Project Description

Heart disease is the major cause of morbidity and mortality globally: it accounts for more deaths annually than any other cause. According to the WHO, an estimated 17.9 million people died from heart disease in 2016, representing 31% of all global deaths. Over three quarters of these deaths took place in low- and middle-income countries.

Of all heart diseases, coronary heart disease (aka heart attack) is by far the most common and the most fatal. In the United States, for example, it is estimated that someone has a heart attack every 40 seconds and about 805,000 Americans have a heart attack every year (CDC 2019).

Doctors and scientists alike have turned to machine learning (ML) techniques to develop screening tools and this is because of their superiority in pattern recognition and classification as compared to other traditional statistical approaches.

In this project, We will be giving you a walk through on the development of a screening tool for predicting whether a patient has a 10-year risk of developing coronary heart disease(CHD) using different Machine Learning techniques.

Project Workflow

1. Splitting to Train,Validation and Test sets to avoid Data bleeding.

2. Simultaneously Data Cleaning of Train and Test sets

3. EDA on features

4. Feature cleaning if needed

5. Solving Class Imbalanced problem

6. Base Model and Candidate Models

7. There Hypertuning

8. Bias-Variance tradeoff

9. Creating Voting Classifier

10. Final Predictions

Feature Engineering

* From the above EDA we try to establish some patterns which influence the cause of heart disease in that we found people both men and women lying in a particular age group **40-42, 50-51** are more prone to heart disease.

* So what I want to try is to create age buckets of population e.g 18-25 -> **20s**, 25-40 -> **Mid30s**
etc in this way we can target the particular age group which have high risk of Heart disease.

Class Imbalanced issue and Evaluation-metric to be chosen

In this problem we have a dataset of patients where we have to find out whether the given features or symptom a person has he/she has a Cardiovascular disease in future.

But here’s the catch… the risk rate is relatively rare, only 15% of the people have this disease.

The Metric Trap

One of the major issues when dealing with unbalanced datasets relates to the metrics used to evaluate their model. Using simpler metrics like accuracy score can be misleading. In a dataset with highly unbalanced classes, the classifier will always “predict” the most common class without performing any analysis of the features and it will have a high accuracy rate, obviously not the correct one.

Random Over-Sampling

Oversampling can be defined as adding more copies to the minority class. Oversampling can be a good choice when you don't have a ton of data to work with.

A con to consider when undersampling is that it can cause overfitting and poor generalization to your test set.

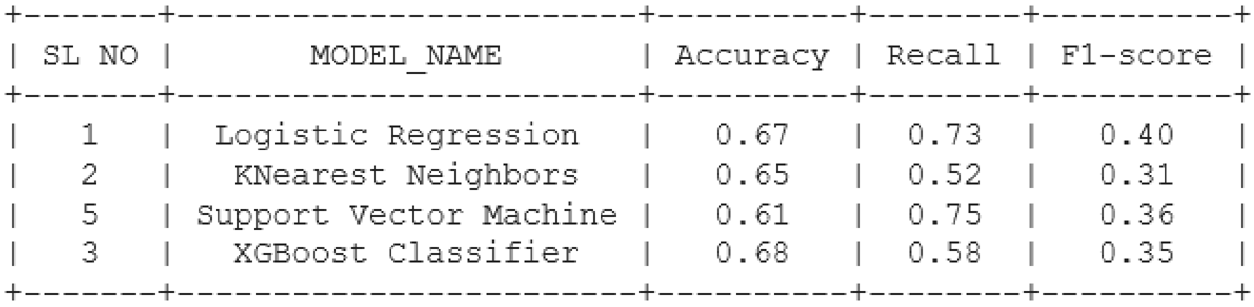
Model Training Algorithm used

1. Logistic Regression
2. Decision Tree
3. K-Nearest Neighbors
4. Support Vector Machine

After training each model and tuning their hyper-parameters using grid search, I evaluated and compared their performance using the following metrics:

- **The accuracy score**: which is the ratio of the number of correct predictions to the total number of input samples. It measures the tendency of an algorithm to classify data correctly.
- **The F1 Score**: Which is defined as the weighted harmonic mean of the test's precision and recall. By using both precision and recall its gives a more realistic measure of a test's performance. (Precision, also called the positive predictive value, is the proportion of positive results that truly are positive. Recall, also called sensitivity, is the ability of a test to correctly identify positive results to get the true positive rate).
- **The Recall**: Which provides an aggregate measure of performance across all possible classification thresholds. It gives the probability that the model ranks a random positive example more highly than a random negative example

Here are the results:



```
+-------+------------------------+----------+--------+----------+
| SL NO |       MODEL_NAME       | Accuracy | Recall | F1-score |
+-------+------------------------+----------+--------+----------+
|   1   |  Logistic Regression   |   0.67   |  0.73  |   0.40   |
|   2   |   KNearest Neighbors   |   0.65   |  0.52  |   0.31   |
|   5   | Support Vector Machine |   0.61   |  0.75  |   0.36   |
|   3   |   XGBoost Classifier   |   0.68   |  0.58  |   0.35   |
+-------+------------------------+----------+--------+----------+
```

Conclusion

We have used Logistic Regression, KNN, SVC and XGBoost for modelling. Based on our observations, the Support vector classifier seems to have performed better with a recall of 0.75, after some hyper-parameter tuning. However the precision of our models still remains a concern.

Scope for improvement : Need to work on improving the precision (currently ~ 0.26) of your model to some extent !