GROUP 1 - CAP 5610

**HEART DISEASE PREDICTION**

FINAL PROJECT REPORT - SPRING 2021horizontal line

# Authors

Lamia Alshahrani

Anwesh Praharaj

Dharitri Rathod

Syed Muhammad Sabih

horizontal line

# Introduction

This Final Project Report is for CAP 5610 (Spring 2021) final submission. This report consists of:

* The Project Statement and the Problem
* Description and Analysis of the Data
* Modeling Approach
* Investigation of Modeling and Results
* Conclusive Solutions and Future Work

To address the aforementioned deliverables for our Final Report, we decided to work on a project in the field of healthcare and collected the data based on patient attributes that classified the presence of heart disease. Our main goal pertained to the accurate classification/prediction of the presence of heart disease in patients. Further in the report, we will see the repository from which the data was collected, description of the data, the approach to use the data to solve the problem, and the results produced with various testing and modeling approaches.

# Table of Contents

[**Authors**](#_arolcxe0i15c) **1**

[**Introduction**](#_5o2z9y11w895) **1**

[**Table of Contents**](#_m8pwmrtouiei) **2**

[**Project Statement and Motivation**](#_7wp28lddszx4) **4**

[Project Statement](#_gaww7gbfo7m) 4

[Motivation](#_da4uro3t2t4o) 4

[**Heart Disease Dataset**](#_5e4bjtqolf9j) **4**

[The Dataset](#_8tofpfab1wu6) 4

[Exploratory Data Analysis of the Dataset](#_sin2utp9luu2) 5

[Loading the datasets and checking for Missing Values](#_o6wsvpyz8kej) 5

[Basic Data Cleaning](#_n7v5oxat9ajj) 5

[Imputation to replace missing values](#_j25rw0veixnr) 6

[The procedure for imputation of missing values is as follows:](#_qtq2z7h1cp1r) 6

[The results of the imputation](#_cbbnv0ji7qm3) 7

[Analysis of the Imputed Dataset (Final Dataset)](#_ludfvkdu8m61) 8

[Visualizing the Final Dataset](#_2qos8ybt054x) 9

[Histograms of the attributes](#_q7xtfr99v8sc) 9

[Density Curves for the attributes](#_om97zo1swp4f) 11

[Box Plots for the attributes](#_hmd1yb407d8x) 13

[Scatterplot for Multimodal Analysis](#_or6861xj53wx) 14

[Correlation Matrix for Multimodal Analysis](#_e68qew5n2s0z) 15

[Interpreting the Visualizations](#_9wavps34mkka) 15

[**Building the Baseline Models**](#_61ec35qospxd) **16**

[Procedure](#_an69t1zgb484) 16

[Baseline Models Performance Conclusion](#_tguc18umfsu4) 16

[**Addressing the Questions from Visualization**](#_bj894llhau9r) **17**

[Feature (Predictor) Selection](#_wv9vd9rupo84) 17

[New Findings after visualizing the Final Dataset](#_a85xp4pilnh) 17

[**Baseline Models - New Imputed Dataset**](#_xuv2gpmunfj4) **19**

[Procedure](#_4pcvtvjh1bt1) 19

[Box Plots for model performance](#_p1bg3qezdk69) 19

[New Baseline Models Performance Conclusion](#_disy58lalnkt) 19

[**Hyperparameter Tuning for the Selected Models**](#_f0emdgu8pgdd) **20**

[Model 1](#_4ux7yq7jmyr5) 20

[Model 2](#_offx8xitaswu) 20

[Model 3](#_wu4p73e01wna) 20

[Model 4](#_s7e9pli9i7ud) 20

[Model 5](#_cpobkcw4dmgh) 20

[Model 6](#_lqiecqd8o5eh) 20

[Model 7](#_d55fk1eld5w5) 20

[Model 8](#_yjb2kfv3k7t7) 20

[Model 9](#_2vfdwa3mkkk2) 20

[**Best Performing Models after Hyperparameter Tuning**](#_x0nacpc1p4s0) **20**

[**Building an Ensemble from the Best Performing Models**](#_92r3p61hp5sh) **21**

[Using Voting Classifier to produce the Mean Accuracy](#_vy92wur6myjs) 21

[**Project Trajectory, Results and Interpretation**](#_53aygx87de3g) **21**

[Changes in the Project](#_29bdhswasg4v) 21

[Analysis and Results of the 5 best performing Models](#_4bftfe873lut) 21

[Measuring the success of the Project](#_uyzwmgfgtuyb) 21

[Interpretation of the Results](#_8l3viuhrdlhx) 21

[**Final Conclusion and Future Work**](#_ateifdu5tf8d) **21**

# 

# 

# 

# 

# 

# 

# 

# Project Statement and Motivation

## Project Statement

For the Heart Disease Dataset, we want to classify the presence of heart disease using the best combinations of the dataset files (Cleveland, Hungarian, Switzerland, VA) based on the most relevant patient attributes.

## Motivation

We decided to choose a dataset in the field of healthcare because medical problems are serious significant problems in the society and the prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. Furthermore, more people die due to heart disease everyday, so we found a raw database on the UCI Repository that we could investigate and in turn change it into information that could help make informed decisions and predictions in the future.

# Heart Disease Dataset

## The Dataset

The dataset used for this project was collected from the database located in the [UCI Repository](https://archive.ics.uci.edu/ml/datasets/Heart+Disease?spm=5176.100239.blogcont54260.8.TRNGoO). The database contained 76 attributes, but all published experiments referred to using a subset of 14 of them. In particular, the Cleveland database was the only one that had been used by ML researchers to this date. The "goal" field referred to the presence of heart disease in the patient. It was integer valued from 0 (no presence) to 4. Experiments with the Cleveland database had concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). The names and social security numbers of the patients were recently removed from the database and replaced with dummy values. The input variables for all the databases used were: *age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca*, and *thal* while the output variable used was *num*.

Despite the Cleveland database only being used for research to this date, we decided to test different combinations of databases rather than only using the Cleveland database, along with imputing the missing values to see if we can achieve better results.

## 

## 

## Exploratory Data Analysis of the Dataset

### Loading the datasets and checking for Missing Values

After loading the dataset files for Cleveland, Hungarian, Switzerland, and VA, the exploration of the data showed that the Cleveland dataset had 303 rows and 14 columns, the Hungarian dataset had 294 rows and 14 columns, the Switzerland dataset had 123 rows and 14 columns, and the VA dataset had 200 rows and 14 columns.

The datasets were further investigated for missing values and the following missing values were found:

* *Cleveland: 6 missing values*
* *Switzerland: 273 missing values*
* *Hungarian: 782 missing values*
* *VA: 698 missing values*

Finally, the descriptive statistics for each feature in each dataset were produced to identify type errors or zero rows. The results of the descriptive statistics are shown in the accompanying Jupyter Notebook for this Milestone of the project.

### 

### Basic Data Cleaning

Once the missing values were checked, the next step was to look for redundancy within the datasets. First, we checked for redundant columns/features in the datasets and as a result of the investigation, we found that no column had only one value and the columns with more than one values were categorical variables. Second, we checked for duplicate rows in the datasets and as a result of this investigation, we found that the duplicate values might exist due to the missing values.

### 

### 

### 

### 

### Imputation to replace missing values

At this point, we decided to impute the missing values before we could proceed with further exploration and investigation of the datasets. As mentioned before, our team wanted to see if we can use different combinations of datasets rather than only using the Cleveland dataset to achieve better results.

### The procedure for imputation of missing values is as follows:

The 5 combinations of datasets that we decided to use were:

* *Cleveland, Hungarian, Switzerland, and VA*
* *Cleveland, Hungarian*
* *Cleveland, Switzerland*
* *Cleveland, Hungarian, and Switzerland*
* *Cleveland (For comparison with other combinations)*

The Imputation methods we decided to use on the aforementioned combinations were:

* *KNN Imputer - Odd k values ranging 1 through 50*
* *Iterative Imputer - Order = ascending, descending, roman, arabic, and random*

Other changes made to the data for imputation:

* *The inputs were normalized using MinMax Scaler*
* *The Target variable was changed to binary (0: Absence of heart disease, 1: Presence of heart disease)*

Evaluation methods to see which combination has a better *accuracy* score:

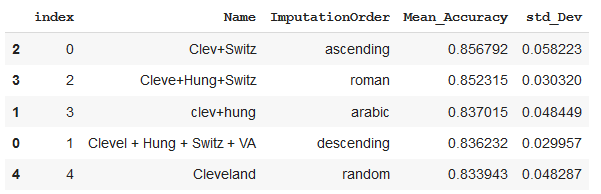
* *Model used: RandomForestClassifier - default*
* *Cross Validation: RepeatedStratifiedKFold - 10 folds, repeated 3 times, with a random state of 1*

### The results of the imputation

* KNN



* Iterative



After the imputation of the missing values using RandomForestClassifier and RepeatedStratifiedKFold to measure which combination would be better for classification, we decided that Imputation with KNN (K=3) produced the highest mean accuracy and the lowest standard deviation for the combination of Cleveland, Switzerland, and Hungarian datasets.

We decided to use the imputer method and combination of datasets mentioned above to produce our final dataset for this Project.

### Analysis of the Imputed Dataset (Final Dataset)

The Final dataset produced after the imputation on the combination of Cleveland, Switzerland, and Hungarian datasets had 720 rows and 14 columns. Upon checking for missing values on the Final dataset, no missing values were found, which meant all missing values were replaced as a result of the imputation. Due to the missing values that existed before imputation, the reduction of redundancy on the raw datasets did not make much sense. After the imputation, the Final Dataset was checked for redundancy and as a result, one duplicate row which existed was removed. Furthermore, the type of variables for the Final Dataset were investigated and findings showed that the input variables were scaled while the target variable remained binary. Additionally, we decided to check that the class distribution is balanced as there are 359 observations in class 0 and 360 observations in class 1.

After double-checking for missing values and exploring the Final Dataset, the descriptive statistics were produced for the Final Dataset and the results are shown in the accompanying Jupyter Notebook for this Milestone of the project. Lastly, the correlation between the attributes and skewness of the attributes was produced. At this point, we decided that we needed visualizations for the descriptive statistics, correlation between the attributes, distribution of the dataset, and skewness of the attributes in order to produce more meaningful statistical inferences for the Final Dataset.

### 

### 

### 

### 

### 

### 

### 

### 

### 

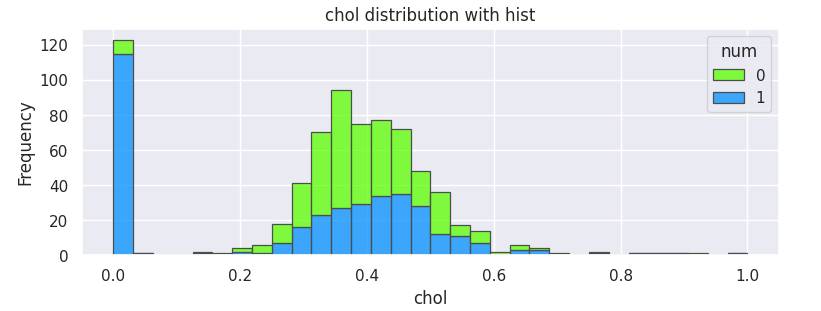
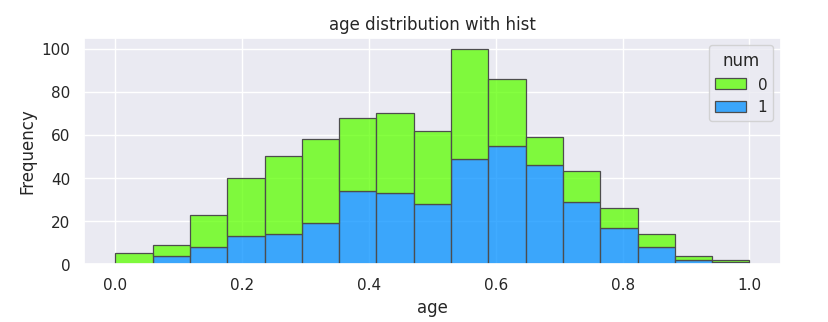
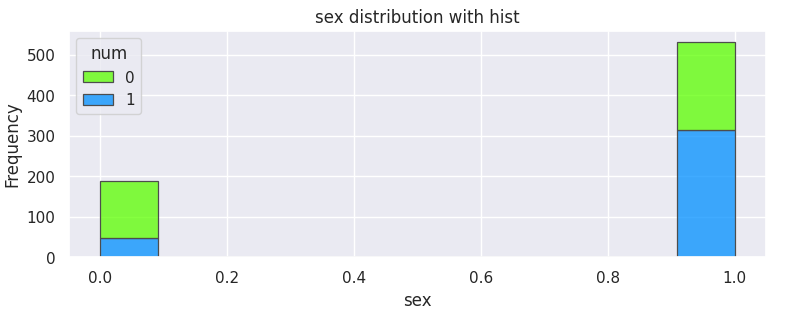
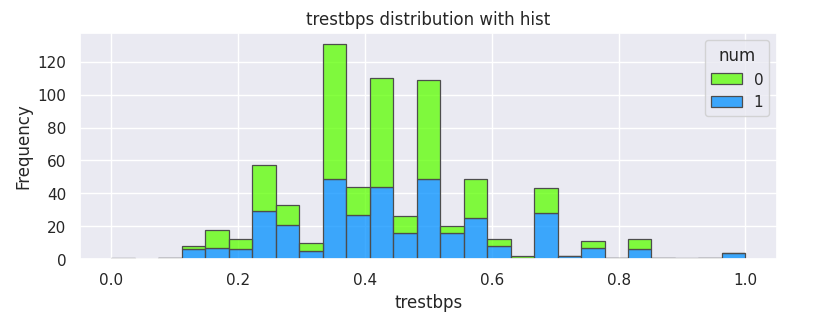
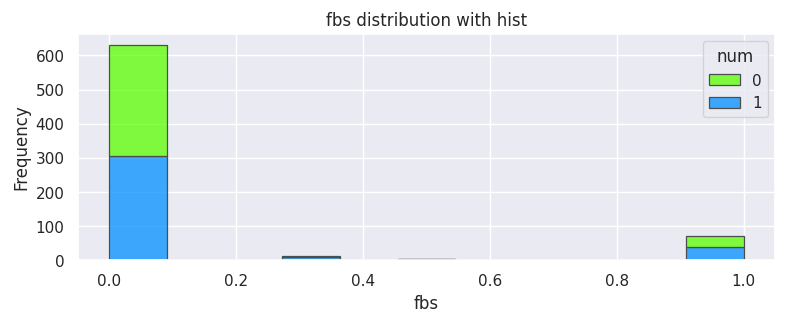
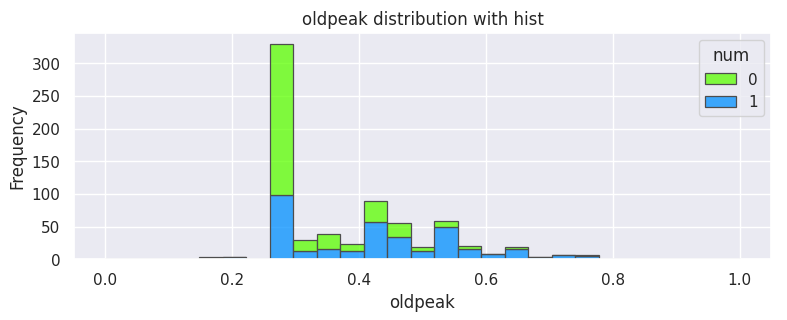
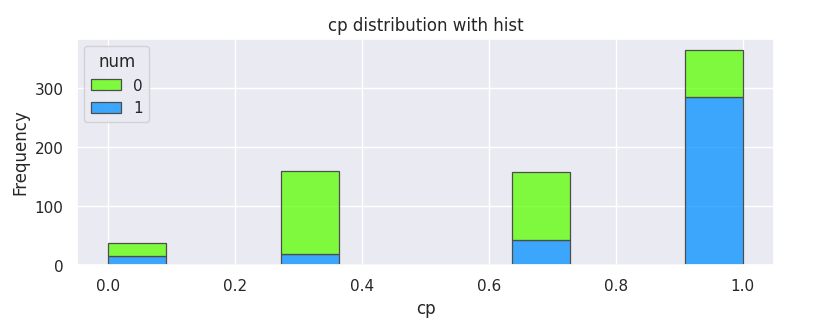
### Visualizing the Final Dataset

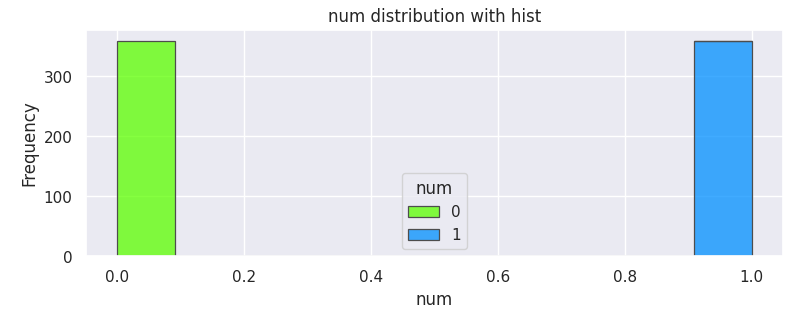
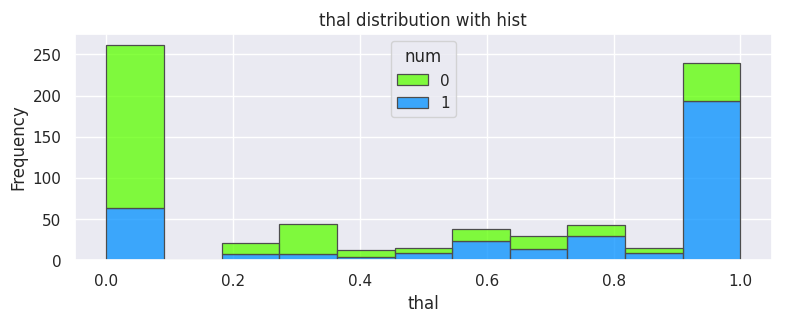
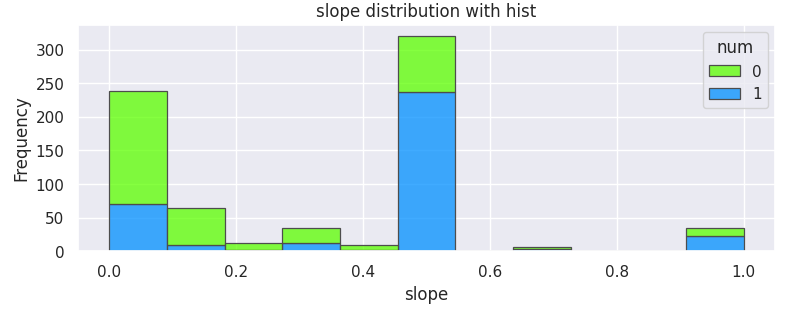
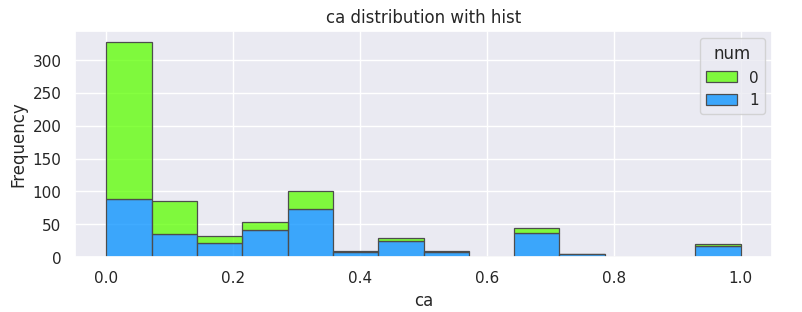
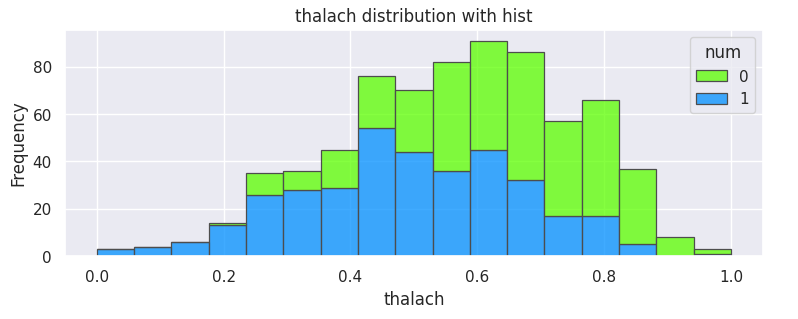
For the visualization of the Final Dataset and its attributes, we decided to produce:

* Unimodal Visualization: *Histogram, Density Curve, and Box Plot*
* Multimodal Visualization: *Scatter Plot and Correlation Matrix*

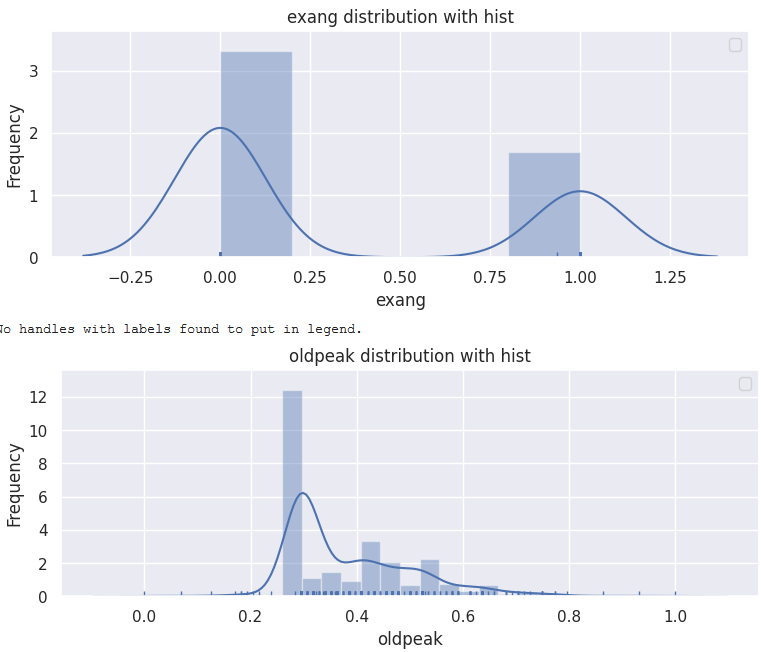
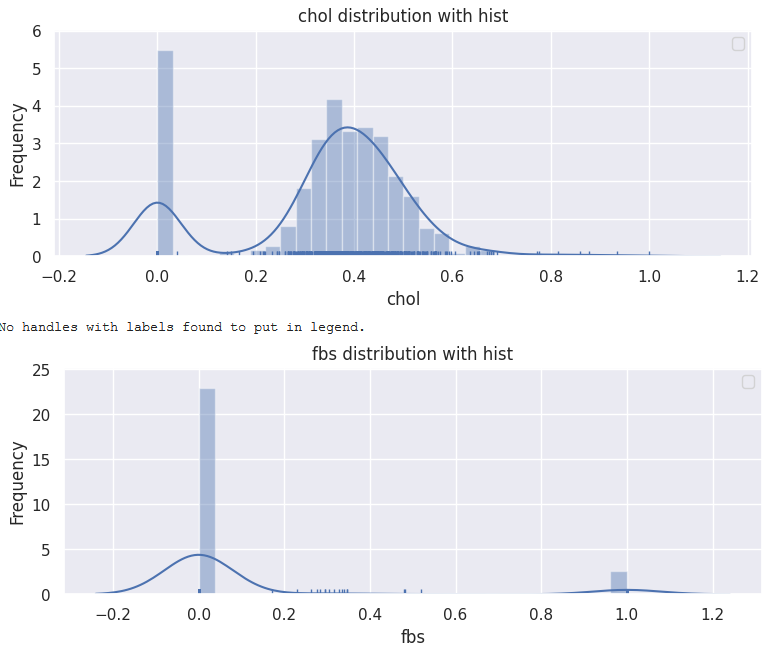
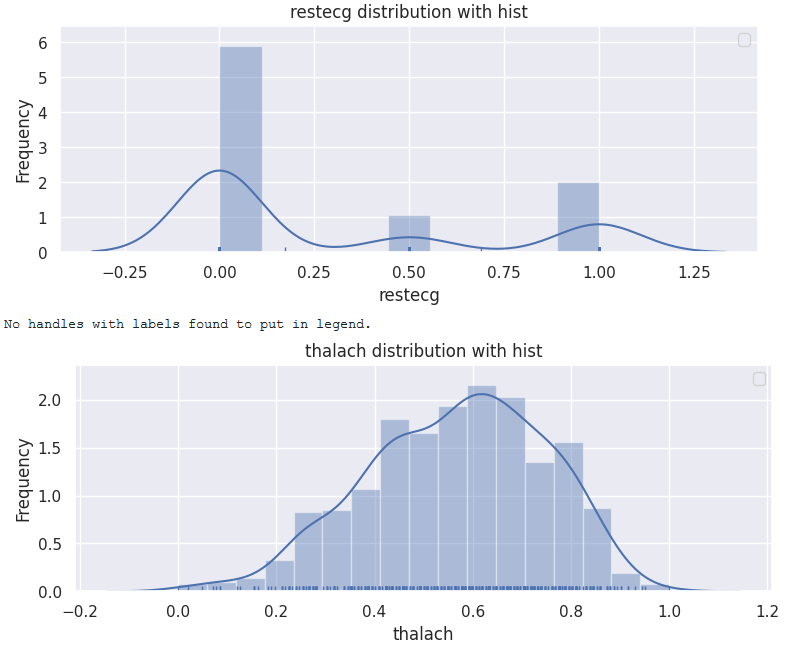
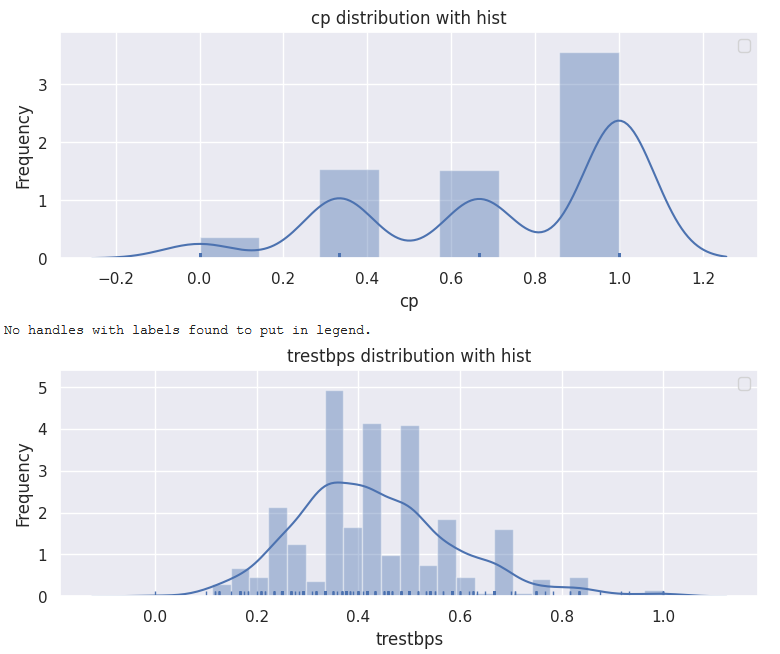
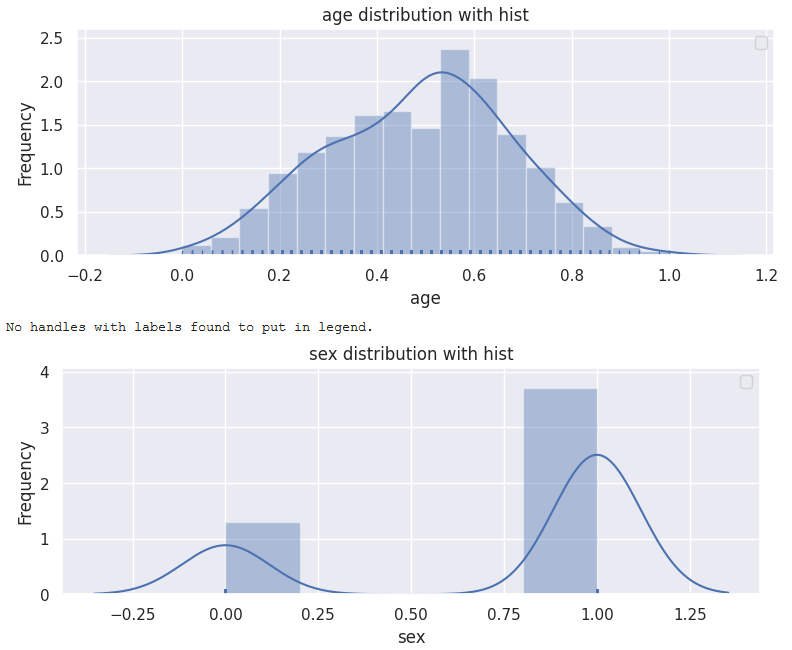
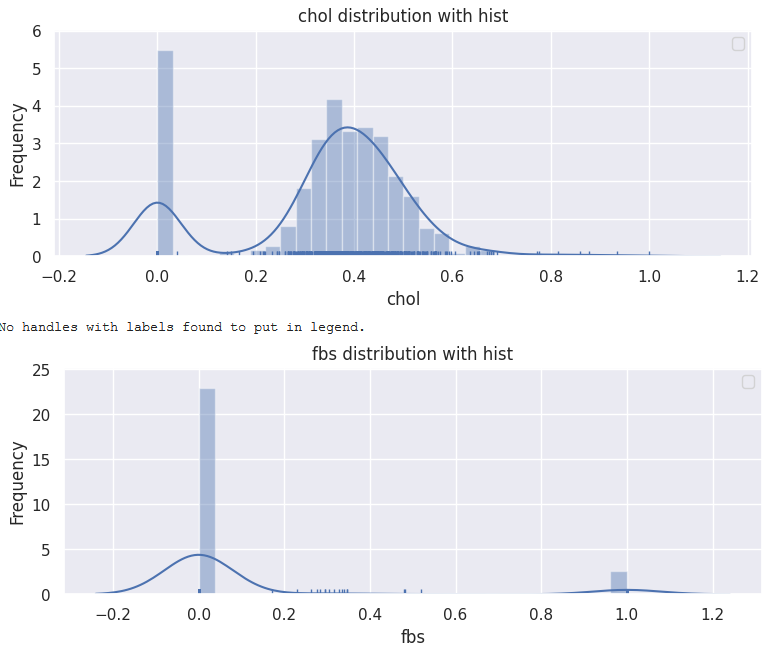
#### 

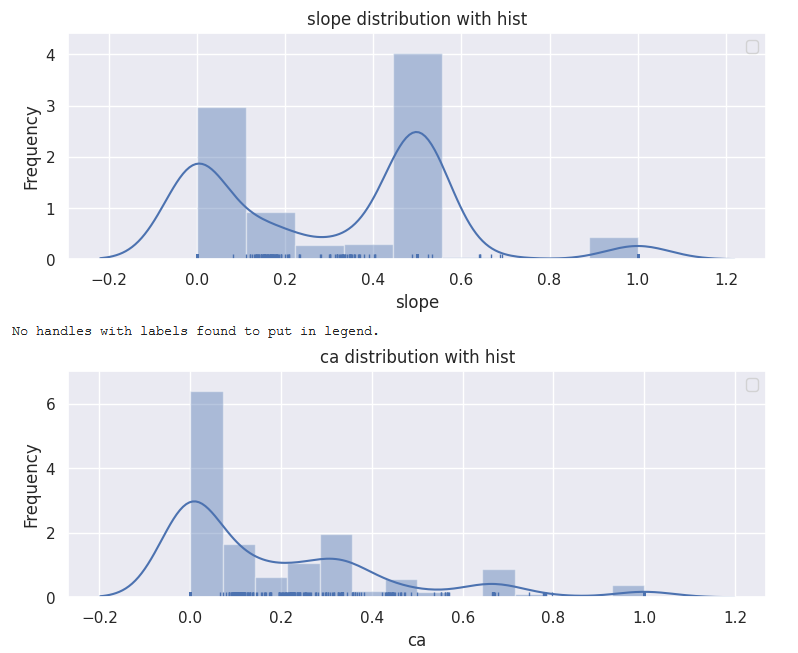
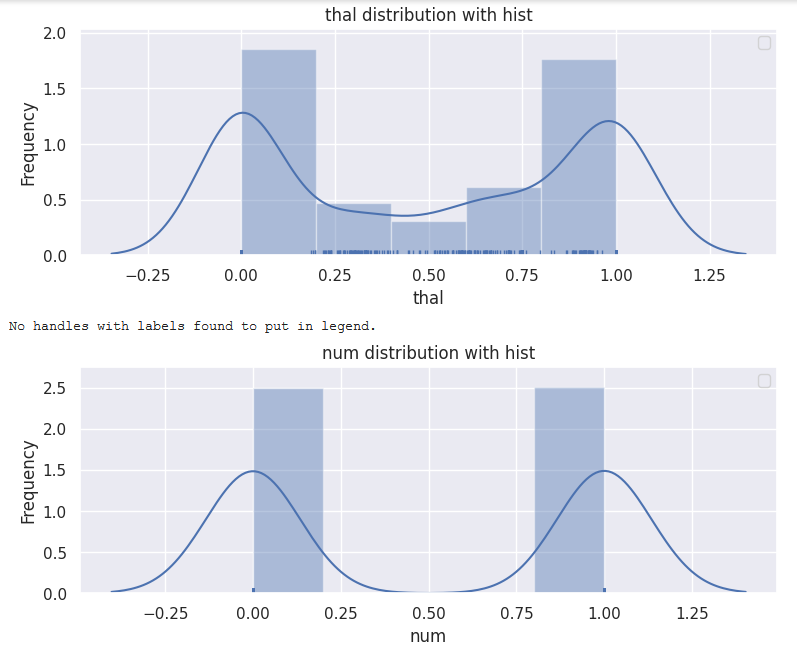
#### Histograms of the attributes



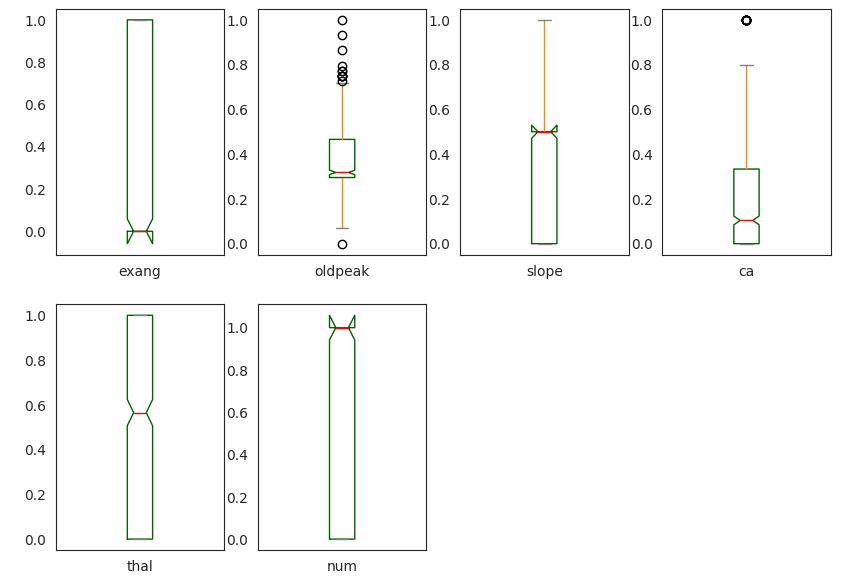
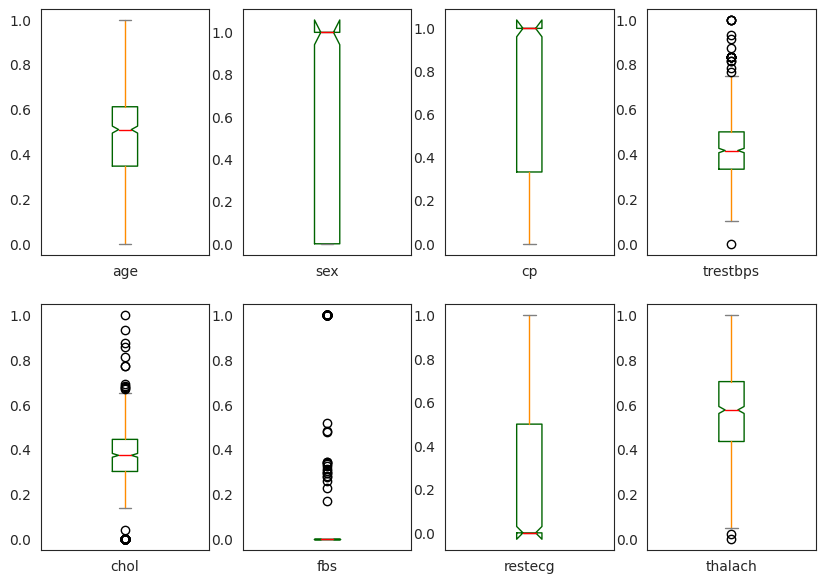


#### Density Curves for the attributes





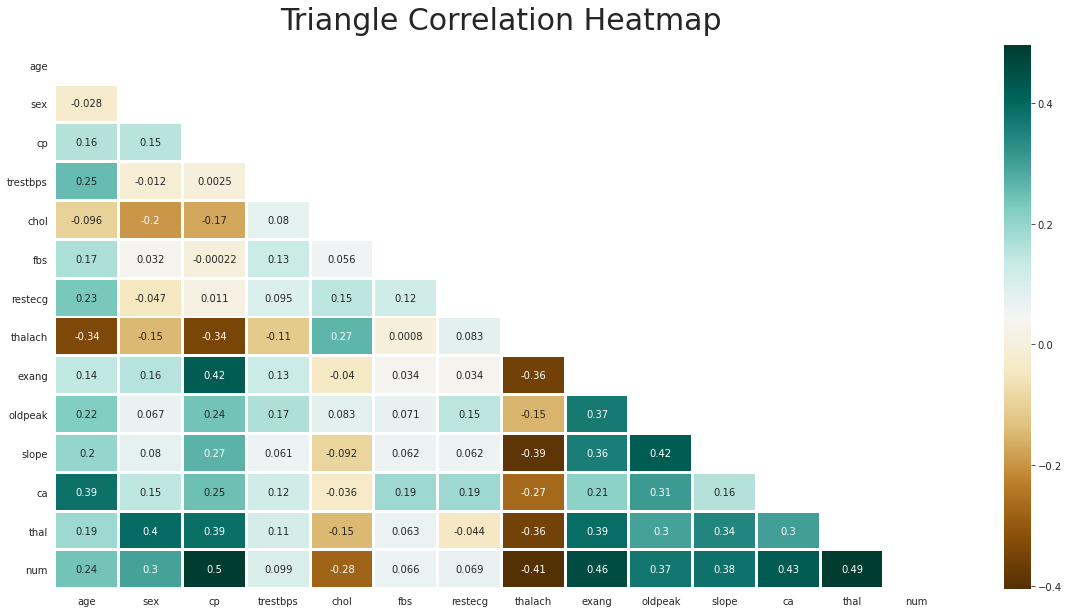
#### Box Plots for the attributes



#### Scatterplot for Multimodal Analysis



#### Correlation Matrix for Multimodal Analysis



### Interpreting the Visualizations

After investigating the visualizations, the findings from the histograms and density curves stated that some of the quantitative attributes are skewed. These findings suggested that standardizing the data might produce better results. Moreover, the investigation of the boxplots indicated that some of the variables have outliers which might affect the results of some of the algorithms. Although, we decided to not remove the outliers since the data is concerned with the medical field and the outliers could work in our favor in producing accurate results. However, results produced with and without the outliers would give us the final verdict. Furthermore, upon investigating the Scatter-plot and the correlation matrix, the findings suggested strong correlation between the output variable, *num*, and some input variables such as *thal, ca, slope, oldpeak, exang, thalach, cp, and sex,* which indicated that these predictors could produce more accurate results.  
After analyzing the data through producing statistics and visualizations, we produced a few questions that need to be investigated for the dataset:

* Would fewer predictors rather than all the predictors in classifying the presence of heart disease produce a more accurate result?
* If yes, which predictors are the best?
* Are the predictors controllable?
* Were there any new findings after visualizing the data?

# Building the Baseline Models

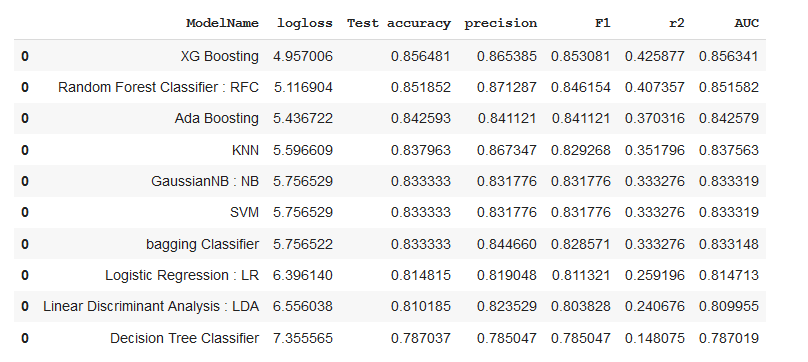
## Procedure

At this point, we decided to use all the predictors to build our Baseline Model. Before building a Baseline Mode, we decided to split the data into two parts: *70% training* and *30% testing*. After that, we considered 10 classification algorithms for model selection*: Logistic Regression, KNN Classifier, Support Vector Machine, Random Forest Classifier, Ada Boosting, Decision Tree Classifier, Bagging Classifier, XG Boosting, Linear Discriminant Analysis, and Naive Bayes*.

A spot-check along with the ROC curve and confusion matrix for the algorithms was performed to measure the most effective model performances that produced the best results. Furthermore, we have plotted the variable importance for the tree classifiers to understand the splitting in the data.

During the spot-check for algorithms, performances of the algorithms were measured using metrics such as: prediction accuracy, log loss, precision, F1- score, R-Square, and AUC.

After running the 10 models, the results were as follows:



## Baseline Models Performance Conclusion

Based on the findings above, the results indicated that *XG Boosting, RandomForest, Ada Boosting, KNN Classifier, and SVM* were performing in the most effective manner and producing the best results.

# Addressing the Questions from Visualization

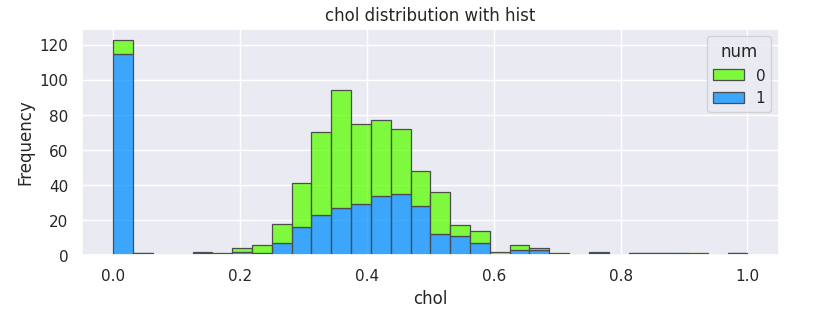
* Would fewer predictors rather than all the predictors in classifying the presence of heart disease produce a more accurate result?
* If yes, which predictors are the best?
* Are the predictors controllable?
* Were there any new findings after visualizing the Final Dataset?

## Feature (Predictor) Selection

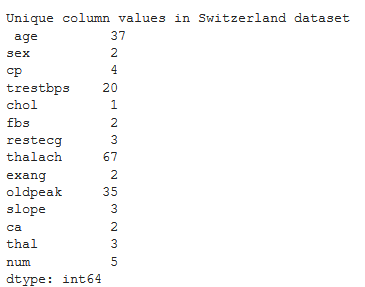
For the most effective results, depending on the algorithm used, the number of features would vary. For example, feature A might be useful for model A, but might not be useful for model B. To tackle this problem, we planned to use Principal Component Analysis(PCA) and SelectKBest depending on the algorithm’s requirement to select the best features which produced the most effective model performance. Depending on the model, different techniques of feature selection should be used.

## New Findings after visualizing the Final Dataset

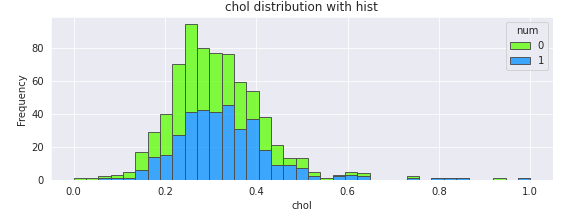
After visualizing the histogram for the feature *chol,* we noticed an abundance of 0 values and the rest of the data was normal.



Hence, we investigated the statistics for unique values where we had previously checked for redundancy.



The findings from the investigation proved that the missing values for *chol* were recorded as ‘0’ for the Switzerland dataset. Therefore, to solve this problem, we replaced the ‘0’ values with NaN and imputed the missing values. Following the [imputation procedure](#_qtq2z7h1cp1r) mentioned earlier in this report, we imputed the missing values using KNN Imputer (K=47) on the same combination of datasets (Cleveland, Switzerland, and Hungarian) as it produced the best accuracy and standard deviation. A new Final Imputed Dataset was produced for building the models. The histogram for the feature *chol* after imputing the ‘0’ values in the Switzerland is as follows:



The feature *chol* above after the imputation of ‘0’ values seems to follow a normal distribution.

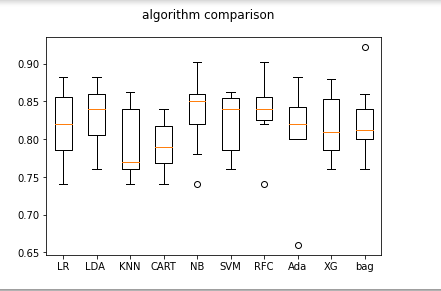
# Baseline Models - New Imputed Dataset

## Procedure

At this point, we decided to build the Baseline models on the New Imputed Dataset using the [procedure](#_an69t1zgb484) used previously. This was necessary so could compare the model performances to the previous Baseline models using the same metrics used before (prediction accuracy, log loss, precision, F1- score, R-Square, and AUC). The results are as follows:

## Box Plots for model performance

The box plots for the prediction accuracy for the New Baseline Models is as follows:



## New Baseline Models Performance Conclusion

Based on the findings above, the results indicated that *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*were performing in the most effective manner and producing the best results.

# 

# 

# 

# 

# 

# 

# Hyperparameter Tuning for the Selected Models

## Model 1 LR

Logistic regression is one of the statistical models , mainly used in modeling a binary dependent variable.

**Rescaling:** For LR we had applied StandardScaler()to standardize the data inside the pipeline .

**Feature selection:** In the feature selection we had used the PCA .Also, we had tried the RFE but it did not improve the accuracy .

**Hyper parameter tuning:** In the LR we had used the default hyper parameters pca\_\_n\_components': [5, 15, 30, 45, 64], 'logistic\_\_C':(-4, 4, 4) and run the gridsearch to look for the best accuracy we got : 0.8240740740740741 using Best parameter (CV score=0.837), 'pca\_\_n\_components': 5 .

**Final Model1:**

Finally, we had fit and used the best parameters we had fit and predicted the model in order to print the performance metrics to evaluate the model.

## 

## 

## Model 2 KNN

Basemodel:

Rescaling:

Feature selection:

Hyper parameter tuning:

Final Model1:

## Model 3 NB

Basemodel:

Rescaling:

Feature selection:

Hyper parameter tuning:

Final Model1:

## Model 4 LDA: LDA is a linear model for the multi-class classification. problem, the base accuracy of LDA was 0.8379 on the training dataset.

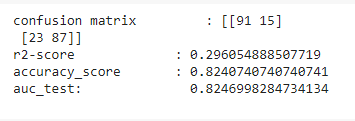
## Rescale: We have applied the different rescaling methods to improve the results such as StandardScalar(), Normalizer() in the pipeline and outside the pipeline, but didn’t see any improvement in the accuracy. For further analysis, we will continue to Standardize data inside the pipeline to avoid leakage of the data.

**Feature selection:**  We have tried SelectKBest(), RFE, and PCA for the feature selection, but we got the best accuracy of 0.840392 with PCA(n\_component=6) in the pipeline.



**Hyperparameter tuning:** most important hyperparameters are *solver=*  ['svd', 'lsqr', 'eigen'], *Shrinkage* =[0,1], and n\_components[1,13]. As PCA gave us n\_components=6,we have run the grid search to find the best solver=’lsqr’ and shrinkage=0.0 with accuracy 0.835.



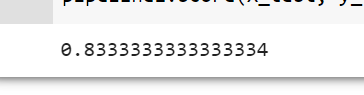
We will fit and predict the dataset with n\_component=6, which is 

## Model 5 SVM

A support vector machine (SVM) is a supervised machine learning model that mainly uses in the classification problems

**Rescaling**: In SVM we had applied the StandardScalar() inside the pipeline , in order to improve our accuracy .However, it did not change or raise that.

**Feature selection:** In this model, we had tried RFE,PCA,and RepeatedSelectKBest() for the feature selection . The best accuracy we got assigning the (n\_features\_to\_select=8)



**Hyper parameter tuning:** The most important hyperparameters in SVM are C and the Kernel and gamma . In the REF we had assigned the default value of SVM (kernel="linear", C=1). Using the grid search we had to add gamma to search for the best accuracy.Also, we had assigned random\_state=7 to generate the same result.

**Final Model1:** Finally , using the best parameters we had fit and predicted the model in order to print the performance metrics to evaluate the model.

## Model 6 Decision Tree

Basemodel:

Rescaling:

Feature selection:

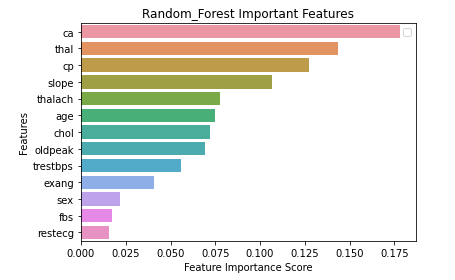
Hyper parameter tuning:

Final Model1:

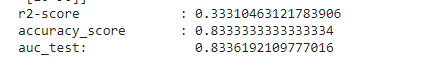
## Model 7 Random Forest: It is one of the ensemble models. Hence, there is no need to rescale the data. Also, it generates its own feature of importance while building a tree. Base accuracy is 0.8379



Hyper parameter tuning: Random Forest has mainly 3 features to tune namely: n\_estimator, max\_features, and max\_depth. Also, to generate the same result, we have used random\_state=7. To select the best parameters using grid search we have n\_estimator= 1000 (number of trees), max\_features=’sqrt’ ( take only sqrt(p) features in consideration for the split of the node), and max\_depth =12 with training accuracy of 0.85533 and predicted accuracy of 0.8333.



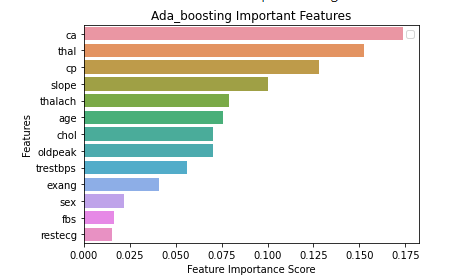




## Model 8 Adaboosting: it is another type of ensemble, but it is a boosting algorithm. We will use Adaboosting on top of the previous Random Forest algorithm to improve its accuracy by passing it as a parameter in Adaboosting. Base accuracy is of Ada boosting algorithm is : 0.828



As it is a tree based classifier, we don’t need to rescale or do feature selection. It has mainly 3 hyper parameters: base\_estimator (other algorithm), n\_estimators (number of trees), and learning rate. Using hyperparameter tuning on the base\_estimator =[RandomForestClassifier(),LogisticRegression(),DecisionTreeClassifier(max\_depth=8)], n\_estimators= [10, 50, 100, 500],and learning\_rate=[0.0001, 0.001, 0.01, 0.1, 1.0], the best\_estimator is RandomForestclassifier(),n\_estimator=50, and learning\_rate=0.0001.



Using best parameter, we fit the final model and predicted accuracy is: 

## Model 9 XG boosting

Basemodel:

Rescaling:

Feature selection:

Hyper parameter tuning:

Final Model1:

# Best Performing Models after Hyperparameter Tuning

Best 5 models were:

# 

# 

# 

# Building an Ensemble from the Best Performing Models

## Using Voting Classifier to produce the Mean Accuracy

At this point, we decided to use the 5 best performing Models and combined them using the Voting Classifier to build an ensemble. The ensemble would help us see if combining these algorithms produced an increase in mean prediction accuracy. The results are as follows:

# Project Trajectory, Results and Interpretation

## Changes in the Project

## Analysis and Results of the 5 best performing Models

## Measuring the success of the Project

Did we achieve our goals?

## Interpretation of the Results

What do the results mean? What impact does it have? Is there any take-away message? Were there any surprises?

# Final Conclusion and Future Work

Conclusion \_\_\_\_\_\_\_\_\_\_

What could we do to improve? What could we have done differently?

Strengths and shortcomings and how could we overcome these if we had more time?

Perhaps have a more populated dataset and less missing values.

Perhaps build a Neural Network and try Deep Learning.

Perform Hyperparameter tuning using BayesSearchCV from the skopt library.

To conclude ,