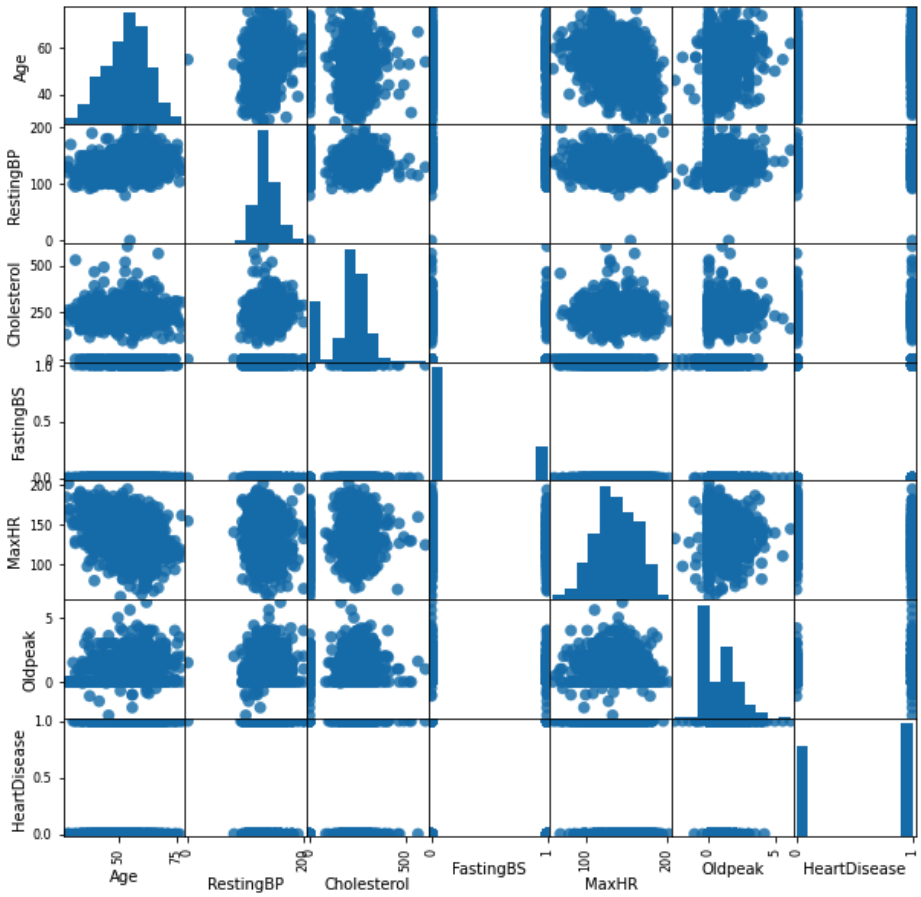
Project 3 Report

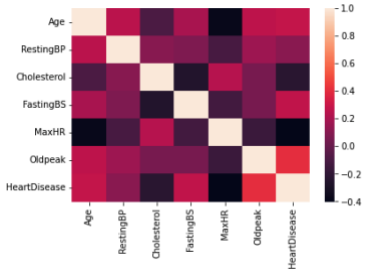
Description: In this report, I learned to use a combination of Pair Plot, Correlation Matrix, and Heatmap are created to examine the highly correlated variables at the preliminary stage for data exploration. Then, I preprocessed the data set by using Principal Component Analysis as the dimensionality reduction technique to select the features that carry the most information for training and testing the machine learning models. Later in section II, I discussed most problems addressed in the assignment prompt (Project 3: Task 3 and 4) including demonstrating classification models, applying hyperparameter tuning to improve 4 of the 6 models selected through parameter adjustment, iteration numbers and other criteria, section also includes calculating model evaluation statistics like variance, bias, f1 score, precision, recall, with the help of confusion matrix.

I. Pre-processing, Data Mining, and Visualization

The Pair Plot has shown an inverse relationship between MaxHR and Age and a slightly positive relationship between RestingBP and Cholesterol. In the next section, I will use correlation matrix and heatmap to demonstrate the correlation coefficients for different variables.

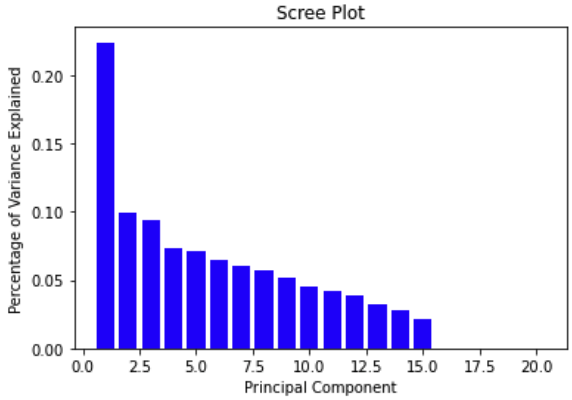


In the Correlation Matrix and Heatmap the strongest positive correlation is found between heart disease and old peak with a correlation coefficient of 0.403. Then the strongest negative correlation is found between heart disease and max heart rate with a correlation coefficient of -0.400421. Further, no correlations are found in variable pairs like Age & Cholesterol, Resting BP & Fasting BS, Resting BP & heart disease with correlation values close to zero. For efficiency purpose, the variables with correlation values close to zero could be put aside. This will lead us with insights to the feature selection in the next step.

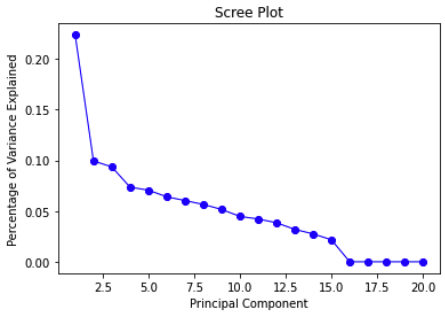


	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	0.282039
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	0.107589
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	-0.232741
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	0.267291
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	-0.400421
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	0.403951
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	1.000000

After applying PCA() to the data set, I plotted two scree plots to eliminate 8 features that carry the least information. From the bar chart, it shows that the last 5 features are not shown in the scree plot since they're close to zero thus have no significant signals as principal components. The features that could be excluded are 'ST_Slope' and 'ExerciseAngina'. Please note, the 5 features are corresponding to 2 features in the original data set prior to the encoding process.



Further reducing the dimensionality to keep the variance percentage to approximately 85% since the last 8 principal components only account for approximately 5% of the total variances that carry the least important information, excluded features are therefore RestingECG_LVH, RestingECG_Normal, RestingECG_ST, ST_Slope_Down, ST_Slope_Flat, ST_Slope_Up, ExerciseAngina_N and ExerciseAngina_Y. Lastly, to eliminate Oldpeak which accounts for only 5.65% of the total variance.




```
array([0.22373241, 0.32316039, 0.41672064, 0.49027509, 0.56089759,
       0.62498389, 0.68532719, 0.74187698, 0.7934767 , 0.83821301,
       0.88039294, 0.91887263, 0.95082033, 0.97831834, 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ])
```

The final set of features includes Age, RestingBP, Cholesterol, FastingBS, MaxHR, Sex_F, Sex_M, ChestPainType_ASY, ChestPainType_ATA, ChestPainType_NAP, and ChestPainType_TA. In statistical context, through the preprocessing stage there are 11 features out of 20 features carry important information in the data or carry significant signals. Lastly in the preprocessing stage, polynomial feature transformation derives new features that could further improve model performance is used.

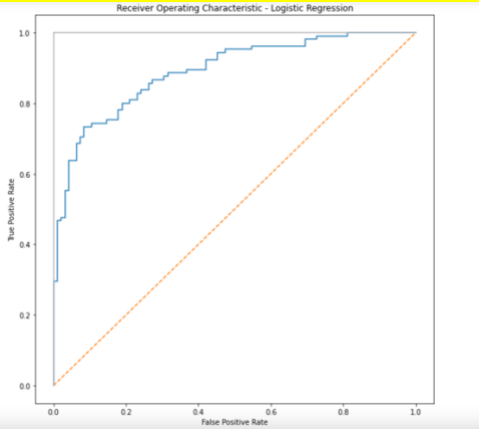
II. Classification Models + Iterations + Evaluations

Questions to explore: why the particular model, why this metrics or what the dataset suggest?

In this section, I will discuss 4 of the 6 ML models used to solve the classification problems of heart disease data set to answer the above questions, the six models are Logistic Regression, Support Vector Machine, Decision Tree, Naïve Bayes, Random Forest, and K Nearest Neighbor models. Then, adjustments of the model parameters will be explained by using GridSearchCV.

1. Logistic Regression Model

As the target class in heart disease data set represents a patient has a heart disease or has not a heart disease in binary results 0 and 1, logistic regression that categorizes results into 0 and 1 groups based on threshold that compares probability of a patient has a heart disease becomes the first choice to use as a classification model creating an S shaped graph dividing two classes. Lastly, I will use hyperparameter tuning to improve the model performance. Preprocessing includes both PCA and polynomial features techniques. As a result, GridSearchCV selected c=0.01, penalty:L2 (Ridge), and solver=liblinear as the best parameters.



2. Support Vector Machine

Based on the logic behind logistic regression model discussed above, support vector machine is a similar model that uses probability to make classification. As SVM maximizes the distance between all the instances thus, serves as another choice of classification model to solve the heart disease problem. Preprocessing includes both PCA and polynomial features techniques. From using the default SVM model parameters to the best parameters that improve the performance selected by GridSearchCV, the following are chosen C=1, gamma=0.01, kernel=rbf (radial basis function kernel).

3. Decision Tree

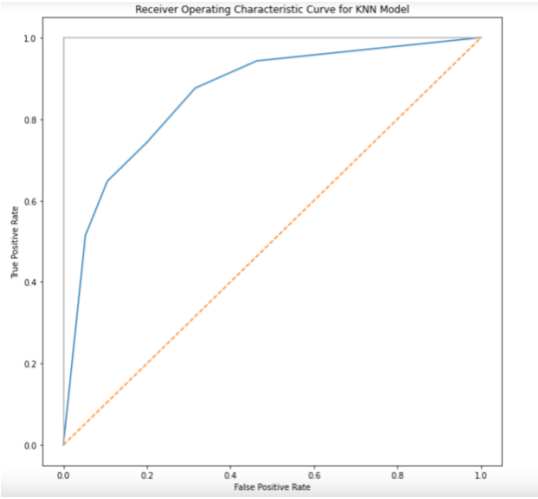
The third model I chose is decision tree due to minimum effort needed for preprocessing the data set. Building a tree based upon features ordered by importance (calculating gini impurity) allows each branch to perform its best and pass only the instances qualified to the next branch to classify and eventually generates classification results does not require much effort for preprocessing. Thus, instead of using any scaling techniques data set will be the same as the original data set however with PCA selected features (Age to ChestPainType_TA). Hyperparameter tuning significantly improved the performance from 0.705 accuracy to 0.788, it is done through changing the parameters of Criterion=gini, max_depth=10, min_samples_leaf=10.

4. Random Forest

I chose to try random forest because it takes similar tree-based approach shared by decision tree model. Even though there's a risk of overfitting, I'd like to use random forest to randomly form the trees and using voting mechanism to find the best fit rather than purposely for a tree based on entropy or gini impurity. Comparing to decision tree, the accuracy score remains the same when applying random forest while f1 scores improved by 5%.

5. Naïve Bayes

6. K Nearest Neighbor



III. Evaluation for the Logistic Regression

A. Among all the six classifiers used upon un-scaled, pca-processed, and polynomial-featured data sets, logistic regression model is the best performing classifier which generates 83% accuracy based on pca-processed but non-polynomial-featured data set.

B. I used accuracy score, precision, recall values to determine the best classifier, especially the accuracy score.

C. The bias in LR1=0.159, LR1_Variance: 0.02643 are considered low in both index, it means the logistic regression model is pretty good already at making classifications. However, we could reduce the risk of overfitting, therefore next step involves using data regularization. Here with the help of GridSearchCV, ridge regression is selected, other parameter changes are mentioned in section II describing more in details.

D. The best ROC operating point is when threshold is at 0.729

IV. Extensions

- Used 6 ML models
- Applied polynomial features for preprocessing
- Additional ROC curve on KNN Model is shown towards the bottom of the jupyter notebook.
- Reflection: I learned to apply PCA and polynomial features to preprocess the data set and apply multiple classifiers on these different data sets to compare the accuracy scores. In the future work, I would like to work on using precision and recall to give more sophisticated credit for determining the best models rather than use accuracy score only.

Acknowledgement

- **Grid Search with Logistic Regression:** https://www.kaggle.com/code/enespolat/grid-search-with-logistic-regression
- **Hyperparameter Tuning in Decision Trees:** https://www.kaggle.com/code/gauravduttakiit/hyperparameter-tuning-in-decision-trees
- **Optimal ROC Cutoff:** https://github.com/nicholaslaw/roc-optimal-cutoff
-