Final Report

Malignant or Benign?

Table of Contents

[Research Questions 3](#_Toc130450324)

[Malignant or benign? 3](#_Toc130450325)

[What technique should be used? 3](#_Toc130450326)

[Which features should be chosen? 3](#_Toc130450327)

[Which algorithm (model) should be applied? 3](#_Toc130450328)

[The main contribution of the work compared to past research 4](#_Toc130450329)

[Algorisms 4](#_Toc130450330)

[Feature Selection 4](#_Toc130450331)

[Scale 4](#_Toc130450332)

[The link to the GitHub repository where the source code files are checked in 5](#_Toc130450333)

[A more precise description of the applied methodology and the study design 6](#_Toc130450334)

[Applied methodology 6](#_Toc130450335)

[Methodology 6](#_Toc130450336)

[Technical Tool 6](#_Toc130450337)

[Study design 7](#_Toc130450338)

[The conducted analyses, including all activities and their business 8](#_Toc130450339)

[Data Exploration 8](#_Toc130450340)

[Data cleaning & preparation 8](#_Toc130450341)

[Feature selection 9](#_Toc130450342)

[Correlation 9](#_Toc130450343)

[Gain Ratio 9](#_Toc130450344)

[Scale data 10](#_Toc130450345)

[Split the dataset 11](#_Toc130450346)

[Build models 11](#_Toc130450347)

[Evaluate models 12](#_Toc130450348)

[Repeat the process 12](#_Toc130450349)

[A list of all the findings, including a detailed interpretation of the results of the applied techniques 13](#_Toc130450350)

[Correlation 13](#_Toc130450351)

[Gain Ratio 14](#_Toc130450352)

[Combine methods 15](#_Toc130450353)

[Evaluate models 16](#_Toc130450354)

[70%-30% Split 16](#_Toc130450355)

[60%-40% Split 17](#_Toc130450356)

[80%-20% Split 18](#_Toc130450357)

[Combination 19](#_Toc130450358)

[Repeat the process 20](#_Toc130450359)

[Models comparison 23](#_Toc130450360)

[The shortcomings of the work and concluding remarks on the continuity of the work 24](#_Toc130450361)

[Imbalanced 24](#_Toc130450362)

[Skewed data 24](#_Toc130450363)

[Reference 25](#_Toc130450364)

# Research Questions

## Malignant or benign?

This is the main questions of this research. To answer it, I created and solved some questions to meet the target:

## What technique should be used?

Machine Learning is the best choice. This technique is commonly used to make predictions as we have countless data in the real world and it is significantly growing everyday.

## Which features should be chosen?

Real world data is always messy. I need to clean it up, filter out the noise and choose the right features before use. This can improve the performance of models and avoid overfitting / underfitting.

## Which algorithm (model) should be applied?

Choosing the best model is not easy. But I can feed them all and compare their performances. The one with highest score will be my final model.

# The main contribution of the work compared to past research

## Algorisms

Past researches used few algorisms to build models such as SVM, KNN and Decision Tree. I used more like Logistic Regression, Random Forest, Gaussian Naïve Bayes and so on. So that I had more models to compare the performances.

## Feature Selection

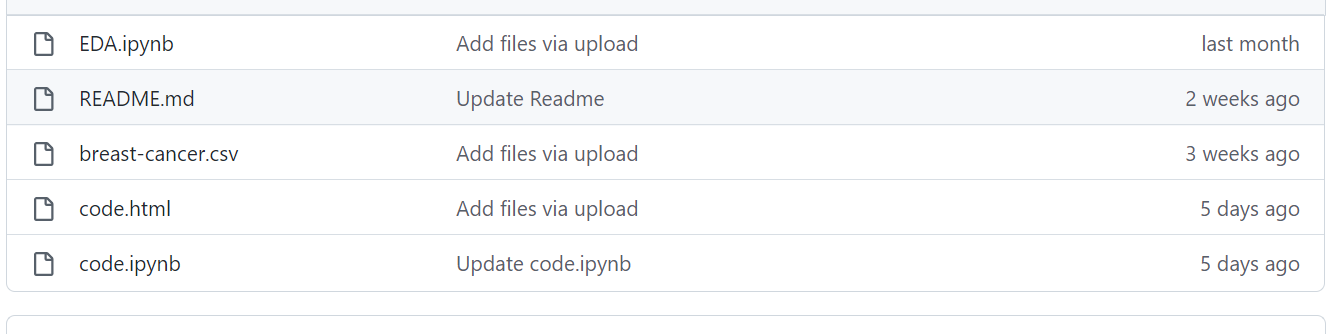
Past researches used single method for feature selection such as correlation, information gain or gain ratio. I used a combination of these methods to choose features. It was probably better than a single one.

## Scale

Past researches did not scale data before fit models. Scaling data is important as not all the features at the same level. It significantly affected the performances of some models without scaling like Logistic Regression and KNN.

# The link to the GitHub repository where the source code files are checked in

Link: <https://github.com/checkming00/Ryerson_Final_Project>



“breast-cancer.csv” is the dataset file.

<https://github.com/checkming00/Ryerson_Final_Project/blob/main/breast-cancer.csv>

“EDA.ipynb” is the code file for data exploration.

<https://github.com/checkming00/Ryerson_Final_Project/blob/main/EDA.ipynb>

“code.ipynb” is the code file for the entire process of machine learning including cleaning, scaling, building models, training models, model evaluation and model comparison.

<https://github.com/checkming00/Ryerson_Final_Project/blob/main/code.ipynb>

“code.html” is the output HTML format of “code.ipynb”.

<https://github.com/checkming00/Ryerson_Final_Project/blob/main/code.html>

# A more precise description of the applied methodology and the study design

## Applied methodology

### Methodology

Machine Learning is the methodology I chose. It is a mutual technique for making predictions which is the best fit for this project.

Machine Learning is based on history data and statistical algorisms. It uses history data and apply an appropriate algorism to fit models (or train models). The trained models then are able to make predictions while fit in new data.

### Technical Tool

I used Python as the technical tool. Python is a popular programming language. It is commonly used in data science field because it is easy and concise.

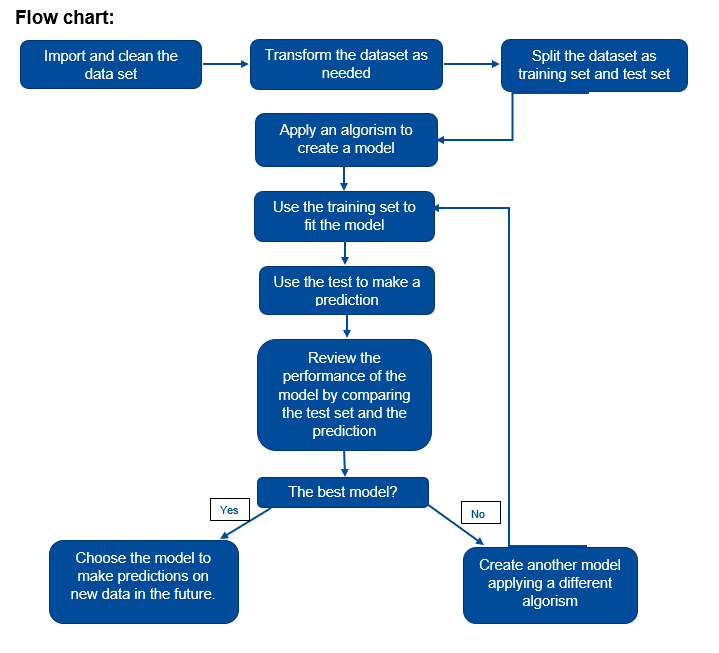
I imported some external packages:

1. Pandas: Pandas is a powerful package. It can be used to manipulate 2-D data structure like spreadsheet in this project. Pandas helped me to explore and clean the data in just a few codes.
2. Matplotlib and seaborn: Both packages are for visualization. They were applied in EDA report to create graphs. Which helped me to learn the dataset intuitively.
3. Sklearn: Or scikit-learn. It is one of the most popular packages for machine learning. I used sklearn in my project to scale data, split data into training set and test, building models and evaluate models.
4. Numpy: Numpy is a base of Pandas. It is used for array manipulation, math calculation and statistical calculation and so on.

## Study design

I followed the steps for the study design:

1. Data import and cleaning (handle null value, drop noise, etc.)
2. Data transformation (convert data type, scale, feature selection)
3. Split the dataset into training set and test set
4. Create models applying algorisms
5. Use the training set to fit the models
6. Use the test set to make predictions
7. Models evaluation
8. Compare performances of models
9. Choose the best model



# The conducted analyses, including all activities and their business

## Data Exploration

* Discover the shape of the dataset (569 X 32). It is for feature selection and the percentage of training and test split.
* Explore the data type of features. It is to decide if the dataset needs further transformation (Convert non-numeric to numerical values).
* Count the unique values of the class to see if the dataset balanced or not.
* Statistical summary including average, standard deviation, minimum, maximum and quarterly percentiles.
* Use histograms to check distribution of all the columns. The histograms can help to see if an attribute normal, left skewed or right skewed.
* Draw a heatmap to see the correlation between attributes, especially between features and the class. It is important for feature selection.

## Data cleaning & preparation

* Import the dataset. The data was saved in CSV format. It was imported and converted to a dataframe type.
* Drop the noise. “id” column is the unique identify numbers of patients. It does not affect the class. So “id” column is a noise should be dropped.
* Convert data type. “diagnosis” column which is also the class has 2 unique string values: M and B. Models in sklearn package can only be fit by numerical values. So I convert the values to integer type: M to 1 and B to 0.

## Feature selection

I used a combination of methods to select features: correlation and Gain Ratio.

### Correlation

Correlation is any statistical relationship, whether causal or not, between two random variables or bivariate data. It means a measure of a feature affects the class.

### Gain Ratio

Information gain measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. It is calculated as:





where 𝑝𝑖 is the ratio of the number of instances of the class in the S to the number of all samples in the S set. c is the number of classes. Values (A) is a set of all possible values of feature A; 𝑆𝑣 is a set of instances with v value of A feature in the S set.

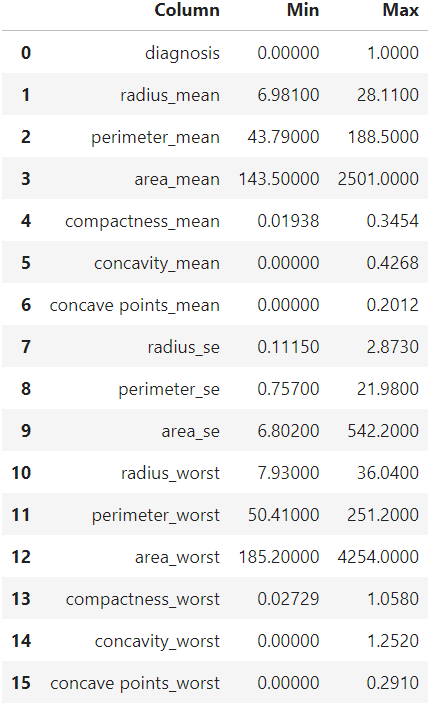
Gain Ratio is modification of information gain that reduces its bias. It can be calculated by:





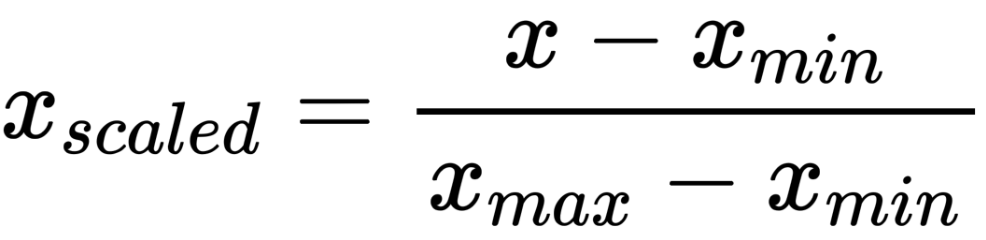
Gain Ratio is used in Decision Tree algorism. Nowadays, it is expanded to be an option for feature selection.

## Scale data



The min-max list shows that the features and the class are not at the same level. It is necessary to scale the dataset.

Here min-max scaler method was used. It is calculated as:



After scaling, all the columns are at the same level ranged from 0 to 1.

## Split the dataset

Before fitting models, the dataset needs to be split into training set and test set.

The training set is for fitting models. The test set is for evaluating models.

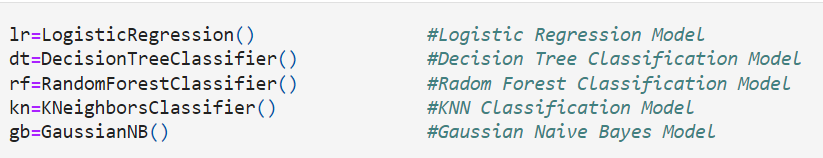
Each record was randomly chosen to be in either training set or test set.

The problem is what ratio of training and test set contained. Larger training data can fit models well, but not enough test data to evaluate the models. Larger test data has sufficient “new” data the evaluate models but maybe not fit models well.

In this study, the dataset has 569 records. A regular test set portion as 20%-40% can be considered. Thus, I used 70%-30%, 60%-40%, 80%-20% to split the data respectively and compare their performances.

## Build models

Simply initial 5 models with sklearn package: Logistic Regression, Decision Tree, Random Forest, KNN and Gaussian Naïve Bayes.



Then fit the models with training data and make predictions with test data.



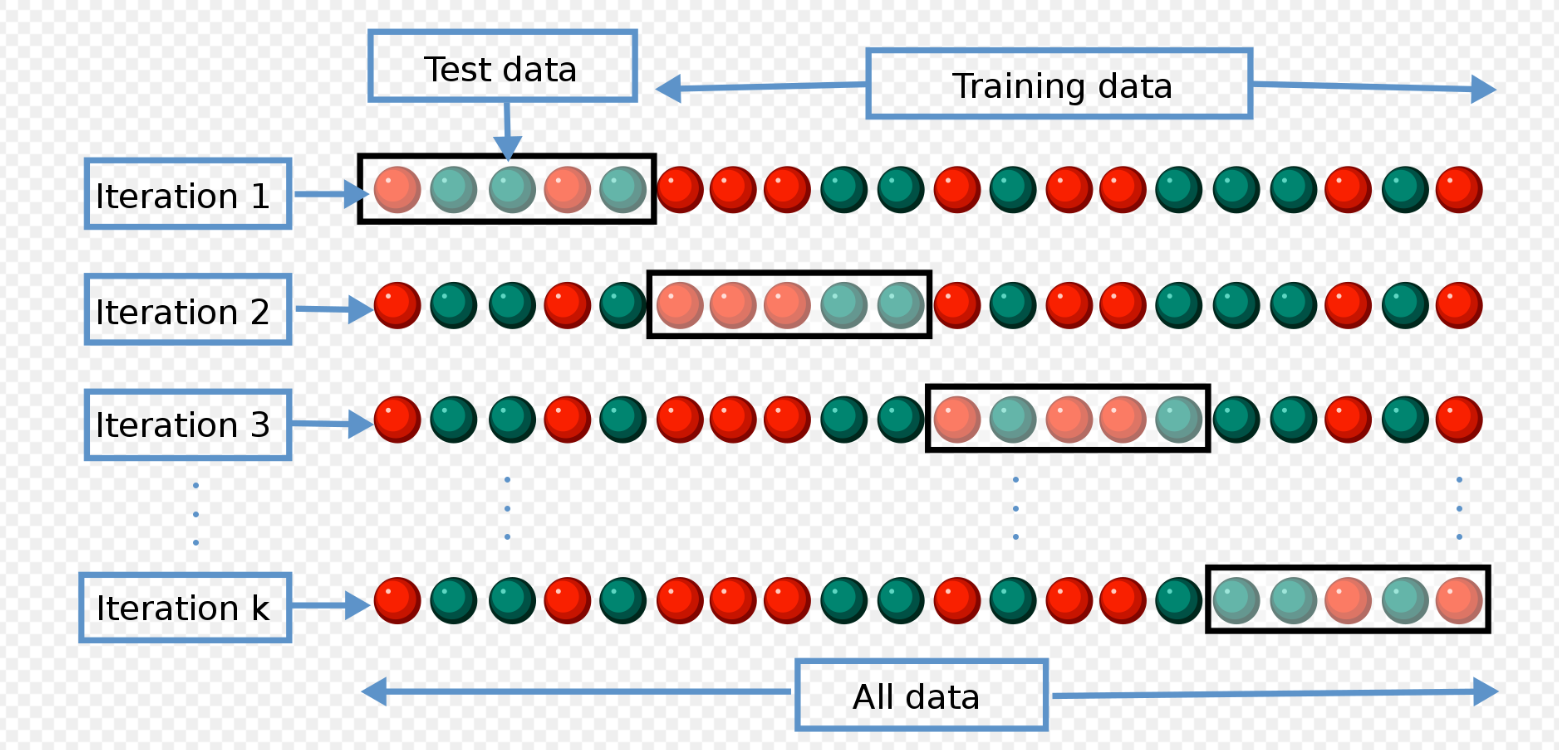
The predictions are a list of predicted values with the same size as the test data. The list can be used to compare to the test data to evaluate models.

## Evaluate models

I used accuracy and cross validation score to evaluate models in this study.

Accuracy is the ratio of correct predictions.

Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. It can be explained by the figure below:



In this study, k=3 to 5 folders were used with the split 60%-40%, 70%-30% and 80%-20% respectively. It means a total of k models were trained and evaluated for every calculation. The average of the k accuracy scores is the cross validation score.

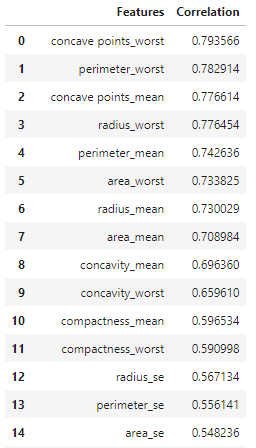
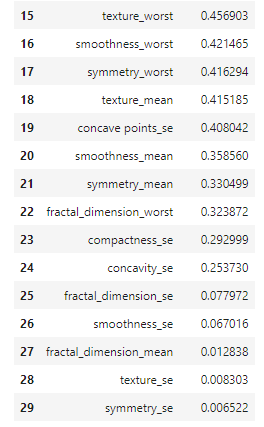
## Repeat the process

Repeat the process from split dataset to models evaluation to see if the result will be stable or not. In this study, it was repeated 5 times.

# A list of all the findings, including a detailed interpretation of the results of the applied techniques

## Correlation

Listed the correlation between all the features and the class, and sorted the list by descending order:

The list shows the features from most related to least related to the class. (from maximum 0.8 to minimum 0.0065)

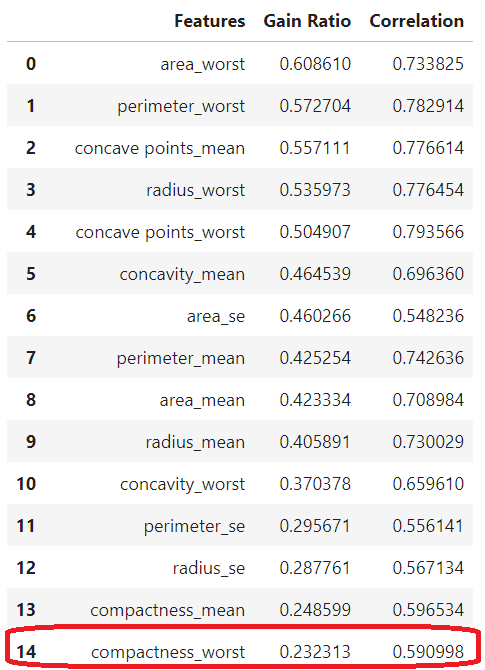
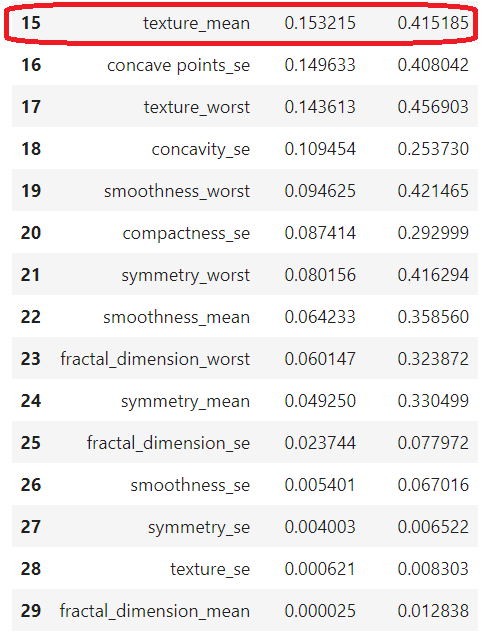
## Gain Ratio

I calculated the Gain Ratio for all features following the formulas above. Then sorted it by descending order:

## Combine methods

The final step is to combine these 2 methods. Merged 2 lists and sorted by Gain Ratio first then correlation by descending order:

As the list shows, correlation is under 0.5 while Gain Ratio is less than 0.2 (begins from feature “texture\_mean”).

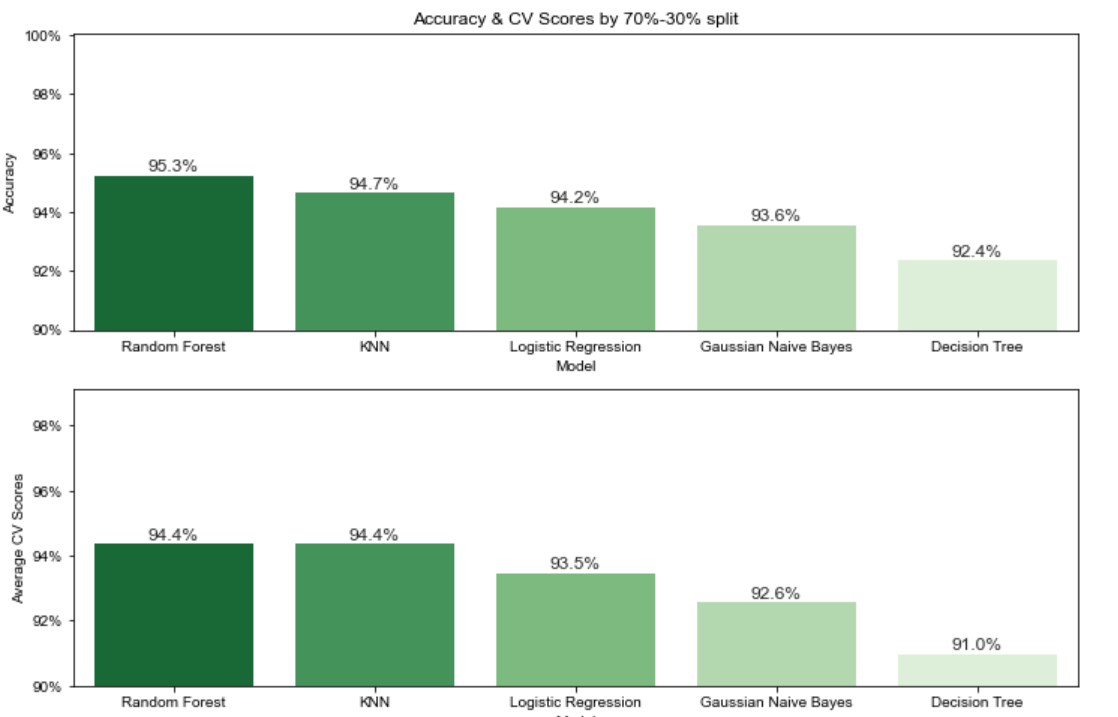
Correlation is weakness when it is under 0.5 and 20% of a Gain Ratio means it can get sufficient information after split.

Thus, I chose 0.2 Gain Ratio as the threshold to select features. That being said, the features with Gain Ratio greater than 0.2 were selected, totally 15 features in this case.

## Evaluate models

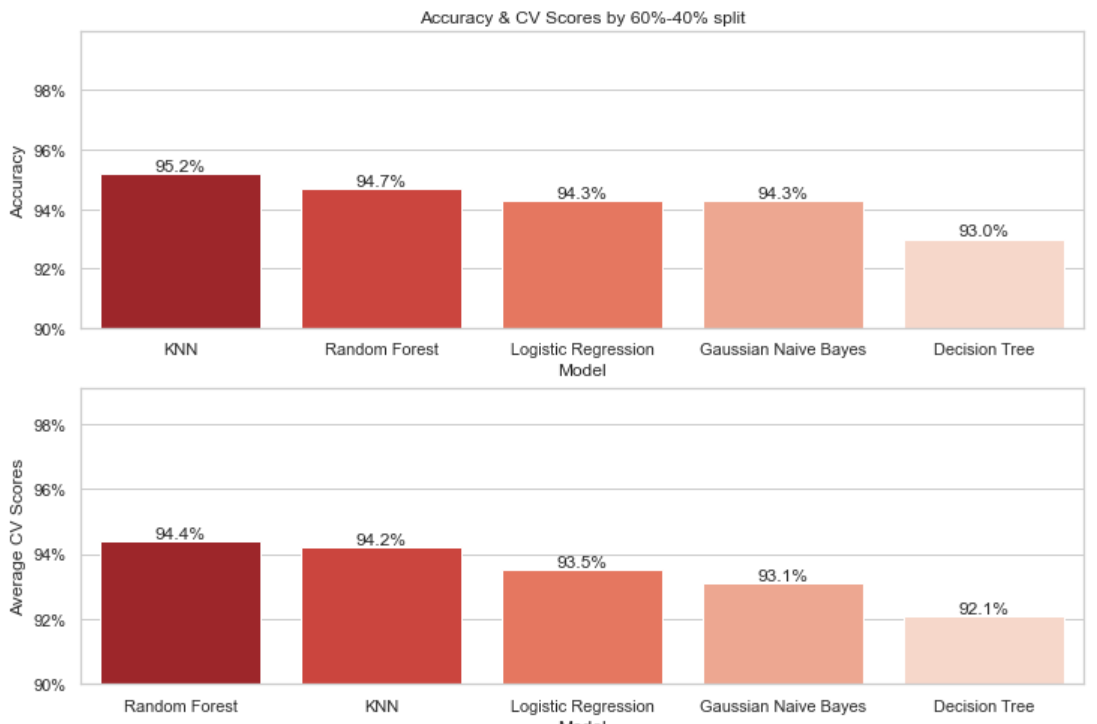
To compare models intuitively, I produced visualization of the performances.

### 70%-30% Split



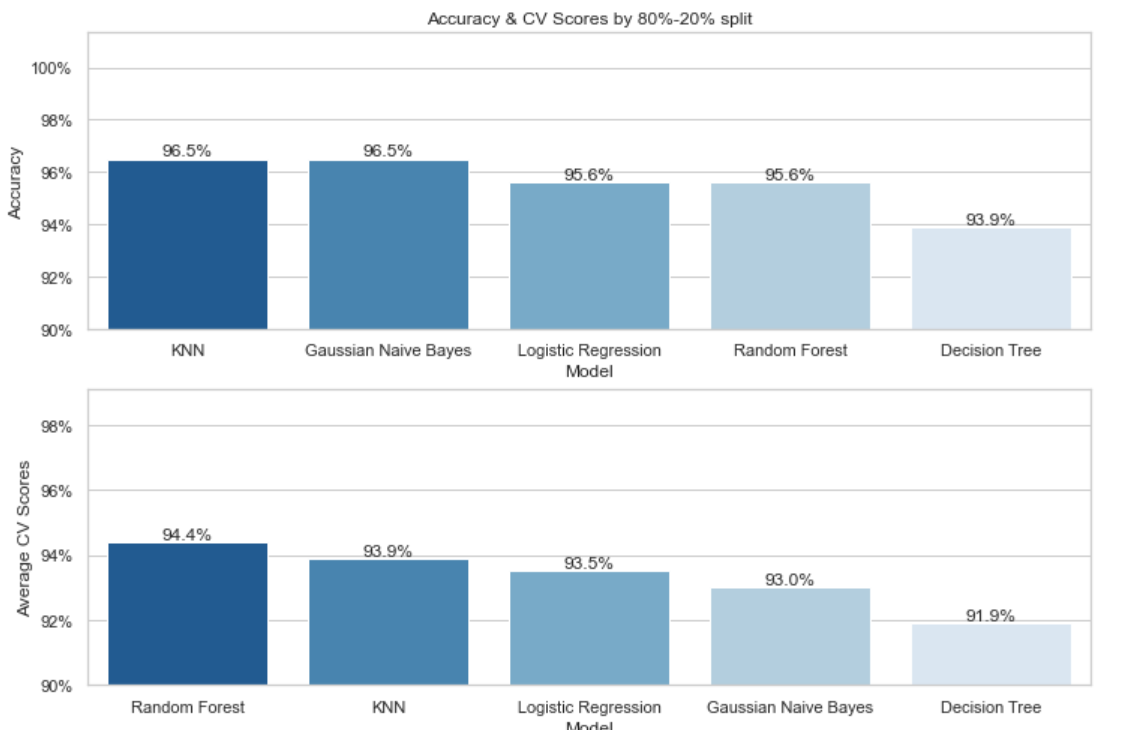
With 70%-30% split, KNN model has the highest accuracy 96.5% and Random Forest has the highest cross validation score 94.6%.

### 60%-40% Split



With 60%-40% split, KNN model has the highest accuracy 95.2% and Random Forest has the highest cross validation score 94.4%.

### 80%-20% Split



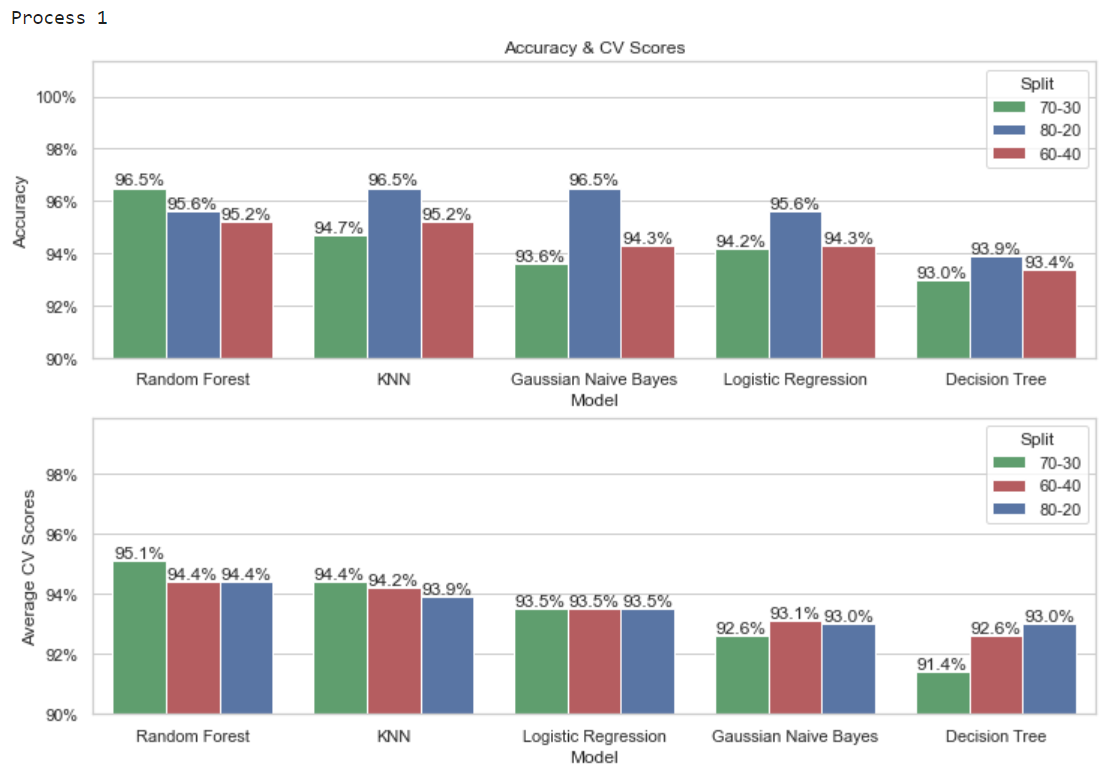
With 80%-20% split, KNN model has the highest accuracy 96.5% and Random Forest has the highest cross validation score 94.4%.

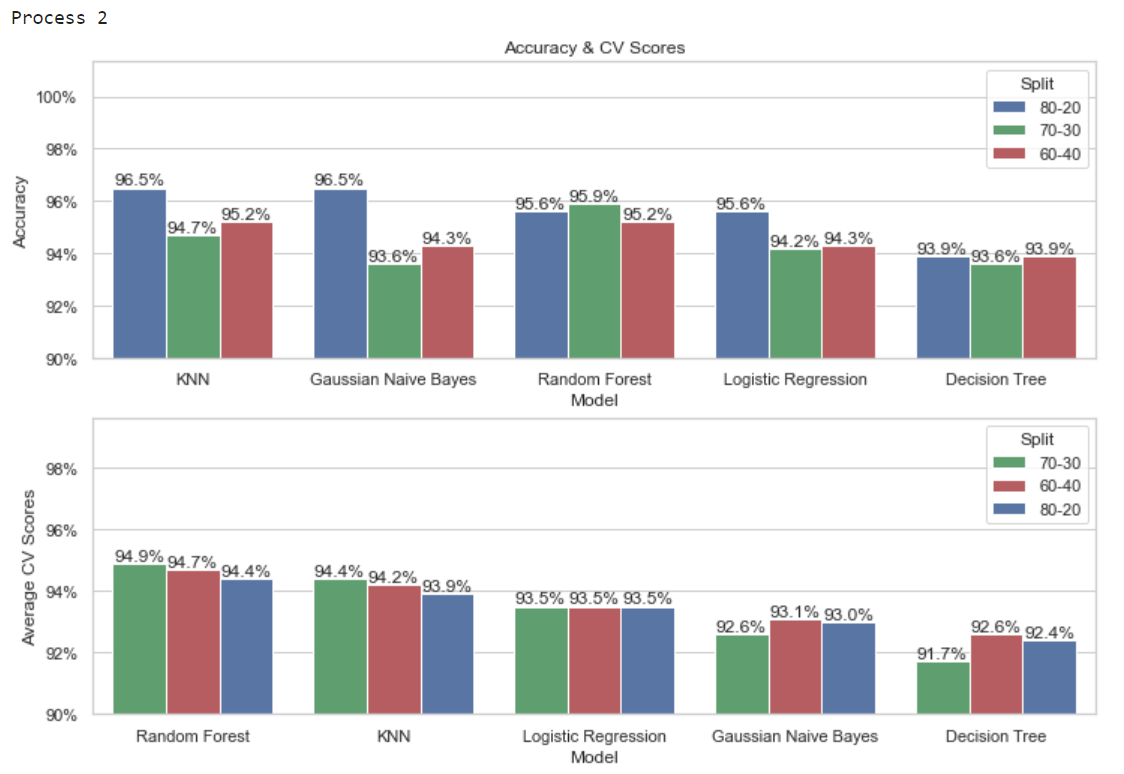
### Combination

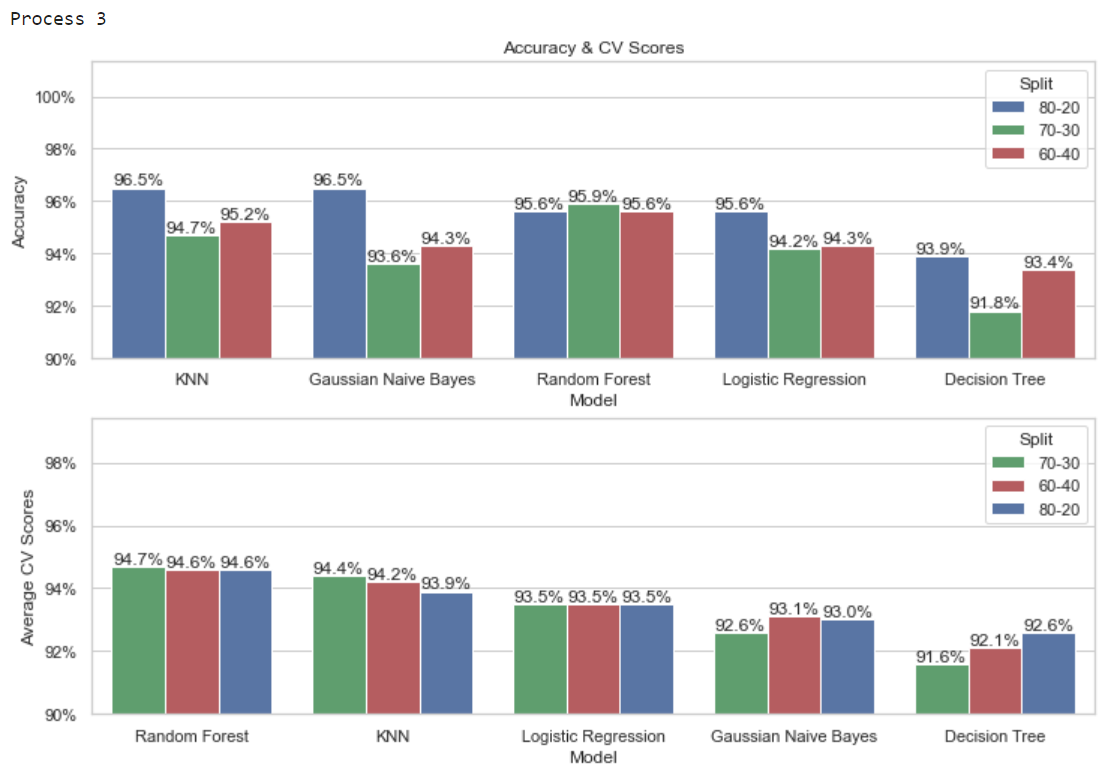


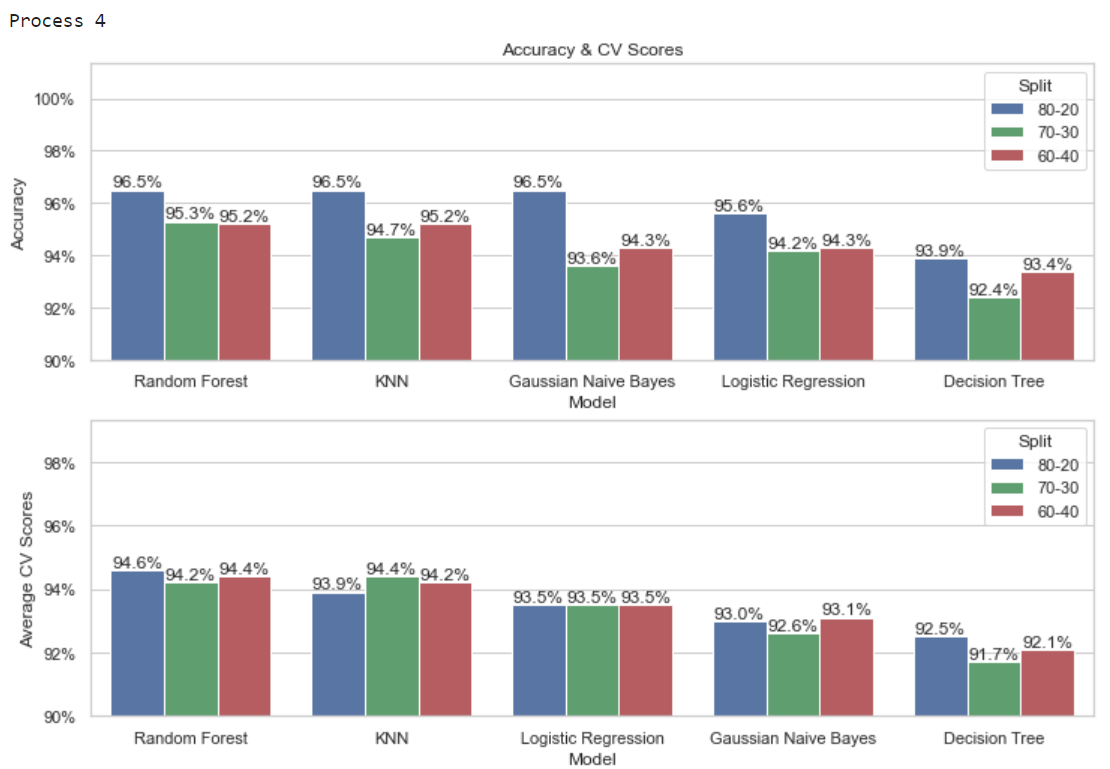
## Repeat the process

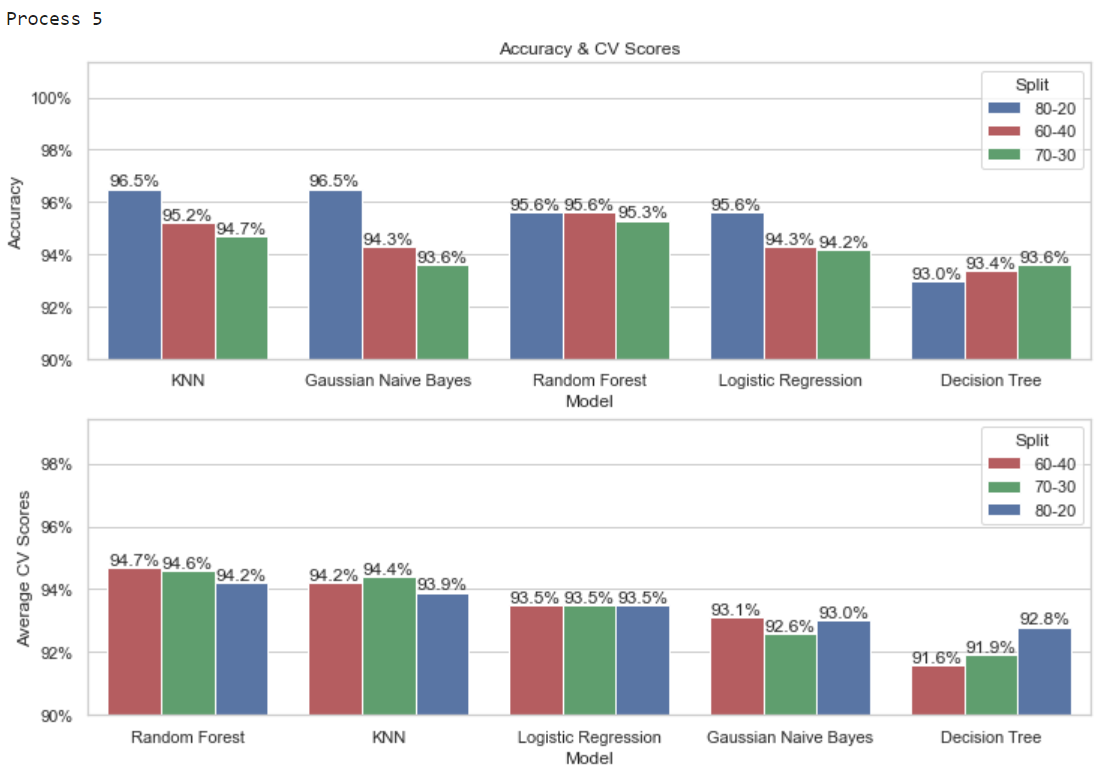
Repeat the process from split the dataset to evaluation for 5 times. 5 combination charts were got:











## Models comparison

By the comparison, no matter the ratio of split, KNN model has the highest accuracy for 3 times and Random Forest has it for 2 times. While Random Forest has the highest cross validation score for all the times

Cross validation score is the average accuracy of all the folders. It trains models with all the data compared to a single accuracy which trains only a part of data. Therefore, in this case, cross validation score is more fair and more accurate.

Above all, the conclusion can be made:

**The Random Forest model (100 trees by default) is the best choice to predict a patient malignant or benign.**

# The shortcomings of the work and concluding remarks on the continuity of the work

## Imbalanced

The class of the dataset is a little imbalanced with 357: 212 or 63%: 37%. But it was not solved in this or past researches.

There are some techniques like Resampling to handle this. Considering the data is in medical science field, it is inappropriate to simply apply Resampling to generate random data for the features. For example, area is calculated from radius. It is impossible to exist a large area with tiny radius. This technique needs to be further adjusted before applying.

## Skewed data

It is obviously by EDA report that only “smoothness\_mean”, “smoothness\_worst”, “symmetry\_mean” and “texture\_worst” features are closed to normal.

Others are either skewed right or skewed left. This may affect some models like Logistic Regression. But it was not solved in this or past researches.

The easiest way to solve it is log transformation. It can be applied before scaling step.

However, skewed data does not affect KNN and Random Forest algorism. It needs further investigation on it to see whether it is necessary to apply the transformation.

# Reference

Ahmet Saygılıa (2018). Classification and Diagnostic Prediction of Breast Cancers via Different Classifiers. ISVOS Journal 2018, 2(2):48-56

Wisconsin Diagnostic Breast Cancer (WDBC) Data Set. (n.d.). Retrieved February 07, 2023, from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology, 1915, 861-870.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3), 389-422.

Rokach, L., & Maimon, O. (2010). Data mining with decision trees: theory and applications. World Scientific.

Sung, H., Kim, J., Lee, H., & Kim, Y. (2015). Prediction of breast cancer subtype based on gene expression data using random forests. Journal of medical systems, 39(2), 313-319.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.