Deliverable II
Marwan Jabbour
Due: October 19, 2020
Dataset: https://www.kaggle.com/johnsmith88/heart-disease-dataset

Evaluation Metric: My project is a classification problem: *Does the patient have heart disease?* The 14th column of my data set indicates whether or not the person has heart disease (1) or does not have heart disease (0).

Data Preprocessing: My initial dataset was quite small, so I replaced it with a new data set that contains 1025 entries. The 14 feature labels are identical to those of the previous data set. There seems to be no duplicate examples and no omitted values. The feature labels are good and there is no inconsistency.

Step 1: Visualize the Data

	age	sex	cp	trestbps	chol	...	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	...	1.0	2	2	3	0
1	53	1	0	140	203	...	3.1	0	0	3	0
2	70	1	0	145	174	...	2.6	0	0	3	0
3	61	1	0	148	203	...	0.0	2	1	3	0
4	62	0	0	138	294	...	1.9	1	3	2	0
...	...	...	..	...	...	...	...	...	..	...	...
1020	59	1	1	140	221	...	0.0	2	0	2	1
1021	60	1	0	125	258	...	2.8	1	1	3	0
1022	47	1	0	110	275	...	1.0	1	1	2	0
1023	50	0	0	110	254	...	0.0	2	0	2	1
1024	54	1	0	120	188	...	1.4	1	1	3	0

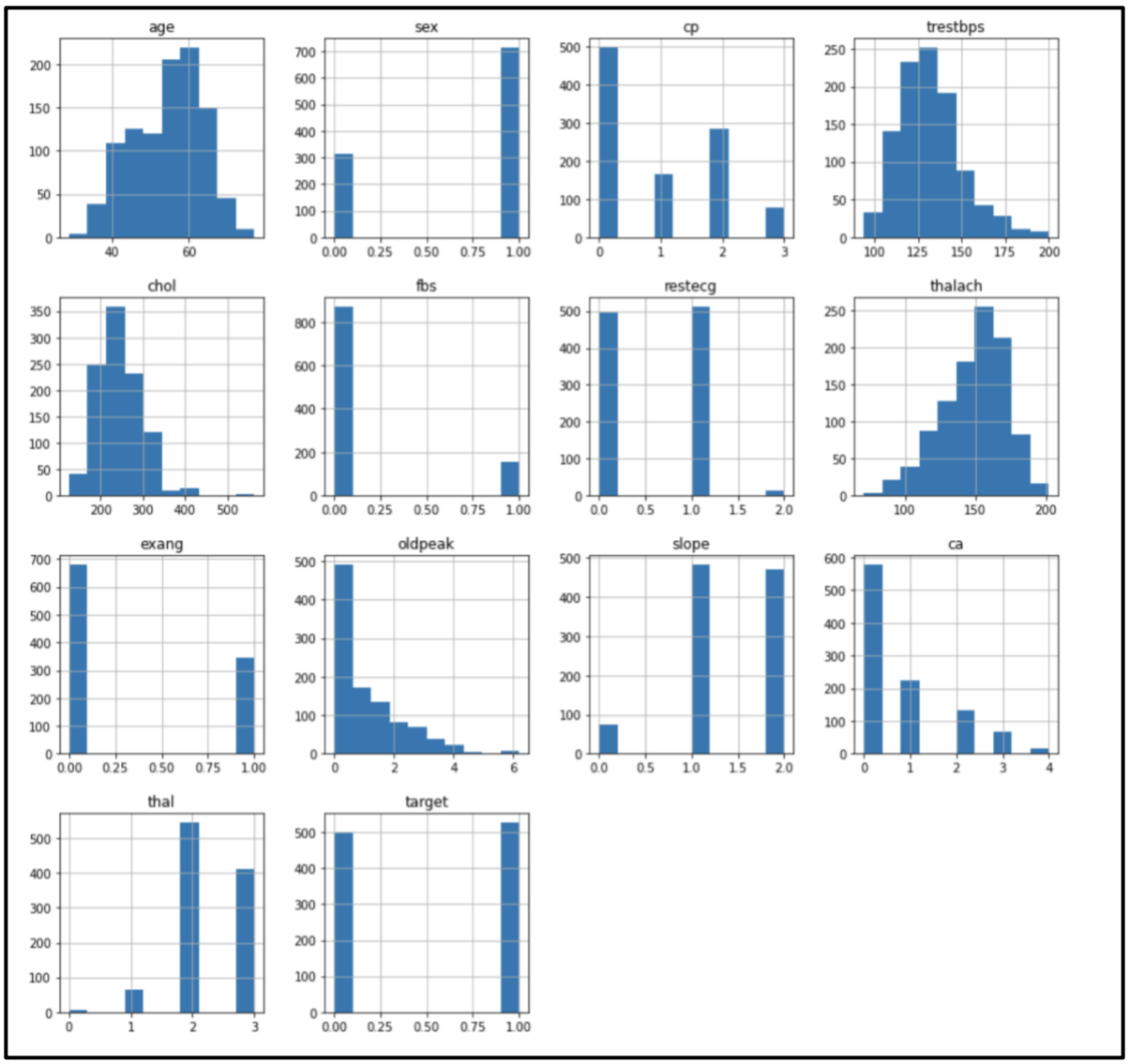
[1025 rows x 14 columns]

	age	sex	...	thal	target
count	1025.000000	1025.000000	...	1025.000000	1025.000000
mean	54.434146	0.695610	...	2.323902	0.513171
std	9.072290	0.460373	...	0.620660	0.500070
min	29.000000	0.000000	...	0.000000	0.000000
25%	48.000000	0.000000	...	2.000000	0.000000
50%	56.000000	1.000000	...	2.000000	1.000000
75%	61.000000	1.000000	...	3.000000	1.000000
max	77.000000	1.000000	...	3.000000	1.000000

[8 rows x 14 columns]

target	
0	499
1	526

As can be seen above, the mean of the target is 0.513. This is important in eliminating bias, as there is a relatively equal number of patients with and without heart disease. Note that there are more males (69.5%) than females (30.5%), and that the patients are relatively old (with an average age of ~ 54).



Step 2: Split the Data

I will assign 70% of my data to the training set, 15% to the validation set, and another 15% to the training set. I will do this using the **train_test_split** method twice.

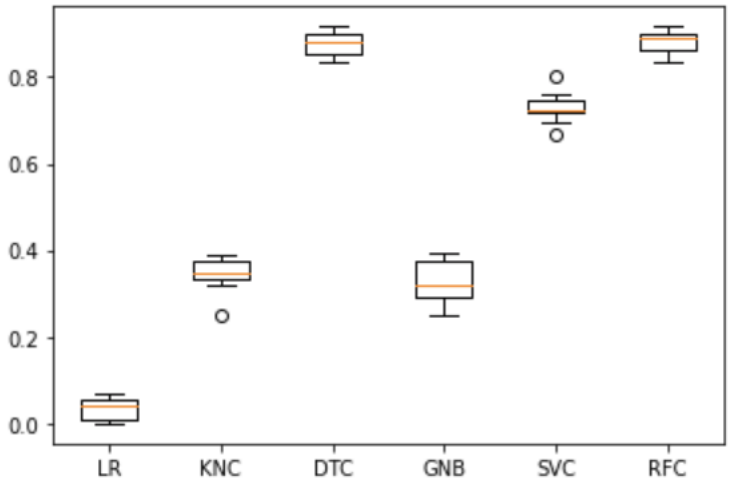
Machine Learning Model:

I decided to study my data using 6 models, and decide on the best one. The models are: Logistic Regressor, K-Neighbors Classifier, Decision Tree Classifier, Gaussian Naïve Bayes, Support Vector Machine, and Random Forest Classifier.

First, I will use the StratifiedKFold function for each model to split my training set. Then, I will implement them using the scikit-learn library (cross_val_score).

```
Logistic Regressor:          0.03480046948356807 0.02510466925906273
KNeighborsClassifier:        0.34593114241001566 0.038575503050078626
DecisionTreeClassifier:      0.8773082942097027 0.026036592498666973
GaussianNB:                  0.3293622848200313 0.049009906051344455
Support Vector Machine:      0.7295579029733961 0.03502179613127894
RandomForestClassifier:      0.8800860719874803 0.024875369262078787
```

In the table above, the first column indicates the mean and the second column indicates the standard deviation of the accuracy. The most accurate models seem to me the Decision Tree Classifier, the Support Vector Machine, and the Random Forest Classifier.



Based on the above 2 graphs, I decided to test the Random Forest Classifier against the Decision Tree classifier for my machine learning models. When tested on the validation set, the former has an accuracy of 0.948 while the second has an accuracy of 0.928. In addition, the macro F1 score is 0.925 for the first and 0.889 for the second. So, I decided to choose the Random Forest Classifier as my machine learning model.

Next Steps:

1) Fine-tuning the model (if necessary)
2) Regularization and Optimization
3) Hyper parameters: number of decision trees in the forest - number of features considered by each tree when a node is split