IE7860: Intelligent Analytics

Assignment 7 : Decision Tree & Ensemble

Heart Diseases UCI

Overview

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

Data Description

Features:

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

The names and social security numbers of the patients were recently removed from the database, replaced with dummy values. One file has been "processed", that one containing the Cleveland database. All four unprocessed files also exist in this directory.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Data Preprocessing

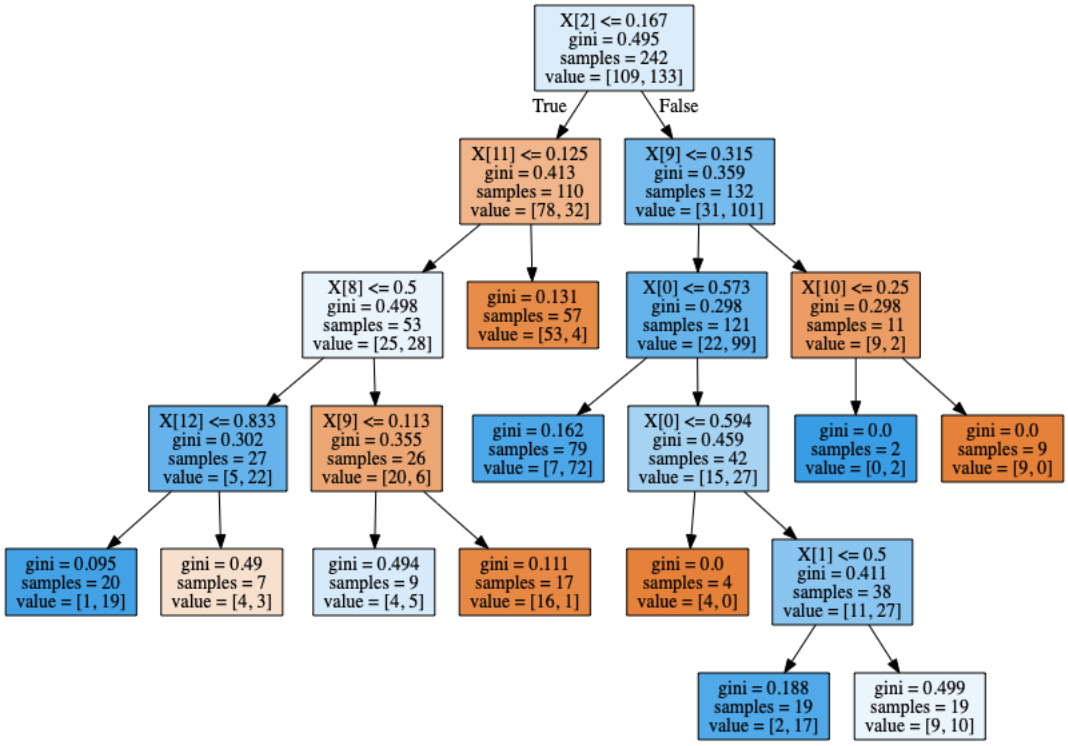
Since the data is relatively complete and there are no outliers or issues with the data only pre-processing required for it was to normalize the values. We can see in the following table that there are 303 data points for the dataset and we can also see that all 303 data points are complete for each of the features. This was done to ensure that no feature with the extreme values is given more importance. Secondly the data has to be converted into train so as to evaluate the models. The train and Test split was done with the ratio of 0.2. The size of the test data was maintained small as the data was in its self very small and only had a few entries.

```
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
```

Modelling Various Decision Trees and Ensemble Models

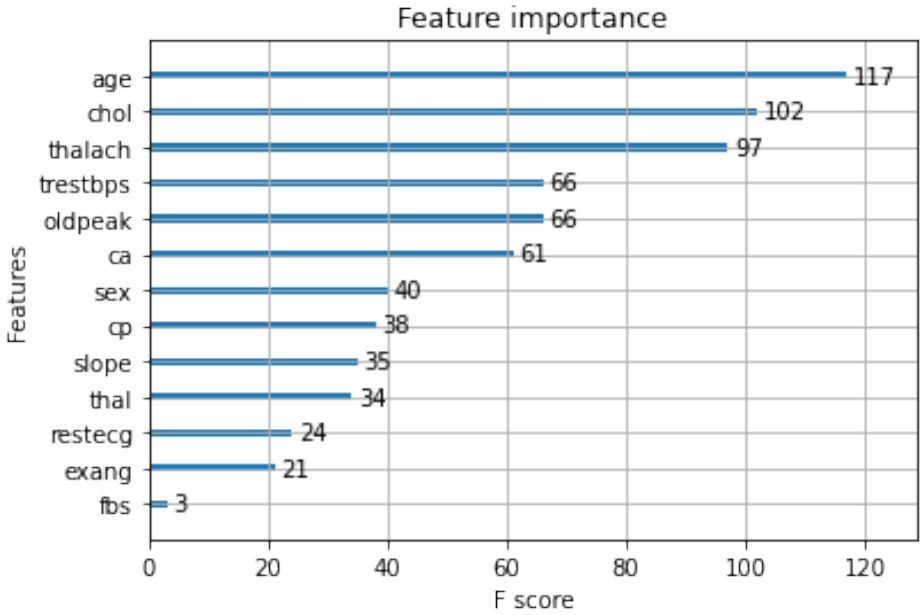
Single Decision Tree

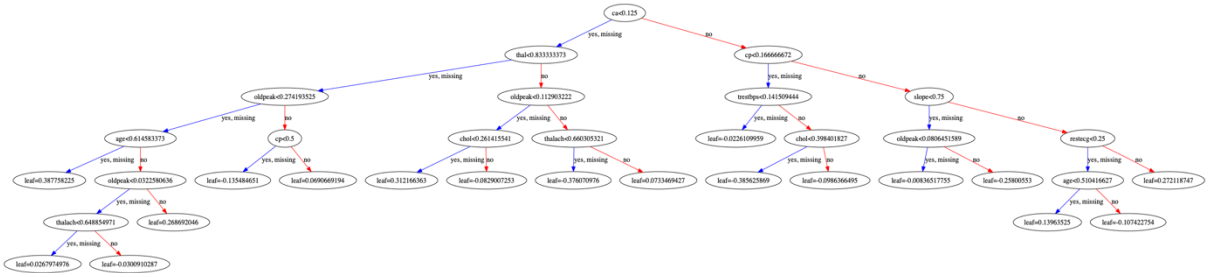
Single decision tree is rather an uninteresting and easy to implement. Using the decision tree classifier in a scikit-learn we could generate a simple single decision tree model. One key observation is that even after being a very simple model introduced better results than a few other models that were tested. This will be later discussed comparing the various models and concluding. The model was then fitted with great search and the best parameters we use to achieve the best possible accuracy over this data. Even after which there was no change in the testing accuracy. The concluding testing at receive 85.24%. .The following image highlights the architecture of the decision tree. Each of the notes also highlights the number of samples that were classified in each of the notes along with genius score for each of them and their value range.



XGBoost with Feature Importance and SHAP

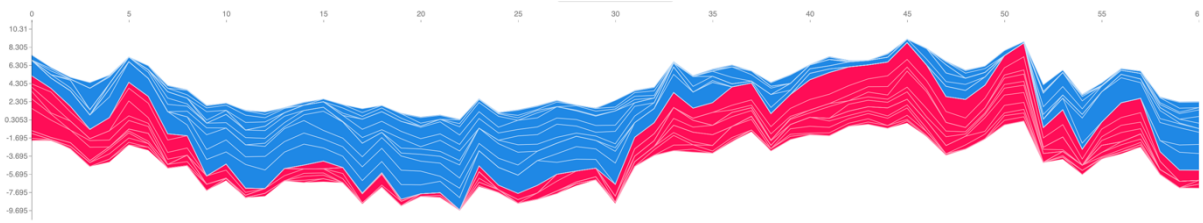
XGBoost is a relatively new ensemble method with boosting capabilities that better most other models. Only issue is the fact that it feels to accurately classify for relatively small data sets. Since the dataset used in this report is relatively small but only 303 samples, XGBoost got an accuracy of 81% only. Even though the accuracy is relatively close to the simple decision tree, such a small change will also not cause a huge number of samples being missed classified. The figure below is the feature importance diagram for XGboost.It highlights the fact that age, chol, thalach were critically important features. Whereas trestbps, oldpeak, ca were of mediocre importance. Features such as fbs, exang, restecg .very less significant and do not contribute much to the model.



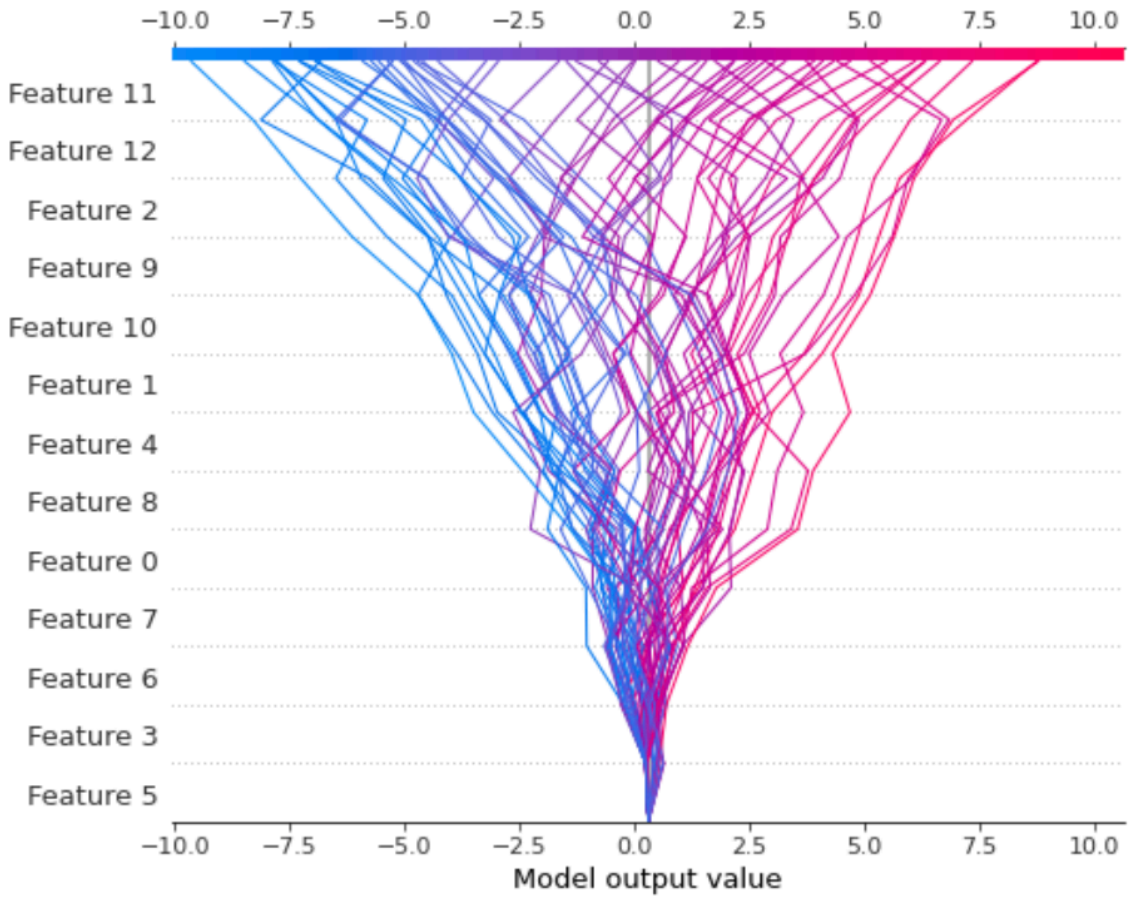


I want to SHAP explainable AI analysis, disguises a totally different story from the default feature importance from XGBoost. Interpreting from this we can see that CE and THAL contribute more to words the model and its classification whereas the non-contributing features are the same. Age which was previously a very important feature has now been demoted to defeat the least important feature.

The blue segment in the following graph the first few features which are of high importance in the middle of the train whereas the others features important to the default feature importance plot by contributing later on in the training process. Hence the disparity between the two features importance plots.



The next image reinforces the above hypothesis and depicts the converging of the input variables into the final output variable. This also highlights the degree of convergence based on the features. Play feature 11,12, 9 and 10 contribute the most.



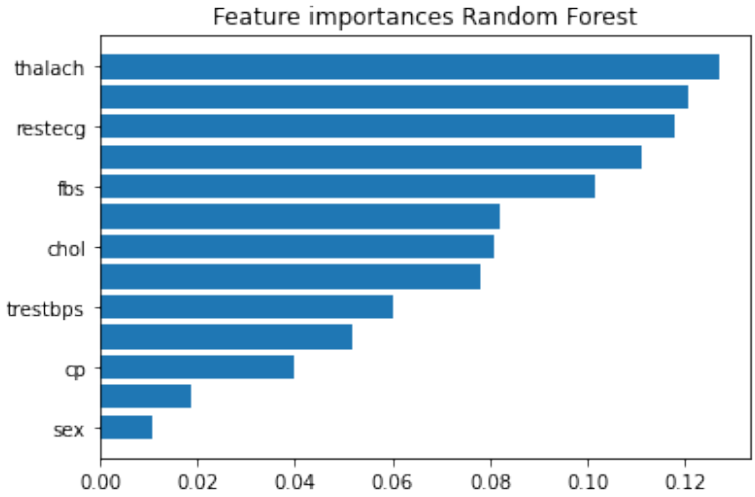
Bagging Model

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator similar to decision trees by introducing randomization into its construction procedure and then making an ensemble out of it. Can simply call the bagging classifier function that is built into scikit-learn. This model gave an accuracy of 83.6%.

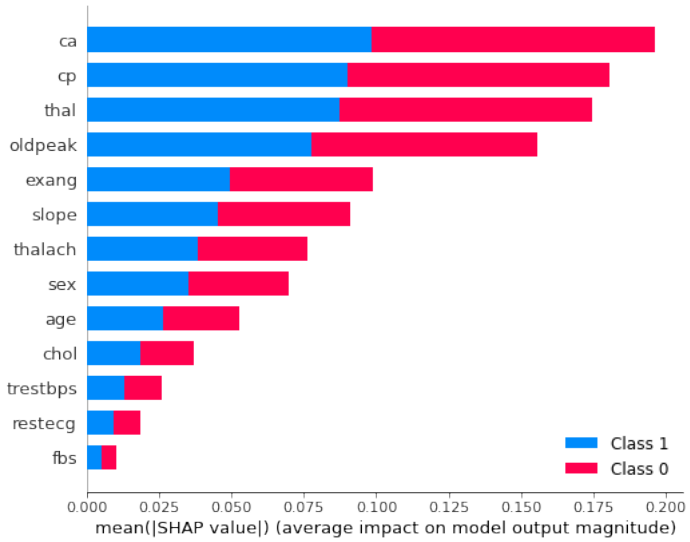
```
Train Accuracy:  1.0
Test Accuracy:  0.8360655737704918

Confusion Matrix:
[[24  5]
 [ 5 27]]

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.83      0.83        29
           1       0.84      0.84      0.84        32

    accuracy                           0.84        61
   macro avg       0.84      0.84      0.84        61
weighted avg       0.84      0.84      0.84        61
```

Random Forest with Feature Importance and SHAP

Random Forest is very similar to decision trees. One major difference between them is that random Forest aggregates each iteration while the former method aggregates after each step. This leads to a regularised fitted line compare to a very aggressive approach in the formal model. Random forest gives us an accuracy of 85%.

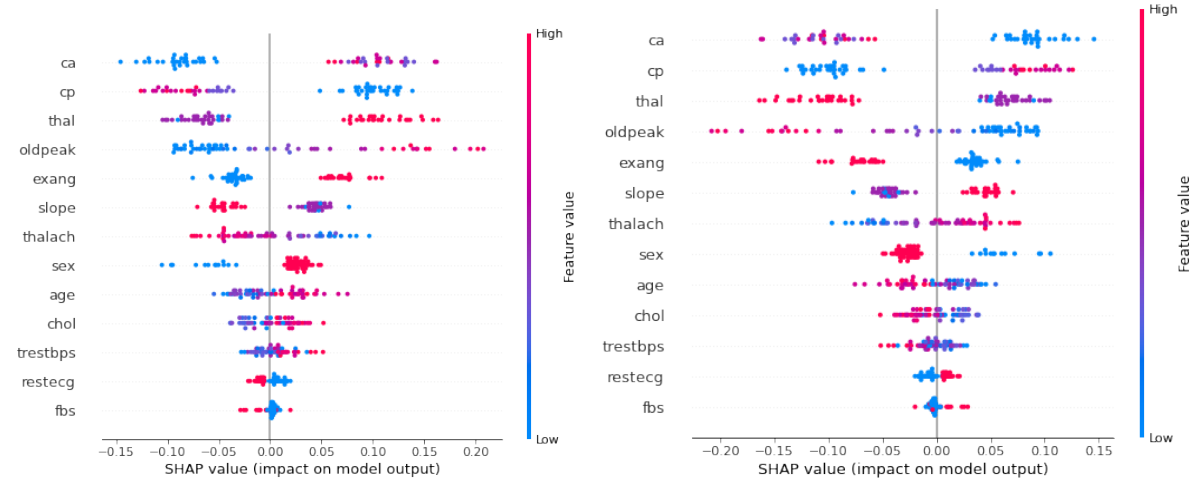


Here we can see that the features that was considered important by the former model of not as important for random for his it however uses this feature is that one of mediocre importance to XGBoost. FBS which was the least significant feature for XG boost is now one of the most important ones relatively.

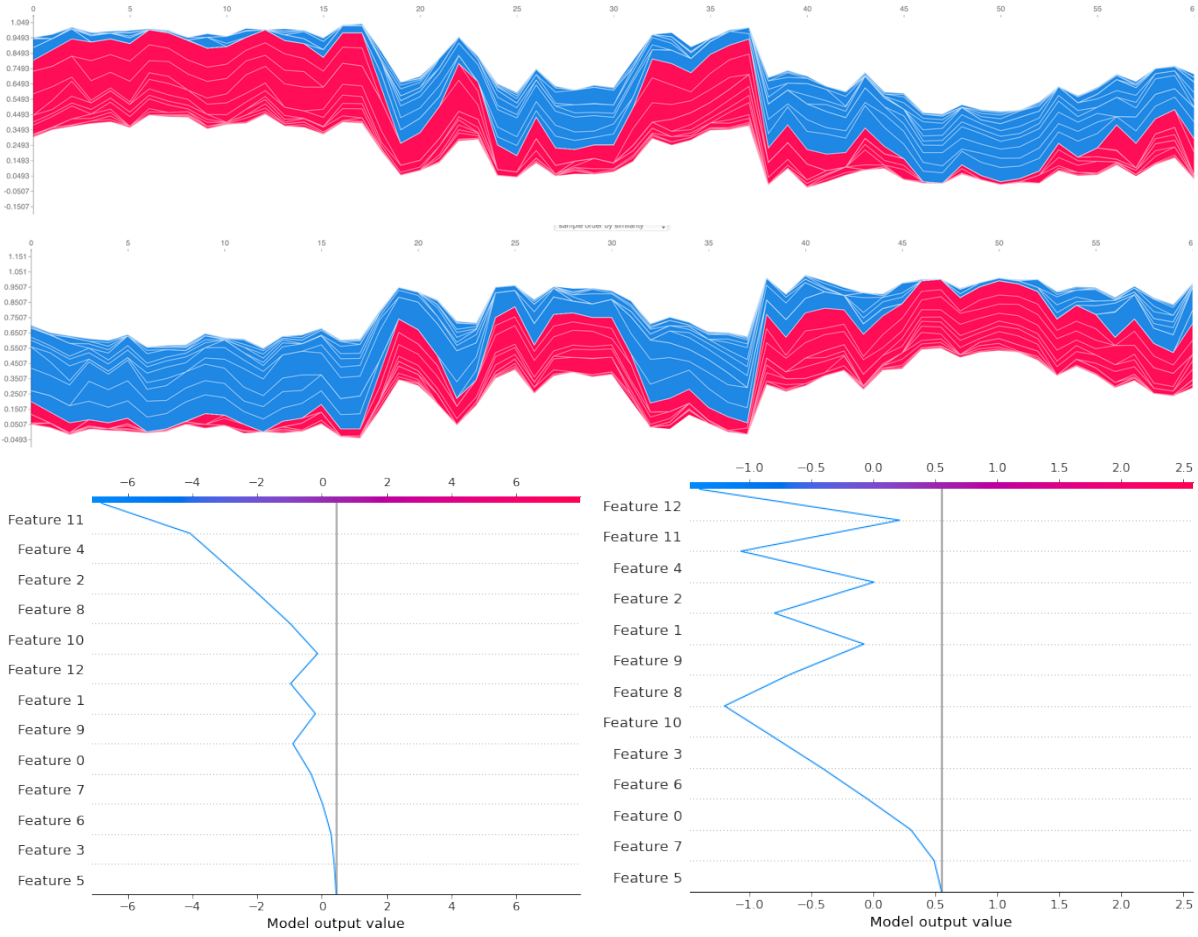


Moving on to SHAP, since the random Forest redressal model hot encodes the output, explainable AI individually for each class. The summary plot above again highlights a change from the original feature importance plot. We can see that certain other features are more important. The images below which I like feature importance based on use and variation over the trading data said reinforce the fact that SHAP various other factors in deciding importance. The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The colour represents the value of the feature from low to high. Overlapping points are jittered in y-

axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance.



The continuous plots below highlight why the default feature importance might misunderstand certain features as important but actually they are not of much significance. The conclusion drawn might be due to the fact that certain features are more significant in the initial part of the training and become less and less significant as training progresses. However, SHAP scores features based on the shapely values for each instance of training.



Conclusion

Concluding on the fact that all the models that we have tried have very similar results since the data is very small. Even then they highlight the strains where random Forest is the best estimator for this dataset as it achieve the best score of 85%. I have also simply normalise the values in a range of 0 to 1 so as to see the variation relatively. The second bar graph highlights the decision tree and XGB have similar performance, while bagging does a little better and random forest is the best among them phone did you see a heart disease dataset