Machine Learning - CS 6140 (Spring 2023)

Project 3 – Applying Your Skills

Submitted by – Pulkit Saharan and Shivani Shrikant Naik

Abstract:

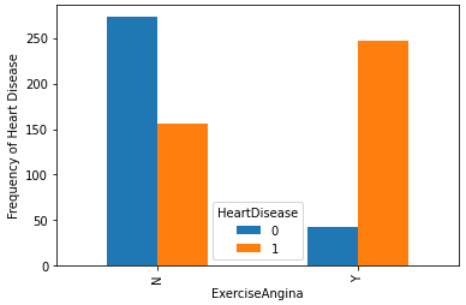
The objective of this project was to apply our acquired skills to a new dataset, specifically the Heart Disease dataset obtained from Kaggle. This binary classification dataset contains a combination of categorical and continuous features, providing us with the opportunity to explore diverse encoding and preprocessing methods. We conducted extensive exploratory data analysis (EDA) to gain an in-depth understanding of the data, followed by clustering and data mining to identify underlying patterns. Finally, we employed a range of machine learning techniques including logistic regression, decision tree, random forest, naïve bayes, adaboost etc. to successfully accomplish the classification task. We evaluated each model's performance using metrics such as accuracy, precision, recall, F1 score, AUC ROC, precision recall curve and compared their results to determine the most effective model.

1. Pre-processing, Data Mining, and Visualization

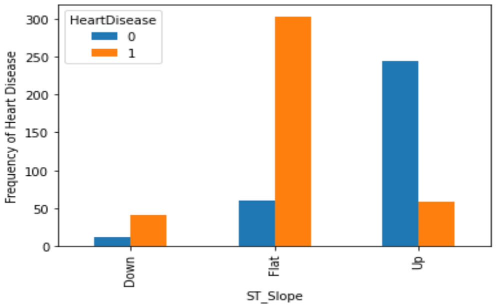
At the beginning of our analysis, we utilized various techniques such as visualizations, principal component analysis, clustering, and linear regression to investigate the relationship among variables and identify any trends or patterns present within the dataset.

1.1. Distribution of Categorical Variables: The bar graphs we generated provide insight into the distribution of categorical variables in relation to the presence or absence of heart disease. Upon examination, it is evident that some variables demonstrate a clear association with the presence of heart disease. For instance, when ExerciseAngina is marked as "Yes," the likelihood of having heart disease is notably higher, whereas the chances of having heart disease are lower when ExerciseAngina is "No".

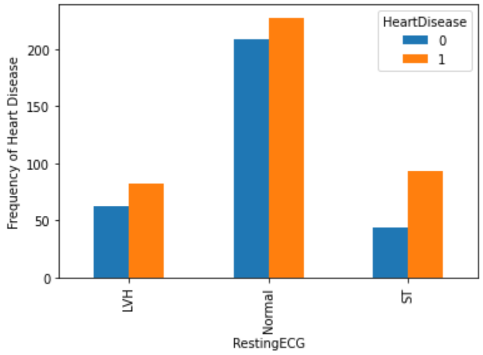
Distribution of Heart Disease for each value in ExerciseAngina



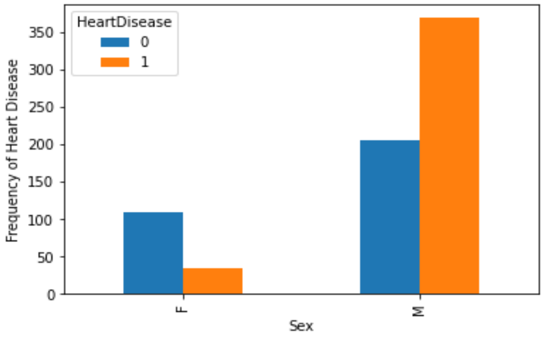
Distribution of Heart Disease for each value in ST_Slope



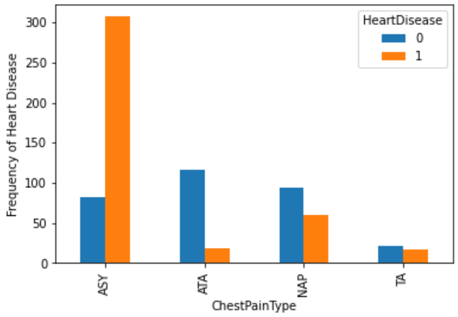
Distribution of Heart Disease for each value in RestingECG



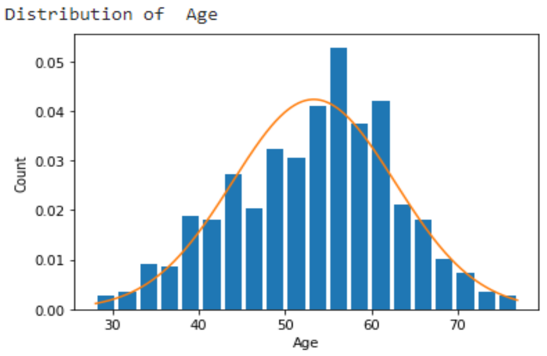
Distribution of Heart Disease for each value in Sex

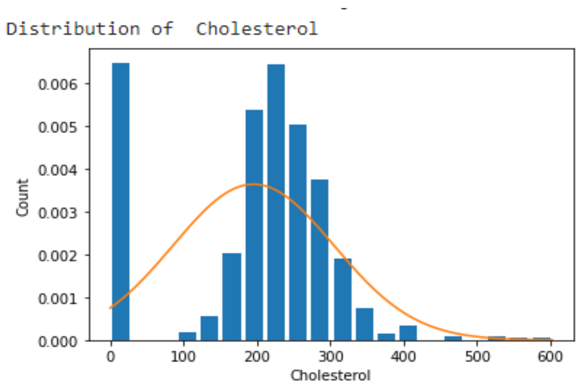


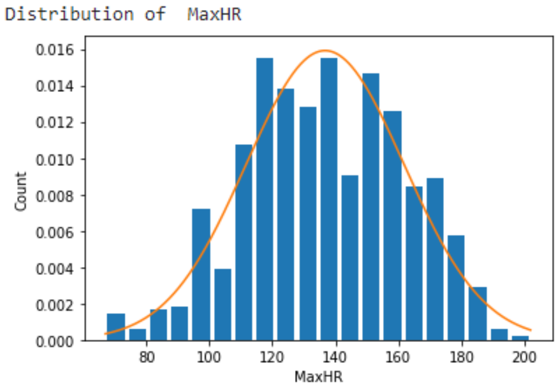
Distribution of Heart Disease for each value in ChestPainType

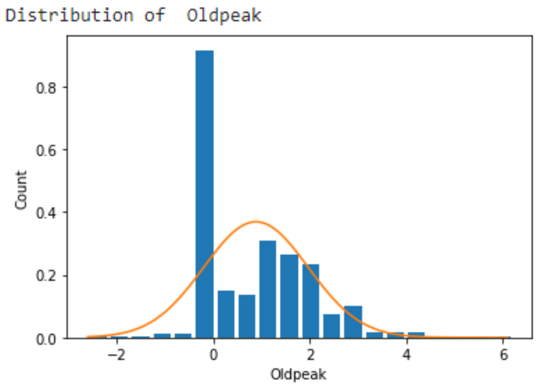


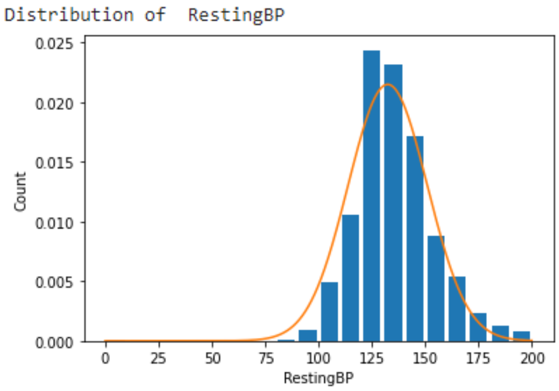
1.2. Distribution of Numeric Variables: By generating histograms of the numerical independent variables, we can observe the distribution of the data. The plots reveal that Age and Max Heart Rate follow a normal distribution. However, Cholesterol and RestingBP appear to be skewed, possibly due to the existence of outliers.











A. What variables do you plan to use as the input features?

In order to optimize our models, we employed various approaches to select the most effective combination of input variables. We experimented with several pre-processing techniques to identify the optimal approach that yielded the best performing model.

Our analysis incorporated all of the available independent variables in the dataset, including their polynomial transformations. By using this comprehensive approach, we sought to maximize the predictive power of our models and identify the most impactful features to explain the target variable.

Overall, our approach involved rigorous experimentation and testing to determine the most effective model configuration. By exploring various combinations of input variables and pre-processing techniques, we aimed to identify the model that would produce the most accurate predictions.

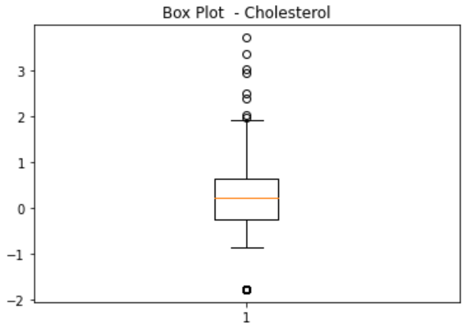
B. What pre-processing (if any) did you execute on the variables?

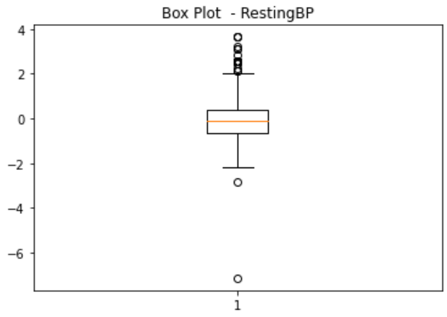
B.1. Scaling: The dataset we are analyzing is composed of heterogeneous variables, and it is necessary to standardize the data for some of the machine learning algorithms. This is done to prevent variables with larger units from having an unfair advantage over those with smaller units during modeling. To achieve this, we utilized the standard scaler from the sklearn library to scale the data.

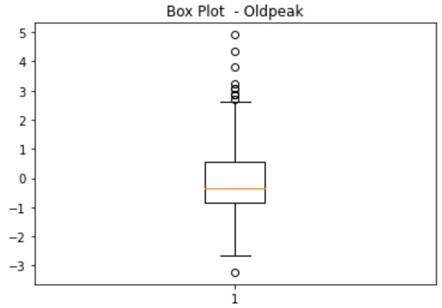
B.2. One-hot Encoding: Within the dataset, there are several categorical columns that contain two or more categories. In order to incorporate these variables into the model, we utilized a technique called one-hot encoding. This method involves creating dummy variables for each of the categories within the categorical columns.

B.3. Label Encoding: We also applied label encoding to transform the categorical variable from object to integer format. Label encoding is a technique used to convert categorical variables into numerical values, where each unique category is assigned a unique integer. By using label encoding, we were able to represent the categorical variables in a way that the model could better understand and use for predictions. In some cases, we found that the performance of these models improved significantly when we applied label encoding to the categorical variables. This is because label encoding allows the model to more accurately capture the relationships between the categorical variables and the target variable.

B.4. Outlier Detection and Treatment – We utilized box plots to identify any outliers within the dataset. The resulting graph revealed outliers in three columns: Cholesterol, RestingBP, and Oldpeak. To handle these outliers, we replaced them with NaN values and subsequently imputed the missing values using KNN imputation. We tried the model with and without outlier treatment.

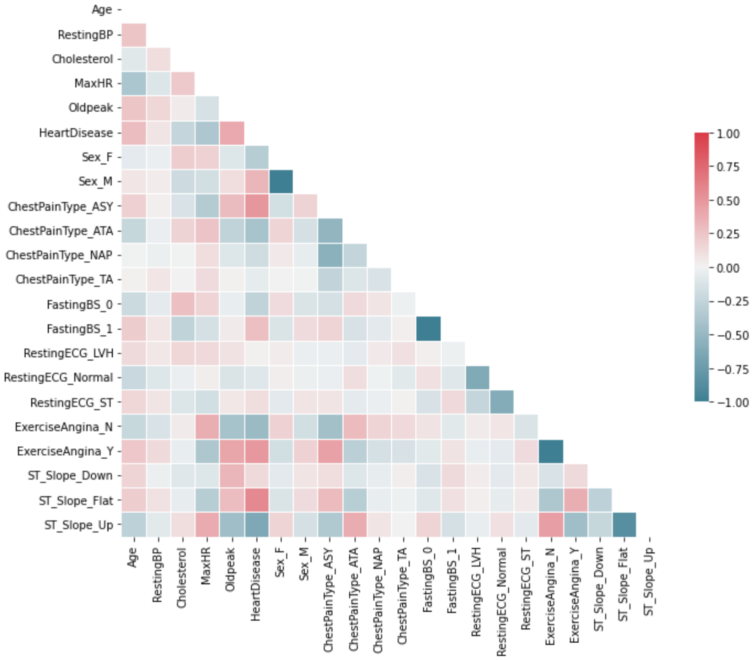




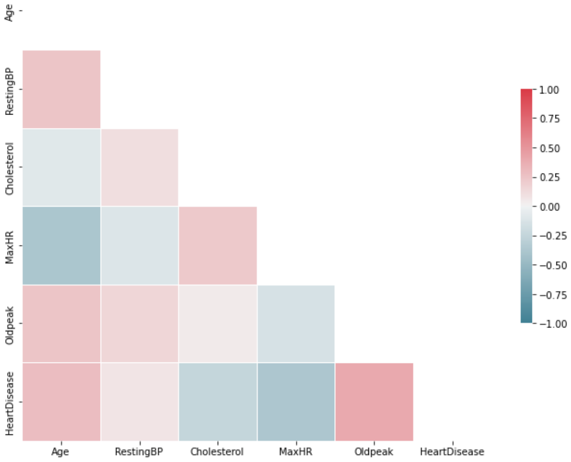


C. Which independent variables are strongly correlated (positively or negatively)?

Correlation among all variables: A correlation plot is an effective method to identify the correlation and direction of the relationship between multiple variables. The plot presented below indicates a strong correlation between Heart Disease and ExerciseAngina, ST_Slope and ChestPain, which is also evident from the distribution graphs above. Furthermore, there is a noticeable high collinearity among the dummy variables, which is consistent with theoretical expectations.

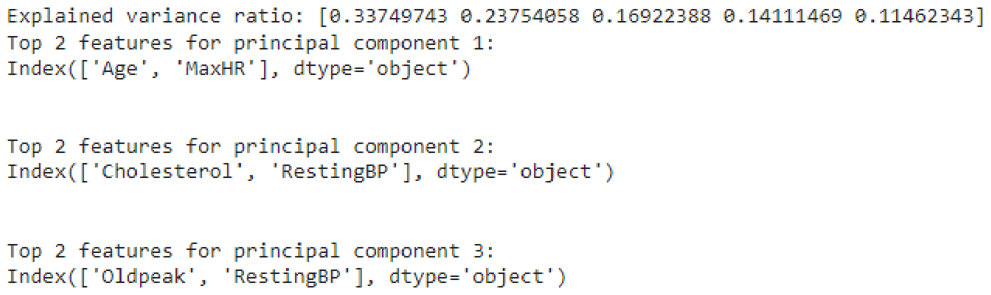


The plot displays the correlation among numeric variables, excluding dummy variables, and highlights that Age exhibits a strong negative correlation with Max Heart Rate and a positive correlation with Resting BP.



D. How many significant signals exist in the independent variables?

D.1. PCA: After conducting Principal Component Analysis on the scaled numeric independent variables in the dataset, we observed that the top three Eigenvalues out of five are able to account for roughly 75% of the variance present in the data. Our analysis further revealed that Age and Max Heart Rate are the crucial features for the first principal component, while Cholesterol and Resting BP contribute significantly to the second principal component. Finally, the third principal component is primarily driven by Oldpeak.

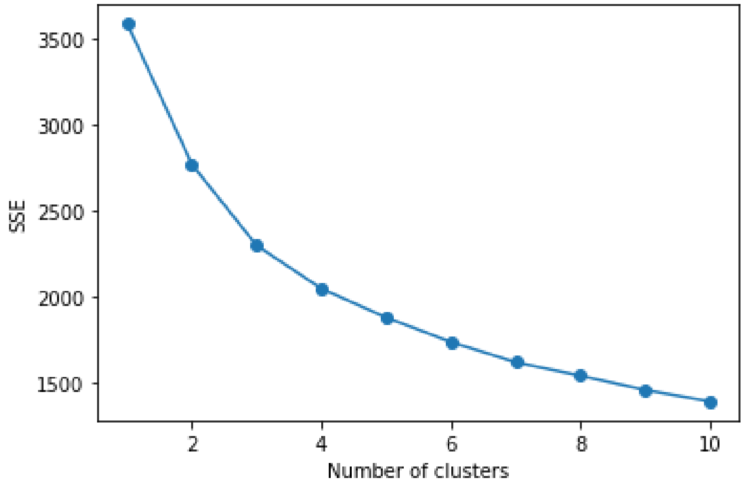


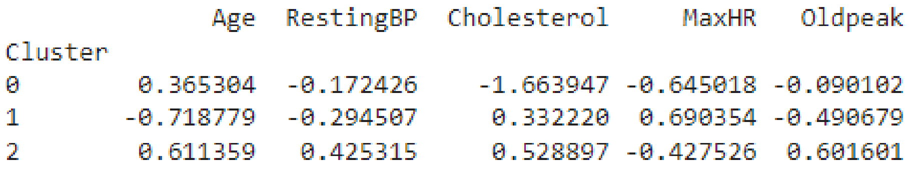
```
Explained variance ratio: [0.33749743 0.23754058 0.16922388 0.14111469 0.11462343]
Top 2 features for principal component 1:
Index(['Age', 'MaxHR'], dtype='object')


Top 2 features for principal component 2:
Index(['Cholesterol', 'RestingBP'], dtype='object')


Top 2 features for principal component 3:
Index(['Oldpeak', 'RestingBP'], dtype='object')
```

D.2. Clustering: After conducting K-means clustering analysis, we noticed a pattern that was consistent with the results of a principal component analysis (PCA) on the same dataset. Specifically, we observed distinct and consistent trends in the centroids of different variables within the same cluster. For example, the combination of Age and Max Heart Rate consistently showed a strong relationship with each other across all three clusters. Similarly, we observed similar trends in Cholesterol, Resting BP, and Oldpeak, suggesting that these variables are strongly associated with each other within the dataset.



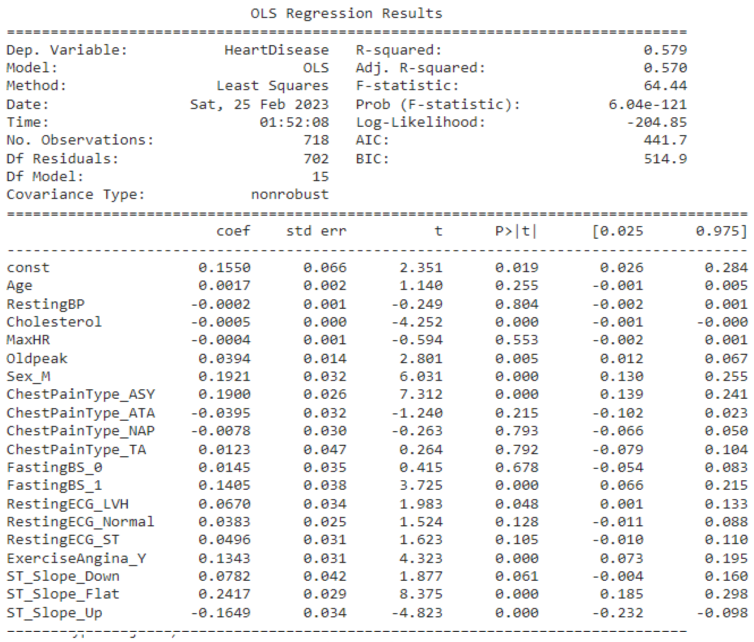


```
              Age  RestingBP  Cholesterol     MaxHR   Oldpeak
Cluster
0        0.365304  -0.172426    -1.663947 -0.645018 -0.090102
1       -0.718779  -0.294507     0.332220  0.690354 -0.490679
2        0.611359   0.425315     0.528897 -0.427526  0.601601
```

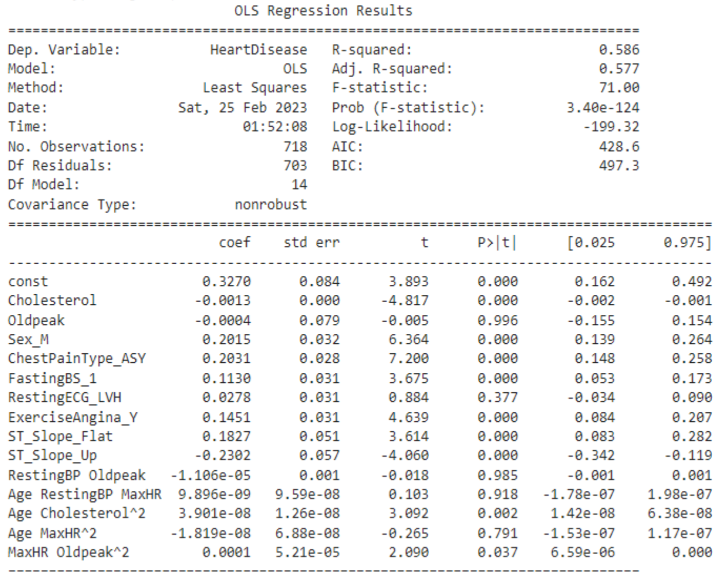
E. What derived or alternative features might be useful for analysis (e.g., polynomial features)?

We employed linear regression to identify significant variables, with a focus on polynomial features. During the initial baseline regression, we found several variables that had a significant impact on the dependent variable, including Cholesterol, Oldpeak, Sex_M, Chest Pain Type ASY, Exercise Angina Y, and ST_Slope.

To explore the relationship between these variables further, we created third-order polynomial features and examined their impact on the dependent variable. Our analysis revealed that the combination of Age and Cholesterol was a significant predictor of heart disease. Additionally, we found that Max Heart Rate, when used in conjunction with Oldpeak, was also a significant predictor of heart disease. By identifying these key predictors, we can better understand the factors that contribute to heart disease and potentially develop more effective prevention and treatment strategies.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:           HeartDisease   R-squared:                       0.579
Model:                            OLS   Adj. R-squared:                  0.570
Method:                 Least Squares   F-statistic:                     64.44
Date:                Sat, 25 Feb 2023   Prob (F-statistic):          6.04e-121
Time:                        01:52:08   Log-Likelihood:                -204.85
No. Observations:                 718   AIC:                             441.7
Df Residuals:                     702   BIC:                             514.9
Df Model:                          15
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 0.1550      0.066      2.351      0.019       0.026       0.284
Age                   0.0017      0.002      1.140      0.255      -0.001       0.005
RestingBP            -0.0002      0.001     -0.249      0.804      -0.002       0.001
Cholesterol          -0.0005      0.000     -4.252      0.000      -0.001      -0.000
MaxHR                -0.0004      0.001     -0.594      0.553      -0.002       0.001
Oldpeak               0.0394      0.014      2.801      0.005       0.012       0.067
Sex_M                 0.1921      0.032      6.031      0.000       0.130       0.255
ChestPainType_ASY     0.1900      0.026      7.312      0.000       0.139       0.241
ChestPainType_ATA    -0.0395      0.032     -1.240      0.215      -0.102       0.023
ChestPainType_NAP    -0.0078      0.030     -0.263      0.793      -0.066       0.050
ChestPainType_TA      0.0123      0.047      0.264      0.792      -0.079       0.104
FastingBS_0           0.0145      0.035      0.415      0.678      -0.054       0.083
FastingBS_1           0.1405      0.038      3.725      0.000       0.066       0.215
RestingECG_LVH        0.0670      0.034      1.983      0.048       0.001       0.133
RestingECG_Normal     0.0383      0.025      1.524      0.128      -0.011       0.088
RestingECG_ST         0.0496      0.031      1.623      0.105      -0.010       0.110
ExerciseAngina_Y      0.1343      0.031      4.323      0.000       0.073       0.195
ST_Slope_Down         0.0782      0.042      1.877      0.061      -0.004       0.160
ST_Slope_Flat         0.2417      0.029      8.375      0.000       0.185       0.298
ST_Slope_Up          -0.1649      0.034     -4.823      0.000      -0.232      -0.098
------------------------------------------------------------------------------
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:           HeartDisease   R-squared:                       0.586
Model:                            OLS   Adj. R-squared:                  0.577
Method:                 Least Squares   F-statistic:                     71.00
Date:                Sat, 25 Feb 2023   Prob (F-statistic):          3.40e-124
Time:                        01:52:08   Log-Likelihood:                -199.32
No. Observations:                 718   AIC:                             428.6
Df Residuals:                     703   BIC:                             497.3
Df Model:                          14
Covariance Type:            nonrobust
=========================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   0.3270      0.084      3.893      0.000       0.162       0.492
Cholesterol            -0.0013      0.000     -4.817      0.000      -0.002      -0.001
Oldpeak                -0.0004      0.079     -0.005      0.996      -0.155       0.154
Sex_M                   0.2015      0.032      6.364      0.000       0.139       0.264
ChestPainType_ASY       0.2031      0.028      7.200      0.000       0.148       0.258
FastingBS_1             0.1130      0.031      3.675      0.000       0.053       0.173
RestingECG_LVH          0.0278      0.031      0.884      0.377      -0.034       0.090
ExerciseAngina_Y        0.1451      0.031      4.639      0.000       0.084       0.207
ST_Slope_Flat           0.1827      0.051      3.614      0.000       0.083       0.282
ST_Slope_Up            -0.2302      0.057     -4.060      0.000      -0.342      -0.119
RestingBP Oldpeak   -1.106e-05      0.001     -0.018      0.985      -0.001       0.001
Age RestingBP MaxHR  9.896e-09   9.59e-08      0.103      0.918   -1.78e-07    1.98e-07
Age Cholesterol^2    3.901e-08   1.26e-08      3.092      0.002    1.42e-08    6.38e-08
Age MaxHR^2         -1.819e-08   6.88e-08     -0.265      0.791   -1.53e-07    1.17e-07
MaxHR Oldpeak^2         0.0001   5.21e-05      2.090      0.037    6.59e-06       0.000
==============================================================================
```

2. Classification and 3. Evaluation

For this task, we have trained logistic regression, decision tree and naïve bayes classifier with different preprocessing and parameters.

1. Logistic Regression

Logistic Regression is a statistical method used for binary classification problems. It models the relationship between the dependent variable (binary) and one or more independent variables using a logistic function. The model outputs a probability score that predicts the probability of an observation belonging to one of the two classes.

Preprocessing:

After experimenting with different techniques like standardization/ no standardization, one-hot encoding, label encoding, polynomial feature generation or no polynomial features, these settings gave good results:

Standardize: For numeric features, we have standardized them using standard scaler to have 0 mean and unit variance

Polynomial Features: We have generated polynomial features upto degree 3 using PolynomialFeatures transformation

One-hot encoding: For categorical features, one-hot encoding was performed, so each category type is converted to a column with 0 or 1

Parameters:

penalty='l2'

C=1.0

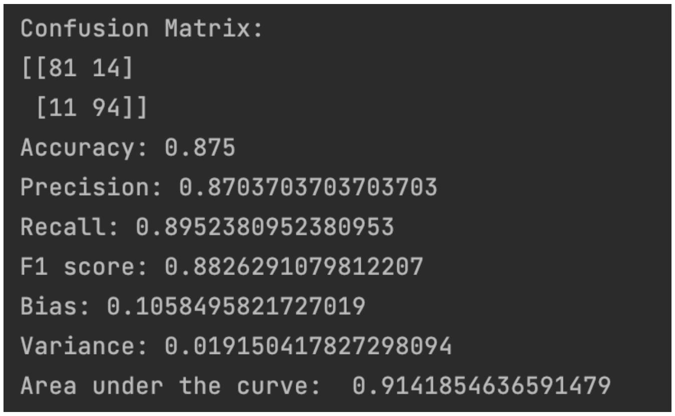
fit_intercept=True

solver='lbfgs'

max_iter=100

Results:

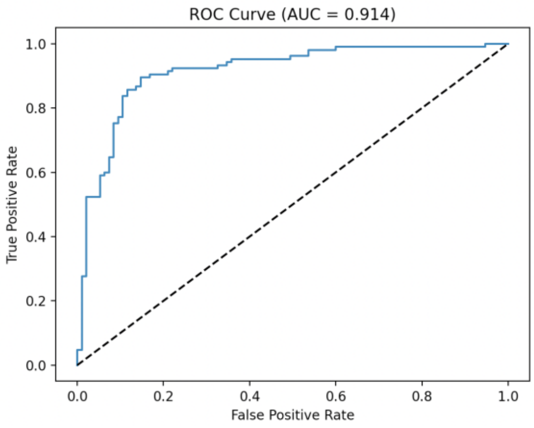
The model scored an accuracy of 87.5%, F1-score 88%, recall 89.5% and AUC ROC of 91.4%

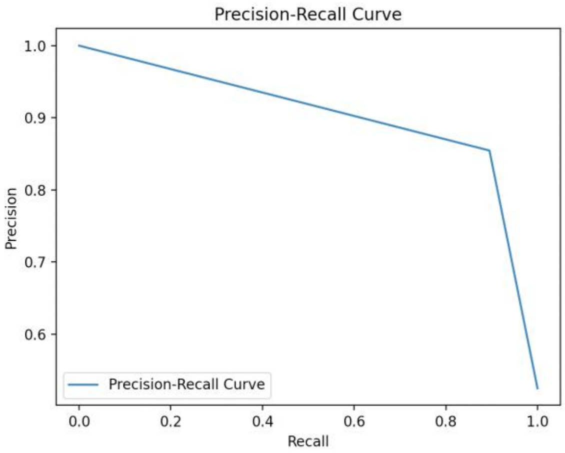


```
Confusion Matrix:
[[81 14]
 [11 94]]
Accuracy: 0.875
Precision: 0.8703703703703703
Recall: 0.8952380952380953
F1 score: 0.8826291079812207
Bias: 0.1058495821727019
Variance: 0.019150417827298094
Area under the curve:  0.9141854636591479
```

- What does the bias and variance indicate as to what the best next steps to take would be to improve performance?

The model has a high bias 10.5% and low variance 1.9%, which indicates we need a more powerful model. This can be done by introducing more polynomial features, reducing regularization, changing model to powerful model.

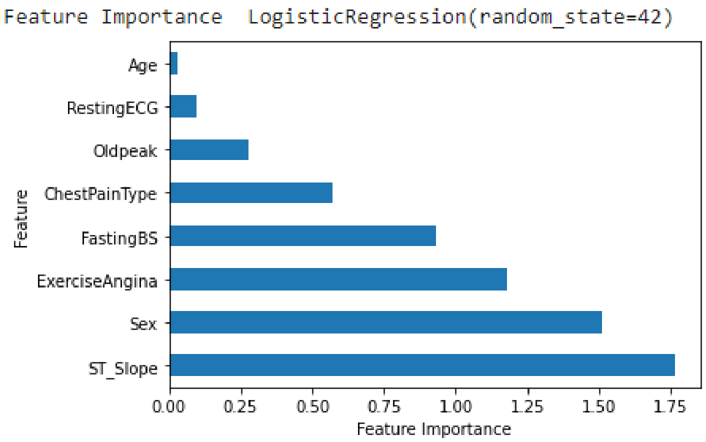




- What would be a good operating point for the classifier for which you generated the ROC curve?

From ROC AUC curve, 0.2 seems to be a good operating point with a balance of TPR and FPR.

Extension: Feature Importance plot



```
Feature Importance  LogisticRegression(random_state=42)
```

2. Decision Tree

Decision tree is a non-parametric model that builds a tree-like model of decisions based on a set of input features. The tree is constructed by recursively partitioning the feature space into subsets, where each partition is selected based on the most informative feature at that level. Once the tree is constructed, it can be used to classify new examples by following the decision path from the root to a leaf node.

Preprocessing:

No Standardize: For numeric features, we found not standardizing gave better results for decision trees.

No Polynomial Features: On comparing polynomial vs no polynomial, no polynomial features model gave better results

One-hot encoding: For categorical features, one-hot encoding was performed, so each category type is converted to a column with 0 or 1

Parameters:

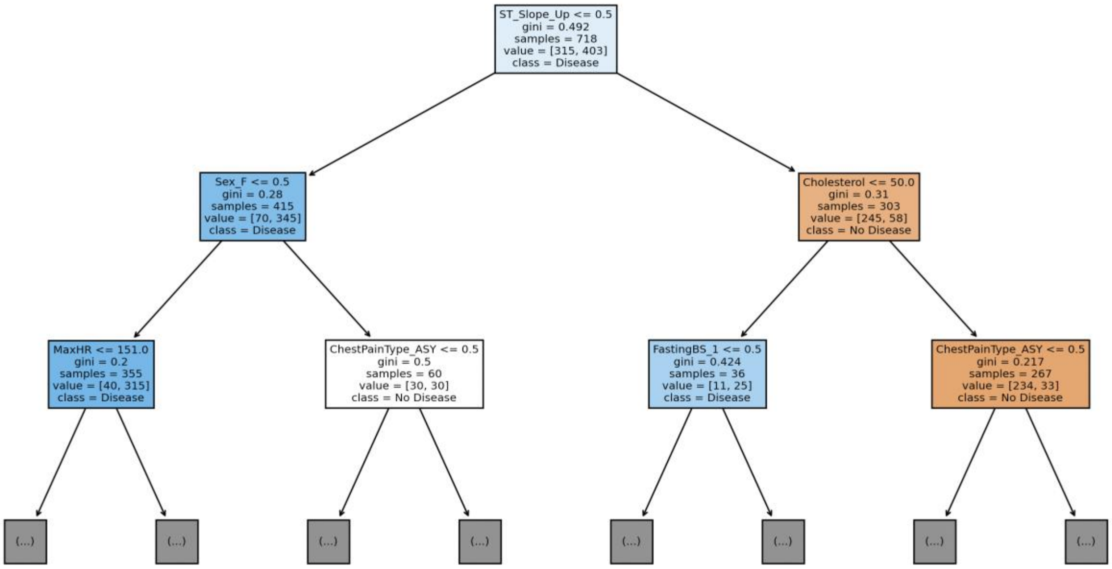
criterion='gini'

splitter='best'

min_samples_split=2

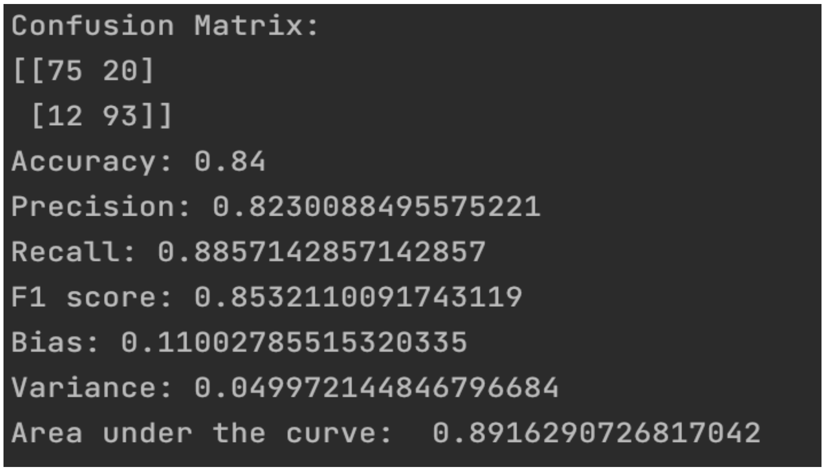
min_samples_leaf=1

This is a diagram of the first 2 levels of the decision trained on the data:



Results:

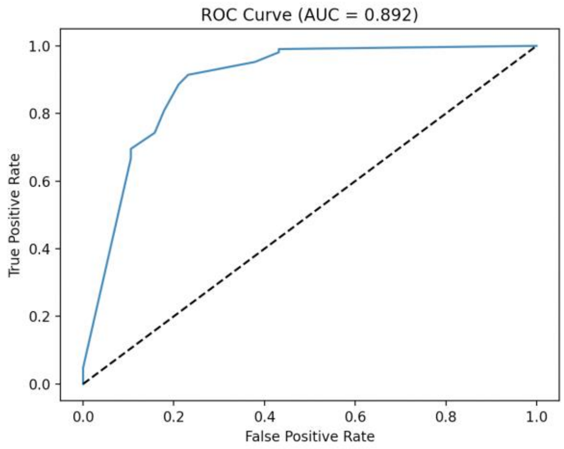
The model scored an accuracy of 84%, F1-score 85%, recall 88.5% and AUC ROC of 89.1%, which is lower than logistic regression.

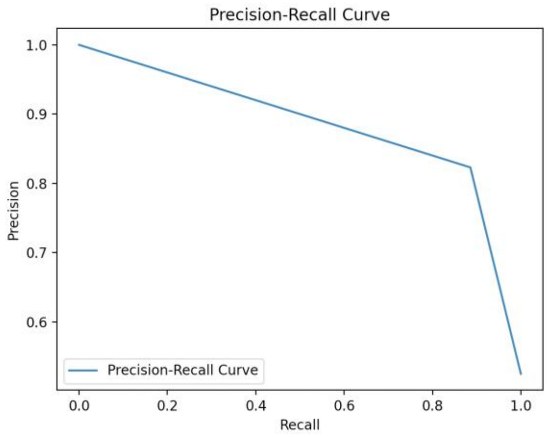


```
Confusion Matrix:
[[75 20]
 [12 93]]
Accuracy: 0.84
Precision: 0.8230088495575221
Recall: 0.8857142857142857
F1 score: 0.8532110091743119
Bias: 0.11002785515320335
Variance: 0.049972144846796684
Area under the curve:  0.8916290726817042
```

- What does the bias and variance indicate as to what the best next steps to take would be to improve performance?

The model has a high bias 11% and low variance 4.9%, which indicates we need a more powerful model. This can be done by introducing more polynomial features, reducing regularization, increasing depth, adding more trees like boosting

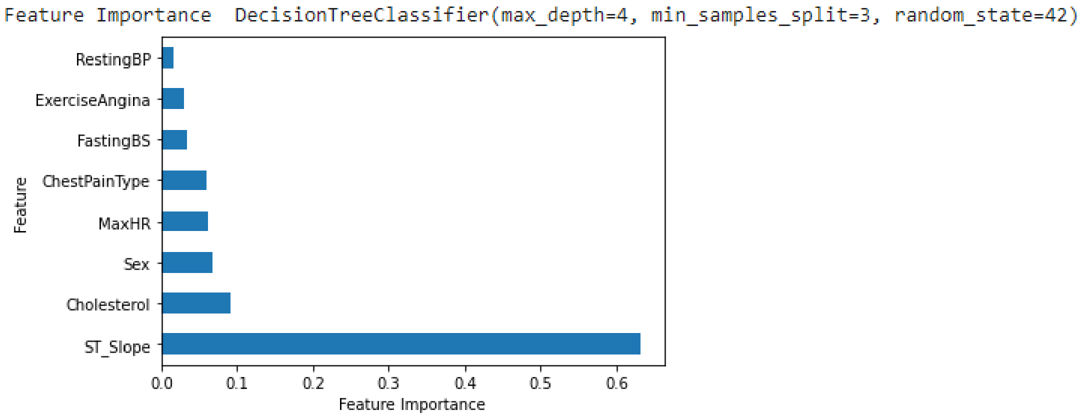




- What would be a good operating point for the classifier for which you generated the ROC curve?

From ROC AUC curve, 0.25 seems to be a good operating point with a balance of TPR and FPR.

Extension: Feature Importance plot



3. Gaussian Naïve Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the features of a dataset are independent of each other, hence the term "naive". It calculates the probability of a new data point belonging to each class and chooses the one with the highest probability as the predicted class.

Preprocessing:

Standardize: For numeric features, we have standardized them using standard scaler to have 0 mean and unit variance

No Polynomial Features: On comparing polynomial vs no polynomial, no polynomial features model gave better results

One-hot encoding: For categorical features, one-hot encoding was performed, so each category type is converted to a column with 0 or 1

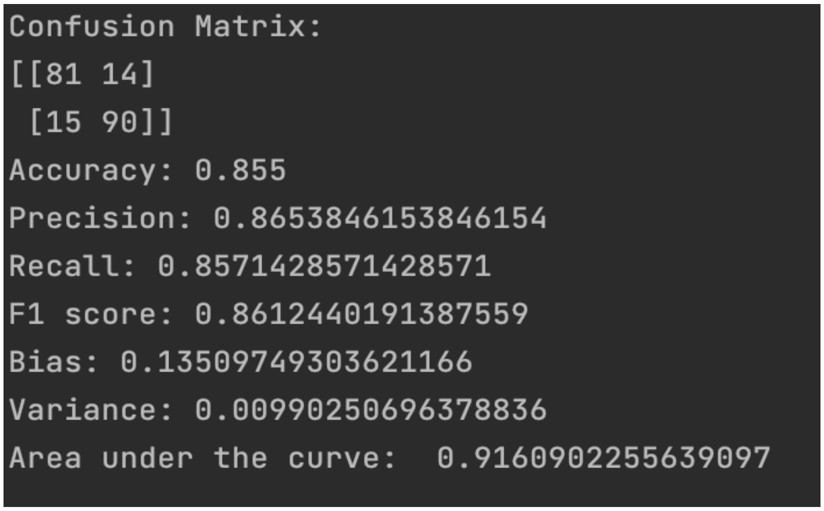
Parameters:

*priors=None (*Prior probabilities of the classes)

*var_smoothing=1e-09 (*Portion of the largest variance of all features that is added to variances for calculation stability)

Results:

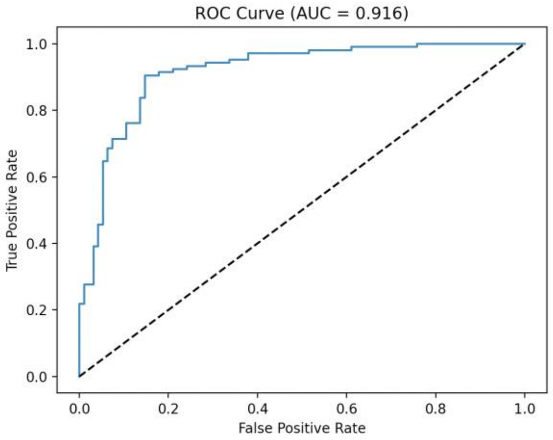
The model scored an accuracy of 85.5%, F1-score 86.1%, recall 85.7%and AUC ROC of 91.6%, which is lower than logistic regression.

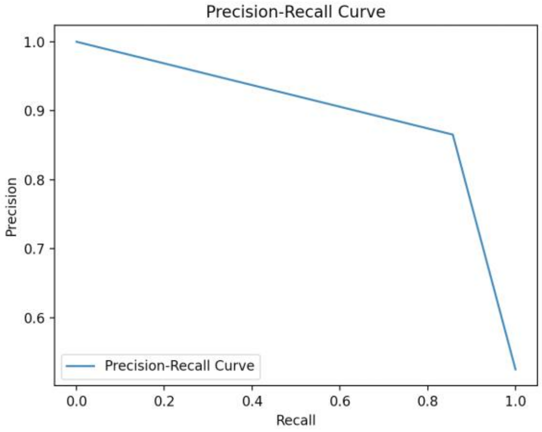


```
Confusion Matrix:
[[81 14]
 [15 90]]
Accuracy: 0.855
Precision: 0.8653846153846154
Recall: 0.8571428571428571
F1 score: 0.8612440191387559
Bias: 0.13509749303621166
Variance: 0.00990250696378836
Area under the curve:  0.9160902255639097
```

- What does the bias and variance indicate as to what the best next steps to take would be to improve performance?

The model has a high bias 13.5% and extremely low variance 0.9%, which indicates we need a more powerful model. This can be done by introducing more polynomial features, reducing regularization, or changing model to a more complex model.





- What would be a good operating point for the classifier for which you generated the ROC curve?

From ROC AUC curve, 0.18 seems to be a good operating point with a balance of TPR and FPR.

- What statistic are you using to decide which classifier is the best performer?

We are using Recall to decide the best performing classifier. Recall tells us how well the model can capture all the cases of heart disease in the dataset. We assume, it is more important to minimize false negatives (i.e., cases where the model incorrectly predicts that a patient does not have heart disease when they actually do), than false positives (i.e., cases where the model incorrectly predicts that a patient has heart disease when they do not). False negatives can have serious consequences for patients, such as delayed treatment or misdiagnosis and we should be able to retrieve maximum possible true positives.

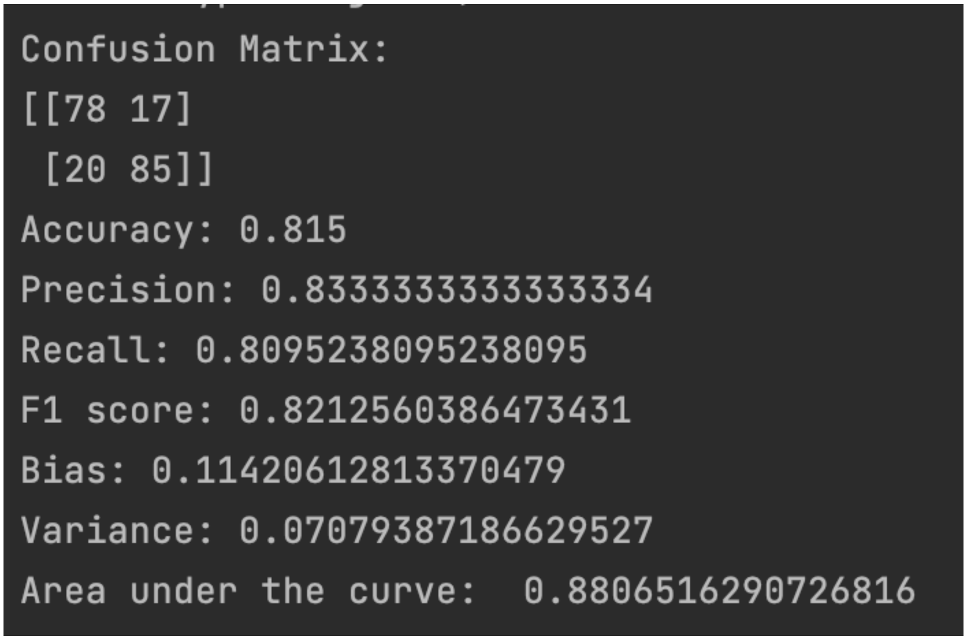
- Which classifier was the best?

Based on these experiments, Logistic Regression seems to perform best with 89.5% Recall. After performing iterations and hyperparameter tuning, Decision tree seems to perform best with 93.33% recall as mentioned in next section.

3. Iteration

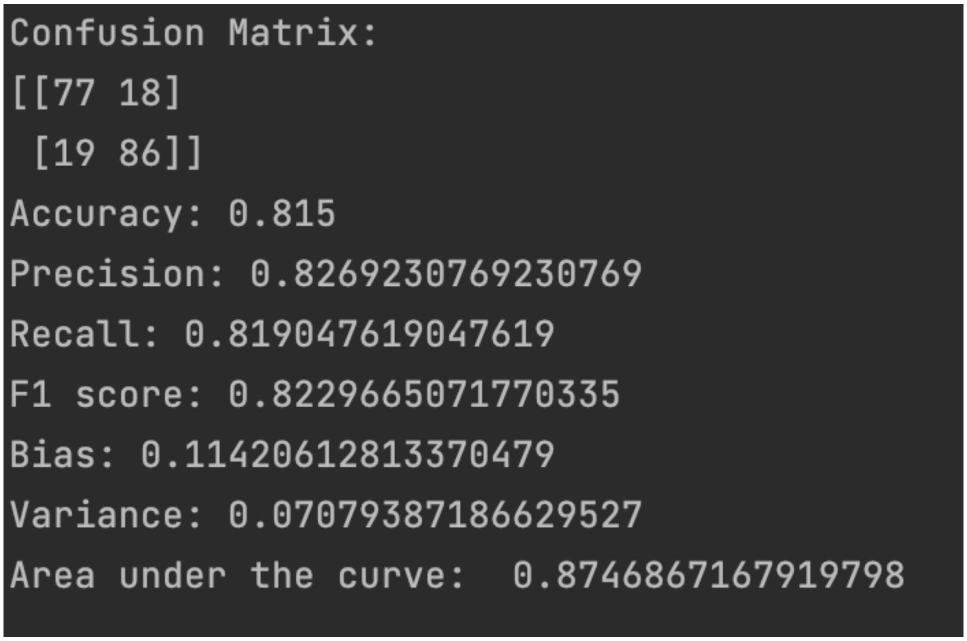
For this task, we decided to use Decision Tree and improve the Recall.

For the baseline decision tree model, we got the following performance. For this model, we used the generated polynomial features and encoded the categorical features using label encoding. The recall was 80.9%



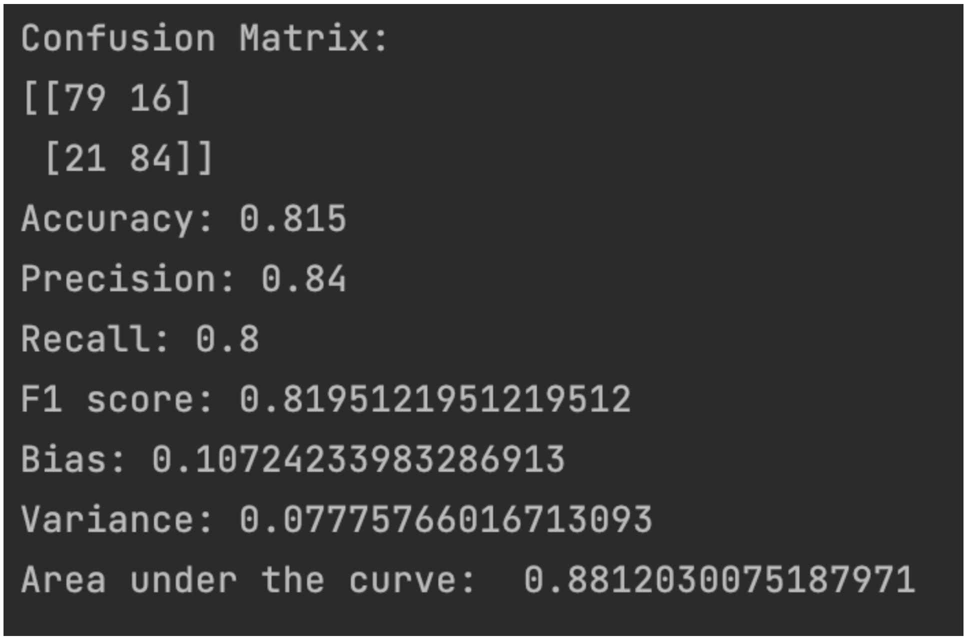
```
Confusion Matrix:
[[78 17]
 [20 85]]
Accuracy: 0.815
Precision: 0.8333333333333334
Recall: 0.8095238095238095
F1 score: 0.8212560386473431
Bias: 0.11420612813370479
Variance: 0.07079387186629527
Area under the curve:  0.8806516290726816
```

Change: Encoding changed to one-hot encoding from label encoding. This increased the recall to 81.9% from 80.9%



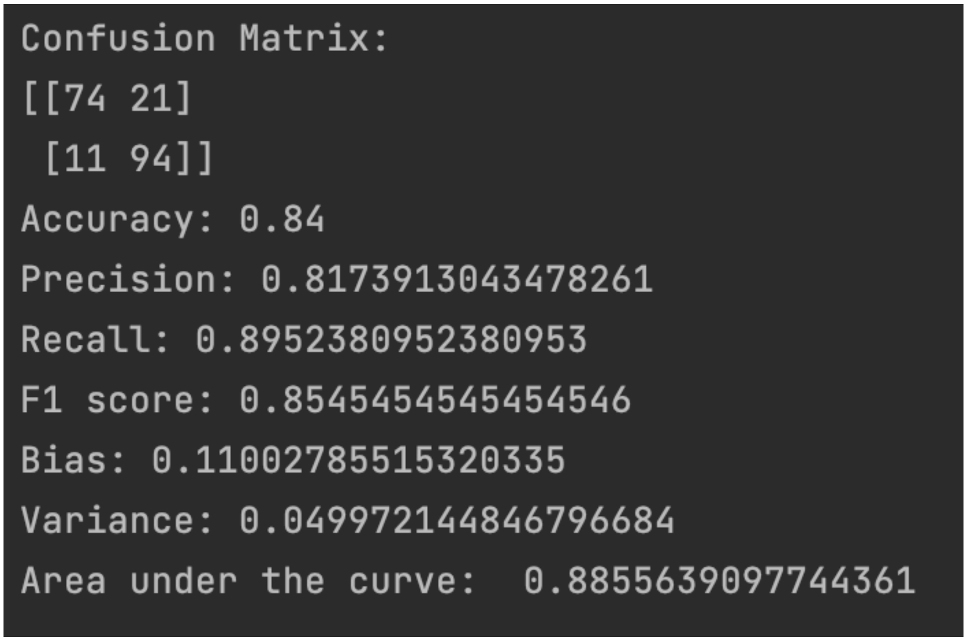
```
Confusion Matrix:
[[77 18]
 [19 86]]
Accuracy: 0.815
Precision: 0.8269230769230769
Recall: 0.819047619047619
F1 score: 0.8229665071770335
Bias: 0.11420612813370479
Variance: 0.07079387186629527
Area under the curve:  0.8746867167919798
```

Change: We then introduced standardization of numeric features. But this decreased the recall to 80% from 81.9%. So, we removed standardization.



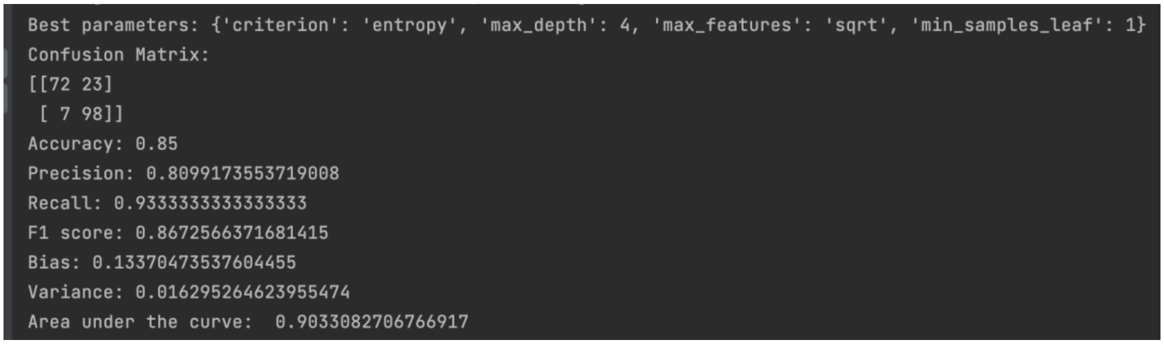
```
Confusion Matrix:
[[79 16]
 [21 84]]
Accuracy: 0.815
Precision: 0.84
Recall: 0.8
F1 score: 0.8195121951219512
Bias: 0.10724233983286913
Variance: 0.07775766016713093
Area under the curve:  0.8812030075187971
```

Change: we removed the polynomial features to see if they were helping or reducing recall of the model. This increased the recall significantly from 81.9% to 89.5%, thus the model was learning a lot better without the polynomial features.



```
Confusion Matrix:
[[74 21]
 [11 94]]
Accuracy: 0.84
Precision: 0.8173913043478261
Recall: 0.8952380952380953
F1 score: 0.8545454545454546
Bias: 0.11002785515320335
Variance: 0.04997214484679684
Area under the curve:  0.8855639097744361
```

Change: For this, we did hyperparameter tuning with grid search for extension. (More details in extension). This increase recall significantly to 93.3%



```
Best parameters: {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1}
Confusion Matrix:
[[72 23]
 [ 7 98]]
Accuracy: 0.85
Precision: 0.8099173553719008
Recall: 0.9333333333333333
F1 score: 0.8672566371681415
Bias: 0.13370473537604455
Variance: 0.016295264623955474
Area under the curve:  0.9033082706766917
```

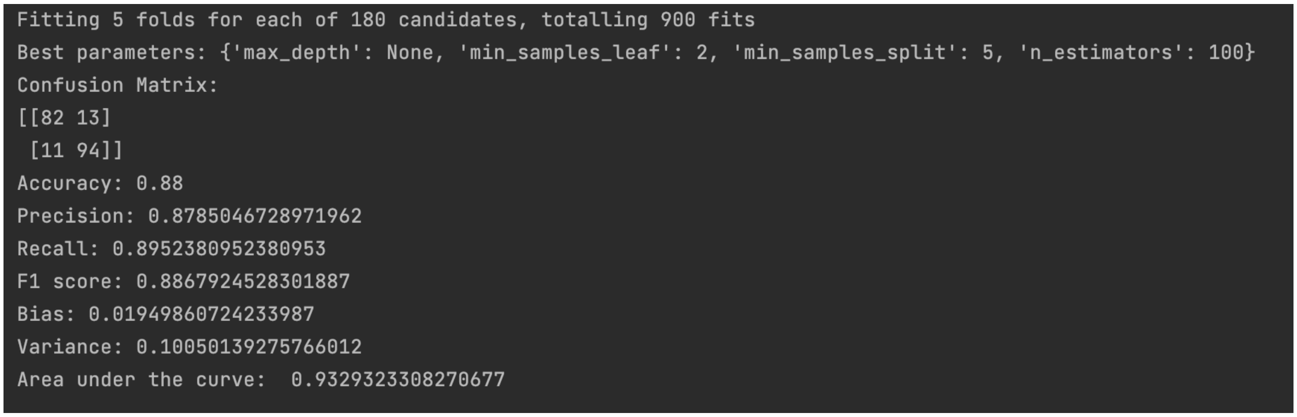
Preprocessing	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC ROC**
Label encoding, With polynomial features (Deg 3)	81.5	83.3	80.9	82.1	88
One-hot encoding, With polynomial features (Deg 3)	81.5	82.6	81.9	82.2	87.4
One-hot encoding, With polynomial features (Deg 3) and standardization	81.5	84	80	81.9	88.1
One-hot encoding, No polynomial features	84	82.3	88.5	85.3	89.1
Label encoding, No polynomial features	84	81.7	89.5	85.4	88.5
Hyperparameter tuning (extension)	85	80.9	93.3	86.7	90.3

Extensions

Extension 1: Hyperparameter tuning with grid search

For this extension, we have performed hyperparameter tuning using grid search for different algorithms. Grid search is a method for hyperparameter tuning that exhaustively searches through a predefined subset of hyperparameters to find the optimal combination for a machine learning model. It involves defining a grid of hyperparameter values and evaluating the model's performance for each combination of values. For different ML methods, we provide a list of possible hyperparameter values and train models to find the best parameters. This improves the performance of a baseline model.

1. **Random Forest**



```
Fitting 5 folds for each of 180 candidates, totalling 900 fits
Best parameters: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}
Confusion Matrix:
[[82 13]
 [11 94]]
Accuracy: 0.88
Precision: 0.8785046728971962
Recall: 0.8952380952380953
F1 score: 0.8867924528301887
Bias: 0.01949860724233987
Variance: 0.10050139275766012
Area under the curve:  0.9329323308270677
```

Classifier	Tuning	Accuracy	Precision	Recall	F1 Score	AUC ROC
Random Forest	No	86.5	87.5	86.6	87	92.7
Random Forest	Yes	88	87.8	89.5	88.6	93.9

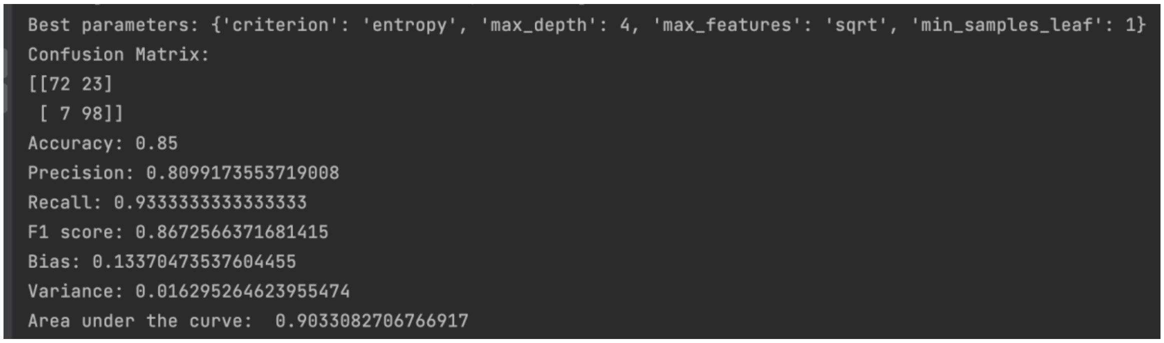
By tuning, we have increased recall from 86.6% to 89.5% for the random forest

2. Adaboost

Classifier	Tuning	Accuracy	Precision	Recall	F1 Score	AUC ROC
Adaboost	No	84.5	84.2	86.6	85.4	90.6
Adaboost	Yes	84	82.8	87.6	0.85	92.4

By tuning, we have increased recall from 86.6% to 87.5% for adaboost model

3. Decision Tree



```
Best parameters: {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1}
Confusion Matrix:
[[72 23]
 [ 7 98]]
Accuracy: 0.85
Precision: 0.8099173553719008
Recall: 0.9333333333333333
F1 score: 0.8672566371681415
Bias: 0.13370473537604455
Variance: 0.016295264623955474
Area under the curve:  0.9033082706766917
```

Classifier	Tuning	Accuracy	Precision	Recall	F1 Score	AUC ROC
Decision Tree	No	84	81.7	89.5	85.4	88.5

Decision Tree	Yes	85	80.9	93.3	86.7	90.3

By tuning, we have increased recall from 89.5% to 93.3% for decision tree model, which is the best model.

Extension 2: Iterations on additional classifiers

For every classifier, we have performed iterations of training with different Preprocessing and features. We have trained the classifiers with polynomial features and without polynomial features. For tree-based methods, we have iterated with label encoding and one-hot encoding as they can handle categorical data.

As mentioned in previous extension, we have also tuned hyperparameters of some classifiers to get even better performance

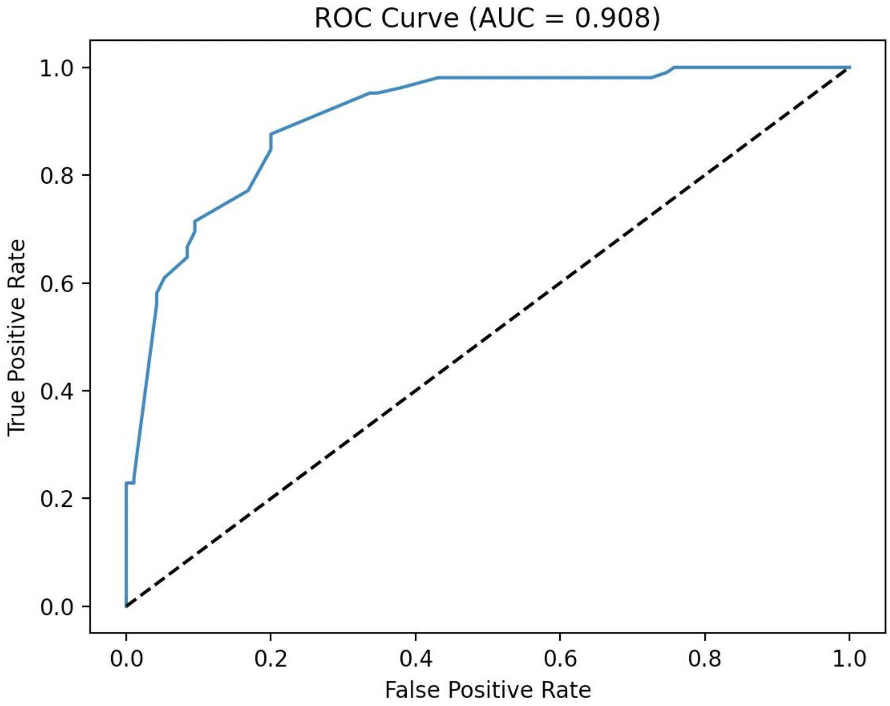
Classifier	**Preprocessing**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC ROC**
Logistic Regression	Polynomial Features (Deg 3)	87.5	87	89.5	88.2	91.4
Logistic Regression	No Polynomial Features	87	85.5	90.4	87.9	91.8
Adaboost	Label encoding, No polynomial features	86.6	86.6	86.6	86.6	91.5
Adaboost	One-hot encoding, No polynomial features	86.6	86.6	86.6	86.6	91.5
Adaboost	Label encoding, With polynomial features (Deg 3)	85	84.4	87.6	85.9	90.9
Adaboost	One-hot encoding, With polynomial features (Deg 3)	84.5	84.2	86.6	85.4	90.6
Random Forest	Label encoding, No polynomial features	88	87.1	90.4	88.7	92.1
Random Forest	One-hot encoding, No polynomial features	86	85.9	87.6	86.7	91.7

Random Forest	Label encoding, With polynomial features (Deg 3)	86	84.6	89.5	87	92.4
Random Forest	One-hot encoding, With polynomial features (Deg 3)	86.5	87.5	86.6	87	92.7
Naïve Bayes	Polynomial Features (Deg 3)	79.5	85.5	73.3	78.9	85.6
Naïve Bayes	No Polynomial Features	85	86.4	84.7	85.5	91.5

Extension 3: ROC Curves of additional classifiers

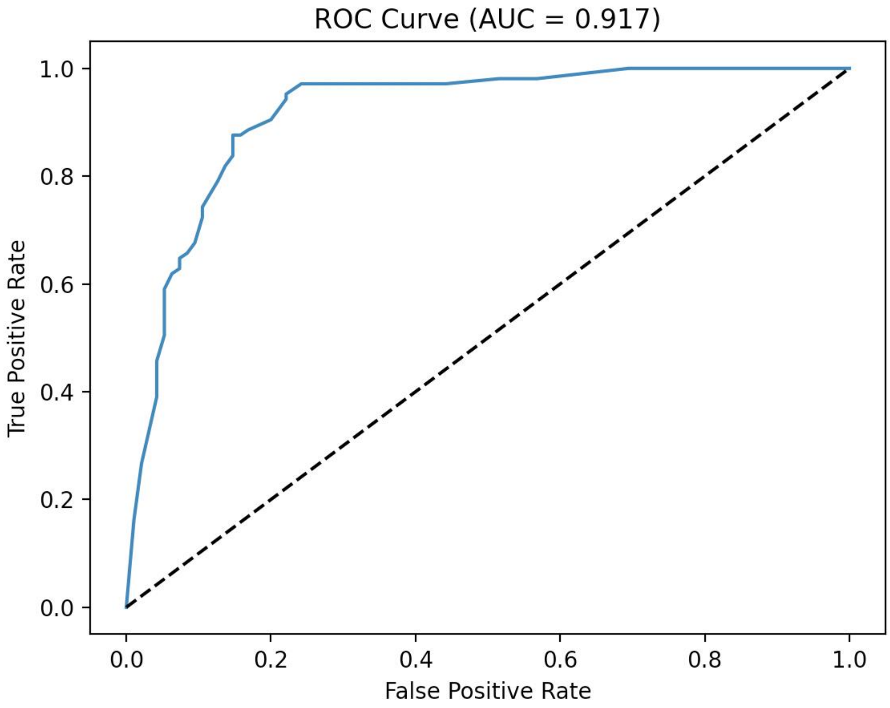
Adaboost

From ROC AUC curve, 0.21 seems to be a good operating point with a balance of TPR and FPR.



Random Forest

From ROC AUC curve, 0.25 seems to be a good operating point with a balance of TPR and FPR.



Extension 4: Used more than required ML methods

For this extension, we have tried more than the required 3 ML methods for classification, including Adaboost, Logistic regression, decision tree, naïve bayes, random forest etc.

Reflection

Throughout the project, we were presented with the chance to apply our acquired skills to tackle a fresh problem statement. Our team effectively utilized a dataset by conducting a thorough exploratory data analysis (EDA) and deepening our understanding of the data. Additionally, we applied multiple machine learning techniques to address the task at hand. The project also allowed us to explore innovative concepts such as feature generation and hyperparameter tuning, opening new possibilities for future endeavors.

Acknowledgement

Professor Maxwell's classes were helpful to implement the concepts in this project.