 - 0.50	0	1	0	1
0.50 - 0.55	0	0	0	0
0.55 - 0.60	0	0	0	0
0.60 - 0.65	0	0	2	2
0.65 - 0.70	0	1	0	1
0.70 - 0.75	0	0	0	0
0.75 - 0.80	0	0	0	0
0.80 - 0.85	0	0	0	0
0.85 - 0.90	0	0	0	0
0.90 - 0.95	0	0	0	0
0.95 - 1	0	0	0	0
Total	71	71	71	

Also, when we look at the Control patients, Random Forest predicted 47 patients as Benign, i.e. having probabilities between 0-0.05. Similarly, XGBoost predicted 61 and Logistic Regression, 50.

6) Variable Importance



Variables	Variable Description	RF MeanDecreaseImpurity (MDI)	XGBoost MeanDecreaseImpurity (MDI
area_mean	mean area of the cell	0.2436	0.291299999
area_se	standard error for the area of the cell	0.2016	0.060400002
concavity_mean	mean of severity of concave portions of the contour	0.1985	0.269800007
symmetry_worst	"worst" or largest mean value for the symmetry of the cell	0.0798	0.0515
texture_mean	standard deviation of gray-scale values	0.0528	0.044199999
concavity_se	standard error for severity of concave portions of the contour	0.0508	0.014
fractal_dimension_se	standard error for "coastline approximation" - 1	0.0383	0.053199999
smoothness_worst	"worst" or largest mean value for local variation in radius lengths	0.0255	0.081699997
symmetry_se	standard error for the symmetry of the cell	0.0221	0.0219
fractal_dimension_worst	"worst" or largest mean value for "coastline approximation" - 1	0.0203	0.0106
fractal_dimension_mean	mean for "coastline approximation" - 1	0.0155	0.0131
smoothness_se	standard error for local variation in radius	0.0148	0.033399999
id	ID number	0.013	0.0253
smoothness_mean	mean of local variation in radius lengths	0.0092	0.0142
texture_se	standard error for standard deviation of gray-scale values	0.0076	0.0054
symmetry_mean	mean of local variation in symmetry of the cell	0.0065	0.0101

This shows the importance of all the variables with their MDI (Mean Decrease Impurity) from the code, along with their variable descriptions. When sorted, the **top variables** for the prediction of Malignant or cancerous tumor cells come out to be: **area_mean, area_se, concavity_mean, symmetry_worst, texture_mean, smoothness_worst, and fractal_dimension_se** (from both XGBoost and Random Forest, as shown in the python code).