Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Preprocessing Steps

Data was downloaded from the sources given in the assignment. For the 2nd dataset we combined both the CSVs present in the drive folder to create a single data source. All the datasets were renamed as ‘dataset1.csv’, ‘dataset2.csv’, ‘dataset3.csv’ for ease of future access and reference.

Our first goal was to perform EDA on all the datasets to understand their format and get to know more about them. First, we see the sizes of the dataset which can be found as below:-

```
Shape of dataset 1 is (858, 37)
Shape of dataset 2 is (2126, 24)
Shape of dataset 3 is (41188, 22)
```

This helped us understand that 2 of the dataset were comparatively very small and contained less number of samples, whereas the third dataset was a larger one, having in excess of 40,000 samples. Snapshots of the datasets are attached below:

Dataset 1:

	Unnamed: 0	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	...	STDs: Time since first diagnosis	STDs: Time since last diagnosis
0	0	18	4.0	15.0	1.0	0.0	0.0	0.0	0.0	0.0	...	?	?
1	1	15	1.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	...	?	?
2	2	34	1.0	?	1.0	0.0	0.0	0.0	0.0	0.0	...	?	?
3	3	52	5.0	16.0	4.0	1.0	37.0	37.0	1.0	3.0	...	?	?
4	4	46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.0	...	?	?

Dataset 2:

	Unnamed: 0	index	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelera
0	0	0	132	0.006	0.000	0.006	0.003	0.0	
1	1	1	133	0.003	0.000	0.008	0.003	0.0	
2	2	2	134	0.003	0.000	0.008	0.003	0.0	
3	3	3	132	0.007	0.000	0.008	0.000	0.0	
4	4	4	131	0.005	0.072	0.008	0.003	0.0	

Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Dataset 3:

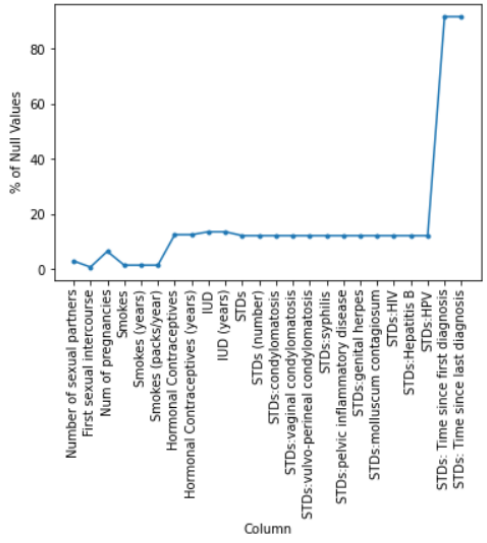
	Unnamed: 0	age	job	marital	education	default	housing	loan	contact	month	...	campaign	pdays	previous	poutcome	emp_var
0	0	44	blue-collar	married	basic.4y	unknown	yes	no	cellular	aug	...	1	999	0	nonexistent	
1	1	53	technician	married	unknown	no	no	no	cellular	nov	...	1	999	0	nonexistent	
2	2	28	management	single	university.degree	no	yes	no	cellular	jun	...	3	6	2	success	
3	3	39	services	married	high.school	no	no	no	cellular	apr	...	2	999	0	nonexistent	
4	4	55	retired	married	basic.4y	no	yes	no	cellular	aug	...	1	3	1	success	

On seeing the datasets, we observed that there were some unnecessary columns like ‘Unnamed:0’, ‘index’ (which might originally refer to the index of the data sample) and needed to be removed. After this, we tried to find what types of values were present in the different columns. The detailed values can be seen using the EDA python notebook. The main observations from here were the following:

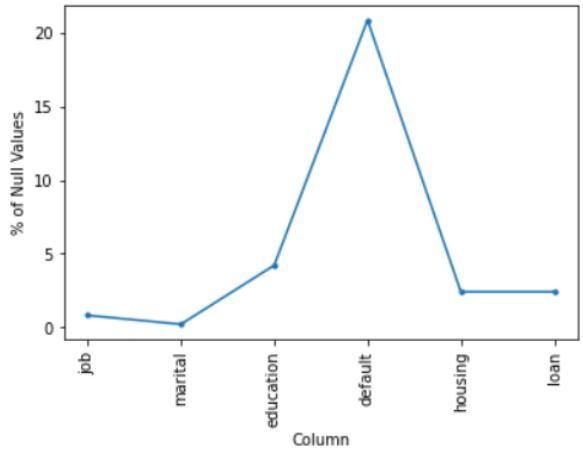
- Dataset 3 had many columns which contained data in form of strings (representing categorical data)
- Dataset 1 had many null values which were represented by a ‘?’ entry.
- Dataset 3 had some null values which were represented by an ‘unknown’ entry.
- The columns `['STDs:AIDS','STDs:cervical condylomatosis']` of Dataset 1 had either a 0 value or a null value (represented by ‘?’).

Thus we removed the above two mentioned columns from the dataset as they did not vary or contribute to the set as they had only a single value (or a none type). Furthermore, to see the pattern of null values, we plotted a graph to see the percentage of null values in different columns. This can be seen below:

Dataset 1:

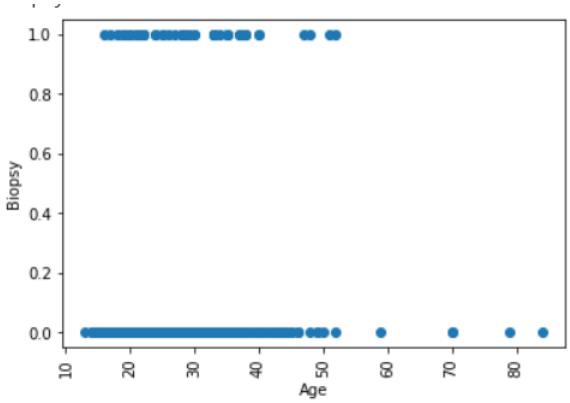


Dataset 3:



We can see that dataset 1 has a lot of columns whose ~15% values are null and dataset 3 also has some columns with null values. More significantly, the columns `['STDs: Time since first diagnosis','STDs: Time since last diagnosis']`of Dataset 1 had in excess of 90% values as null. This means that these columns don't convey any meaning due to the null values, thus we decided to drop these.

Next, we decided to check for the outliers present in the dataset. We make scatter plots for all the features with the target as dependent variable to see if there are any outliers. The plots can be viewed in the EDA python notebook. A sample can be seen below:-



We can see that there are very few samples with age more than 55 and thus these can be considered as outliers in our dataset. On studying each of the column separately, these were the conditions using which the outliers were detected and then removed from all the datasets:

Dataset 1:
- Age>55

Dataset 2:
- baseline value<120 & fetal_health =3
- fetal_movement>0.1 & fetal_health=2
- mean_value_of_short_term_variability>5
- mean_value_of_long_term_variability>25 & fetal_health=2
- mean_value_of_long_term_variability>30
- histogram_max>220
- histogram_number_of_zeroes>6

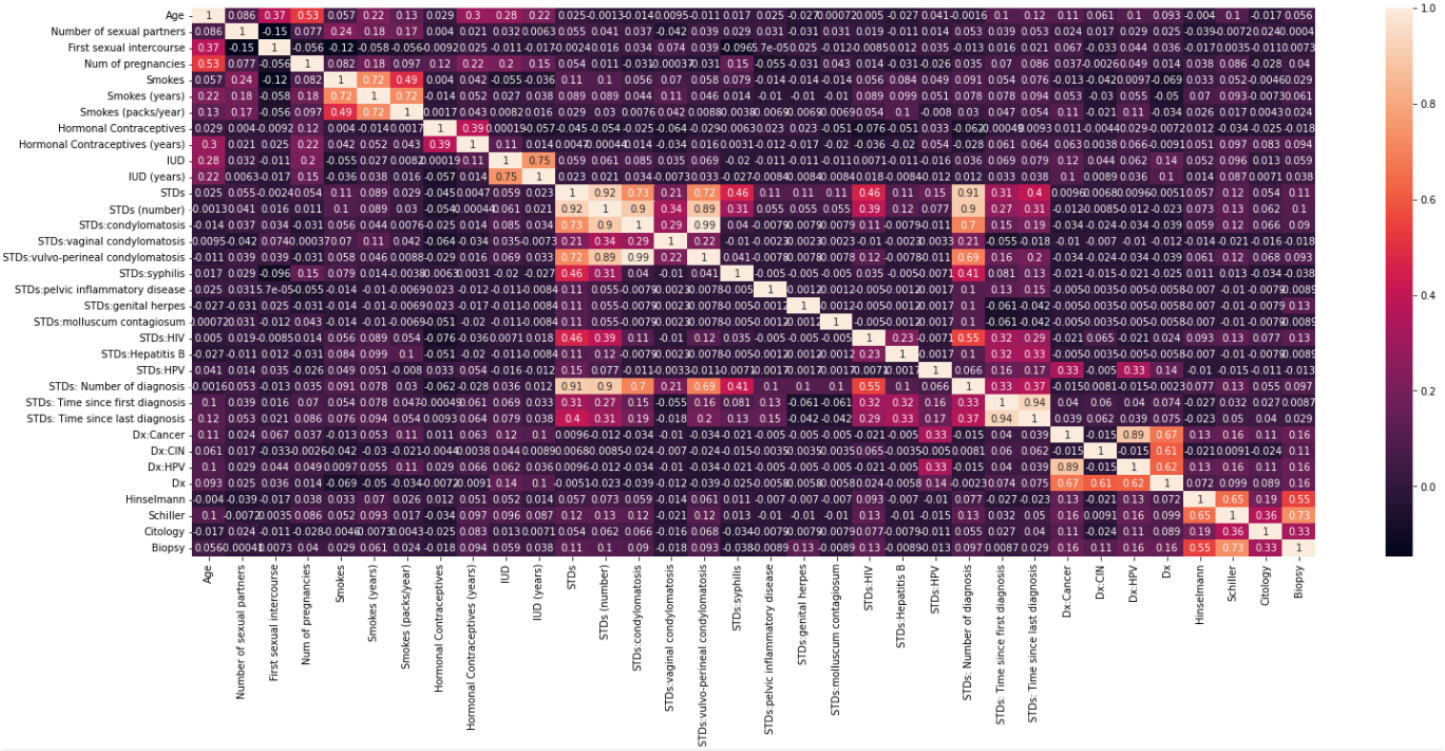
Dataset 3:
- campaign>22 & y=1
- age>95

- duration>=2500
- euribor3m>3.0 & y=1 & euribor3m<4
- campaign>35

Apart from these, we tried to see the pattern of null values across the rows as well and we found that there were some rows in Dataset 3 which had null values for 5 or more columns. In dataset 1, there were several rows which had null values in 14 or more out of the 28 columns. Thus we decided to remove the data samples (rows) from the respective datasets.

For easier representation and code usage, we replaced the '?' and 'unknown' values in dataset 1 and 3 respectively with NaN values supported in python. After this, we try to observe the correlation between the columns of the dataset, inorder to check if any highly correlated columns exist. We do this by generating a heatmap of the correlation values of the columns of the dataset. The heatmaps can be seen below:

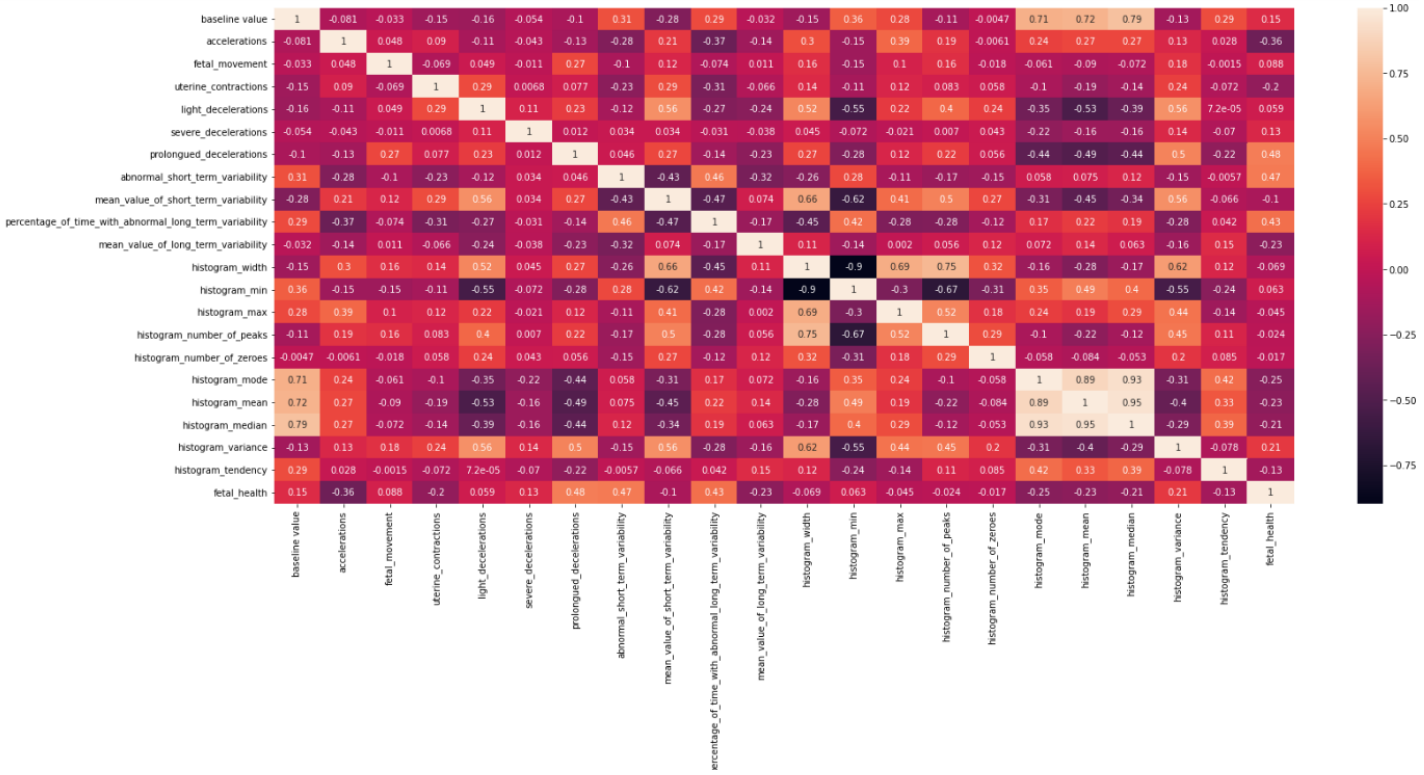
Dataset 1:



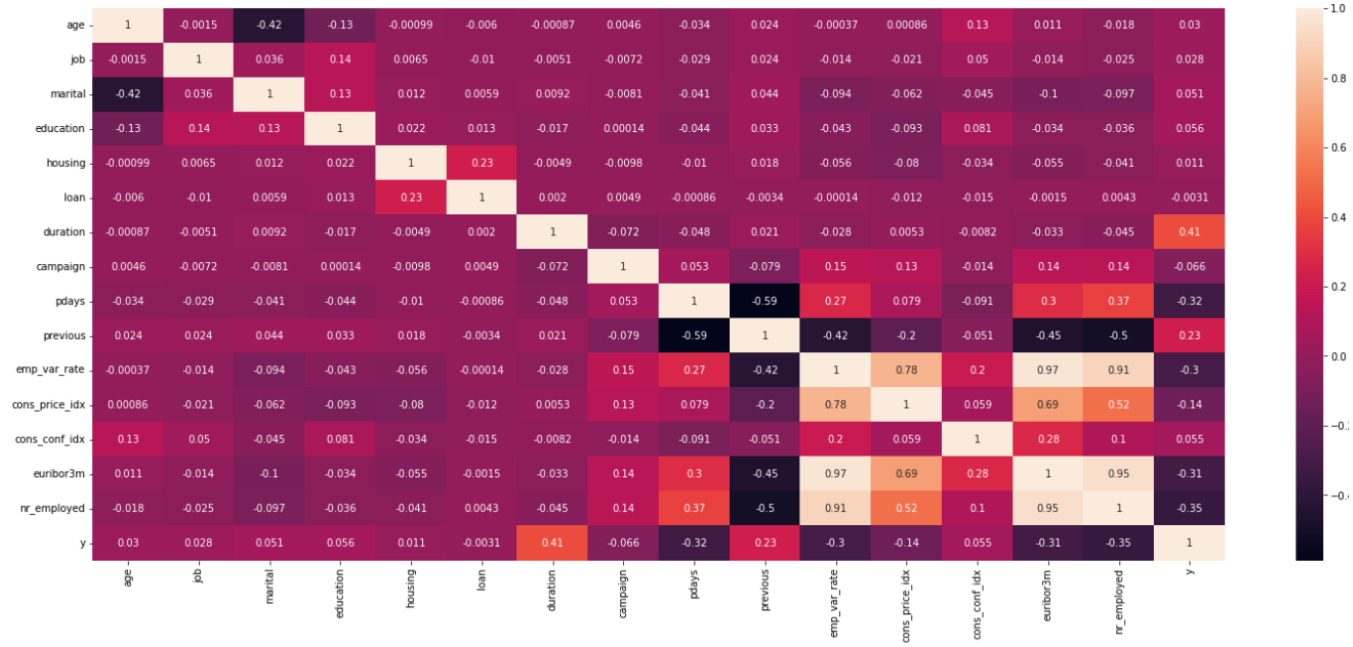
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Dataset 2:



Dataset 3:



After analyzing the heatmaps, we decided to take the threshold value as 0.8. Thus, columns with more than 0.8 correlation score can be removed to help reduce the dimensionality of the database and lead to ease of computation. Based on the above fact, we removed the following columns from the dataset (as they had a correlation score of more than 0.8 with some other column in the dataset):

Dataset 1: `['STDs','STDs (number)','Dx:HPV','STDs:AIDS','STDs:vulvo-perineal condylomatosis']`
Dataset 2: `['histogram_mean','histogram_median']`
Dataset 3: `['emp_var_rate','euribor3m']`

Some of the columns which contained numerical data were present as objects which could lead to some confusions while computation, thus we converted those to appropriate datatypes . Finally, we do a train test split of 80:20 as mentioned in the assignment. This split is done randomly, which helps us to near about same probabilities of occurrence of different target classes in the test and the train split. As we had seen previously, Dataset contained several categorical features. Some of the features had null values of several rows. We replaced these null values with the mode of the each of the features respectively in the training set. These mode values were used to transform the test set as well. For columns containing numerical data, we decided to use mean/median of the training set to update the null values in the corresponding columns (transformed both the train and test set using this).

The categorical data was then processed as we performed one hot encoding fitted on the categorical features of train set and transformed both the train and the test set using that. This lead to change in the dimensionality of the dataset (increase in number of columns). This was done so that, these features could also be processed computationally. We also explore with another method fo the same, where instead of one hot encoding, we convert these to numerical data (without normalizing). Then, for all the non categorical data features, we use the min max scaler approach to normalize the data as the features all belonged to different/varied ranges initially.

Modeling Steps

To model the decision trees, a variety of steps were followed. In the end the implementation of HDTree was used to modify the existing decision tree. Using HDTree, three different splitting functions were introduced, which made sure that the splitting at each node was done using logistic regression instead of any other method. The functions were:

1. **LogisticRegressionSingleSplit**
   This makes sure that for the categorical data, the splitting takes place according to the rules obtained from logistic regression.
2. **TwoQuantileRangeSplit**
   This makes sure that for the continuous data, the splitting takes place according to the rules obtained from logistic regression.
3. **LogisticRegressionDoubleCategorySplit**
   This makes sure that for the 2 attribute splitting, the splitting takes place according to the rules obtained from logistic regression.

Apart from these, multiple other changes were made to make sure that the correct data reaches each node and the splitting is done on the basis of the coefficient and the intercept obtained from the logistic regression model. For the 2nd dataset, which had more than 2 targets, we followed a one versus rest approach where we ran the model three times representing the targets ‘1 vs rest’, ‘2 vs rest’, ‘3 vs rest’. We found the class probabilities for each of the samples in these cases and finally assigned the class with maximum probability for each sample.

The trees were also visualized in each scenario and the nodes clearly mentioned the samples present in each node, the attribute(s) being used for splitting, the number of samples and their labels and how the child nodes are being derived.

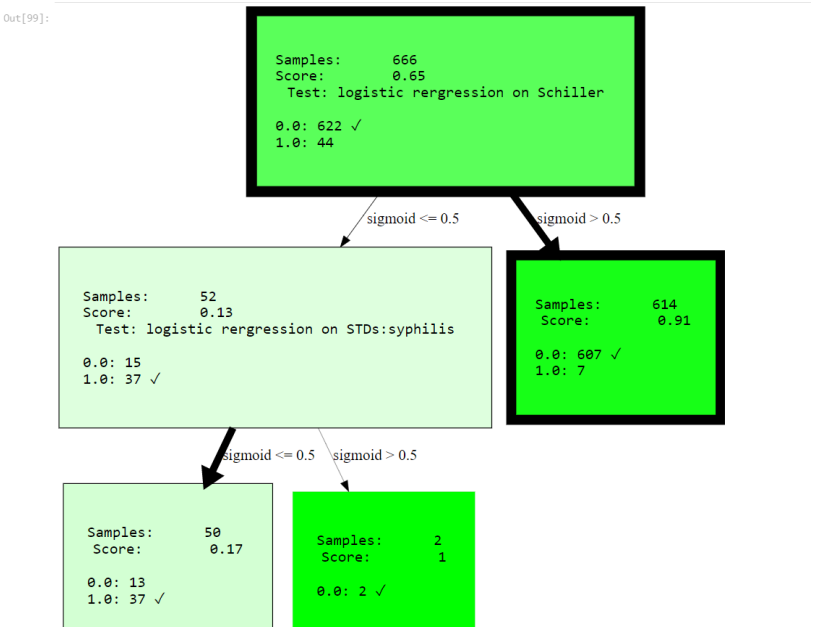
ROC curves and the AUC scores are also generated and the corresponding metrics are also presented for each model. We present the various visualizations and metrics for all the 3 datasets below:-

Dataset 1 (mean replacement of null values) Single Split:
Train Accuracy: 0.96996996996997

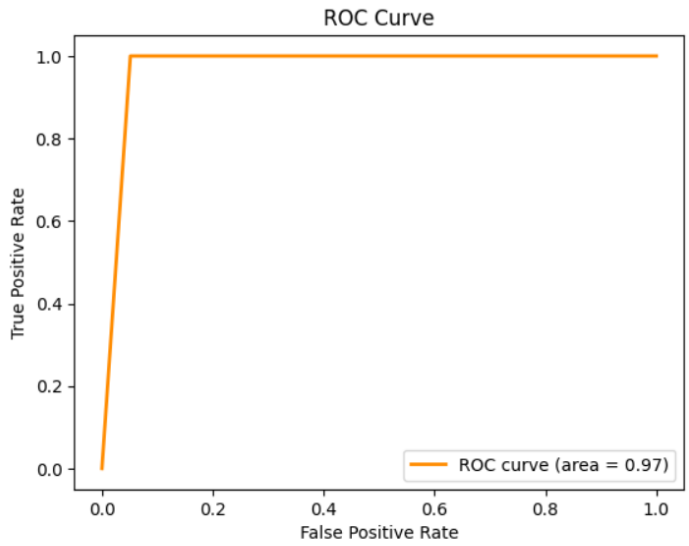
Test Set Statistics:

```
Accuracy =  0.9520958083832335
Macro Precision =  0.7894736842105263
Micro Precision =  0.9520958083832335
Weighted Precision =  0.9722659943271352
Macro Recall =  0.9743589743589743
Micro Recall =  0.9520958083832335
Weighted Recall =  0.9520958083832335
Macro F1 =  0.8535087719298247
Micro F1 =  0.9520958083832335
Weighted F1 =  0.9578527156213887

Classification Report
              precision    recall  f1-score   support

         0.0       1.00      0.95      0.97       156
         1.0       0.58      1.00      0.73        11

    accuracy                           0.95       167
   macro avg       0.79      0.97      0.85       167
weighted avg       0.97      0.95      0.96       167
```

Tree Visualization:



AUC ROC Measure:




```
AUC =  0.9743589743589743
```

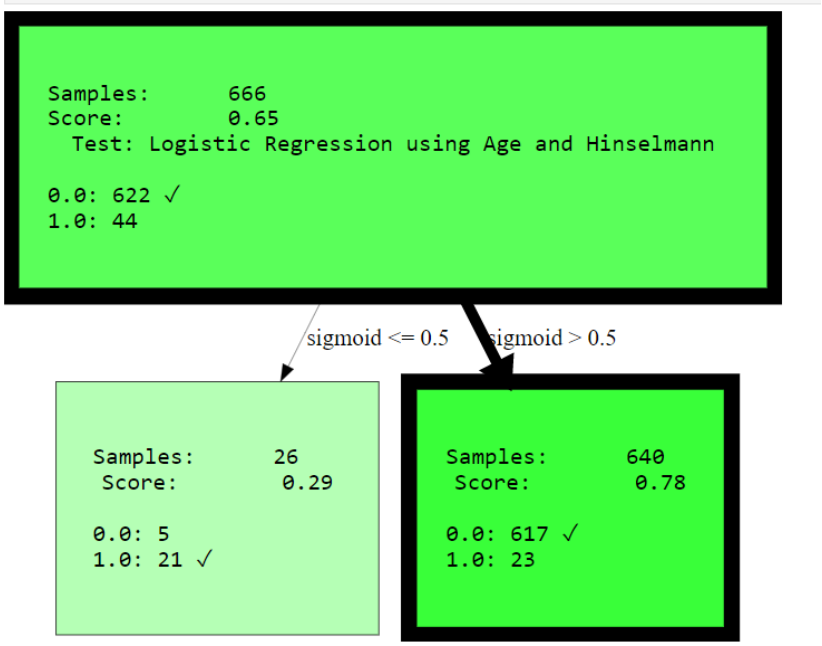
Dataset 1 (mean replacement of null values) Pairwise Split:

Train Accuracy: 0.9579579579579579

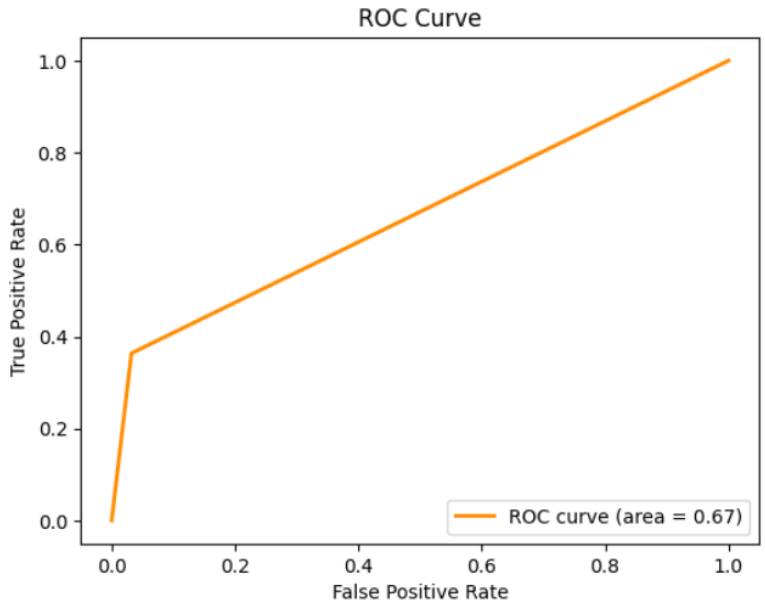
Test Results:

```
Accuracy =  0.9281437125748503
Macro Precision =  0.700070323488045
Micro Precision =  0.9281437125748503
Weighted Precision =  0.9220209370288959
Macro Recall =  0.6657925407925408
Micro Recall =  0.9281437125748503
Weighted Recall =  0.9281437125748503
Macro F1 =  0.6808917197452229
Micro F1 =  0.9281437125748502
Weighted F1 =  0.9247797398832909

Classification Report
              precision    recall  f1-score   support

         0.0       0.96      0.97      0.96       156
         1.0       0.44      0.36      0.40        11

    accuracy                           0.93       167
   macro avg       0.70      0.67      0.68       167
weighted avg       0.92      0.93      0.92       167
```

Tree Visualization:



AUC ROC Measure:



AUC = 0.6657925407925408

Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

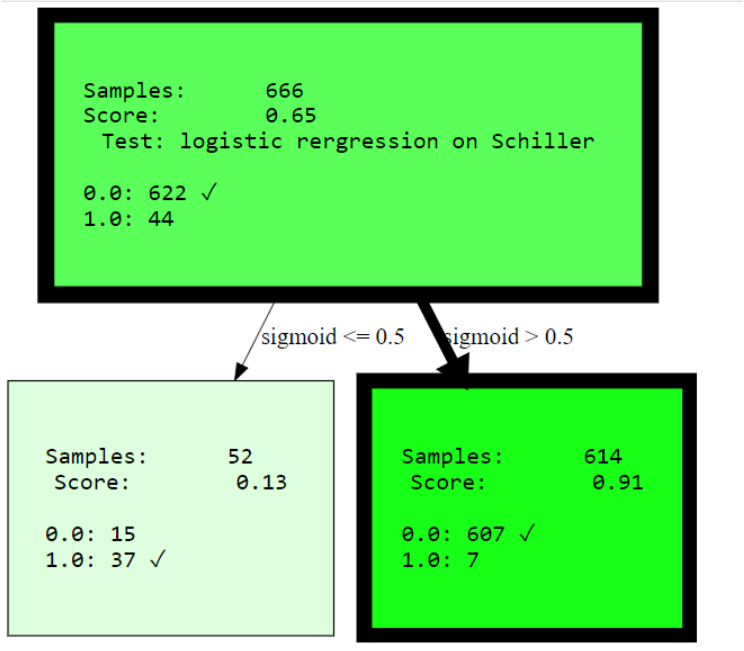
Dataset 1 (median replacement of null values) Single Split:

Train Accuracy: 0.9669669669669669

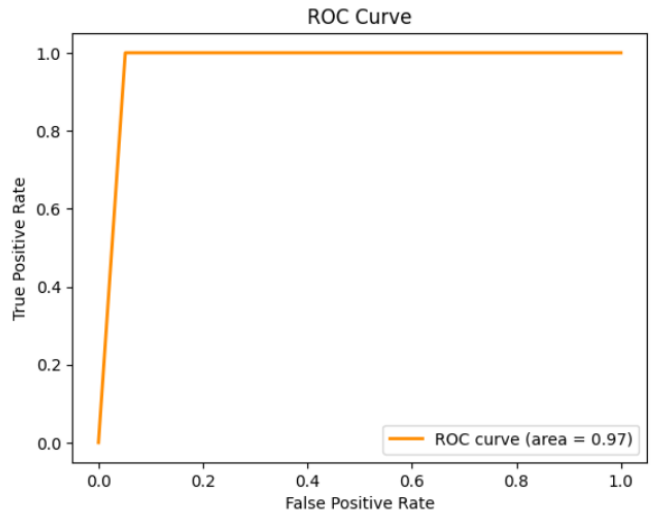
Test Results:

```
Accuracy =  0.9520958083832335
Macro Precision =  0.7894736842105263
Micro Precision =  0.9520958083832335
Weighted Precision =  0.9722659943271352
Macro Recall =  0.9743589743589743
Micro Recall =  0.9520958083832335
Weighted Recall =  0.9520958083832335
Macro F1 =  0.8535087719298247
Micro F1 =  0.9520958083832335
Weighted F1 =  0.9578527156213887

Classification Report
              precision    recall  f1-score   support

         0.0       1.00      0.95      0.97       156
         1.0       0.58      1.00      0.73        11

    accuracy                           0.95       167
   macro avg       0.79      0.97      0.85       167
weighted avg       0.97      0.95      0.96       167
```

Tree Visualization:



AUC ROC Measure:




```
AUC =  0.9743589743589743
```

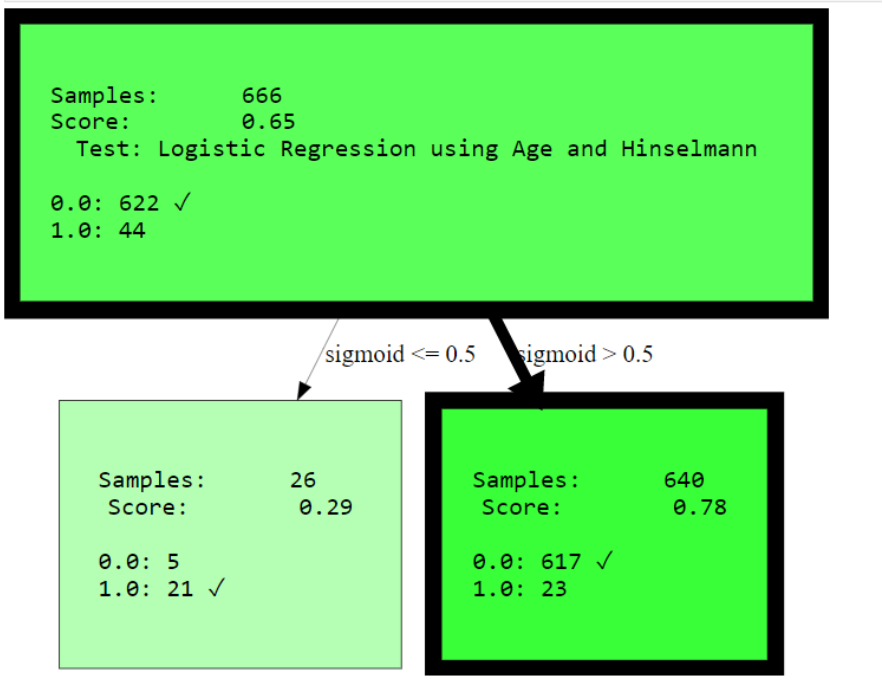
Dataset 1 (median replacement of null values) Pairwise Split:

Train Accuracy: 0.9579579579579579

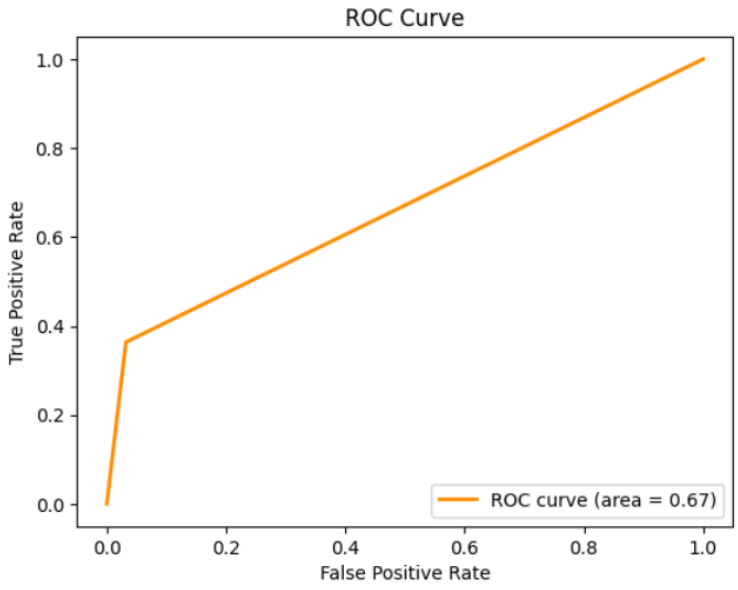
Test Results:

```
Accuracy =  0.9281437125748503
Macro Precision =  0.700070323488045
Micro Precision =  0.9281437125748503
Weighted Precision =  0.9220209370288959
Macro Recall =  0.6657925407925408
Micro Recall =  0.9281437125748503
Weighted Recall =  0.9281437125748503
Macro F1 =  0.6808917197452229
Micro F1 =  0.9281437125748502
Weighted F1 =  0.9247797398832909

Classification Report
              precision    recall  f1-score   support

         0.0       0.96      0.97      0.96       156
         1.0       0.44      0.36      0.40        11

    accuracy                           0.93       167
   macro avg       0.70      0.67      0.68       167
weighted avg       0.92      0.93      0.92       167
```

Tree Visualization:



AUC ROC Measure:




```
AUC =  0.6657925407925408
```

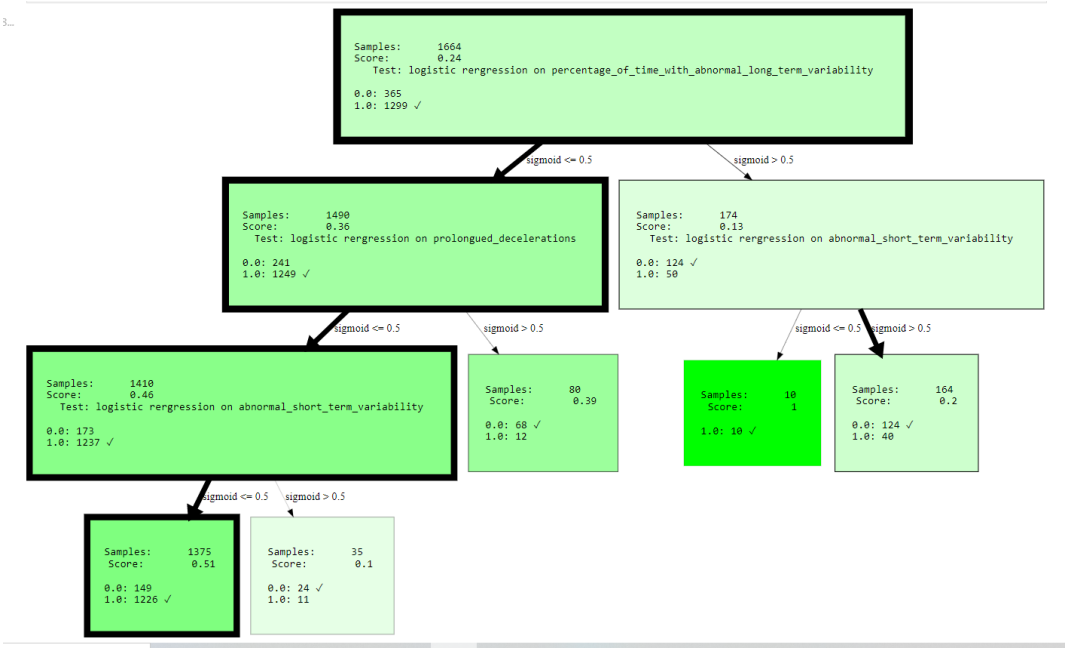
Dataset 2 Single Split:

Train Accuracy for 1 vs Rest: 0.8725961538461539

Test Results for 1 vs Rest:

```
Accuracy =  0.8774038461538461
Macro Precision =  0.8493064920053939
Micro Precision =  0.8774038461538461
Weighted Precision =  0.8719716910184787
Macro Recall =  0.7533976173007721
Micro Recall =  0.8774038461538461
Weighted Recall =  0.8774038461538461
Macro F1 =  0.7870200271043517
Micro F1 =  0.8774038461538461
Weighted F1 =  0.8677317769566677

Classification Report
              precision    recall  f1-score   support

         0.0       0.81      0.54      0.65        87
         1.0       0.89      0.97      0.93       329

    accuracy                           0.88       416
   macro avg       0.85      0.75      0.79       416
weighted avg       0.87      0.88      0.87       416
```

Tree Visualization for 1 vs Rest:



Train Accuracy for 2 vs Rest: 0.8804086538461539

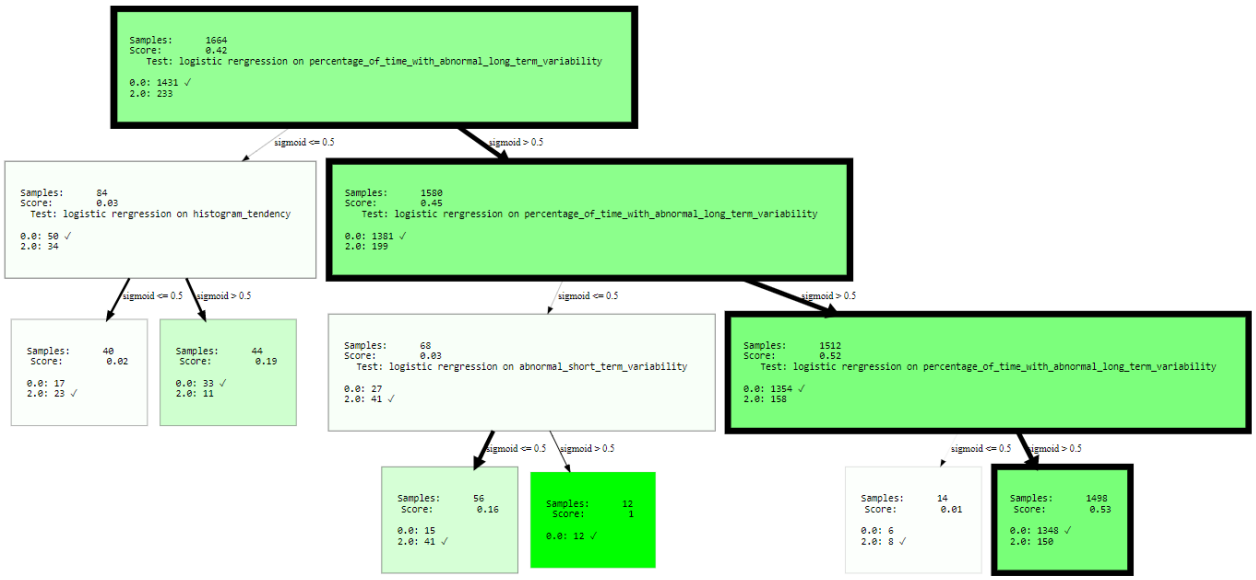
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Test Results for 2 vs Rest:

```
Accuracy =  0.8918269230769231
Macro Precision =  0.7698276682852518
Micro Precision =  0.8918269230769231
Weighted Precision =  0.8749761972769685
Macro Recall =  0.6497252747252747
Micro Recall =  0.8918269230769231
Weighted Recall =  0.8918269230769231
Macro F1 =  0.6853093953300721
Micro F1 =  0.8918269230769231
Weighted F1 =  0.8765066317010438

Classification Report
              precision    recall  f1-score   support

         0.0       0.91      0.97      0.94       364
         2.0       0.63      0.33      0.43        52

    accuracy                           0.89       416
   macro avg       0.77      0.65      0.69       416
weighted avg       0.87      0.89      0.88       416
```

Tree Visualization for 2 vs Rest:

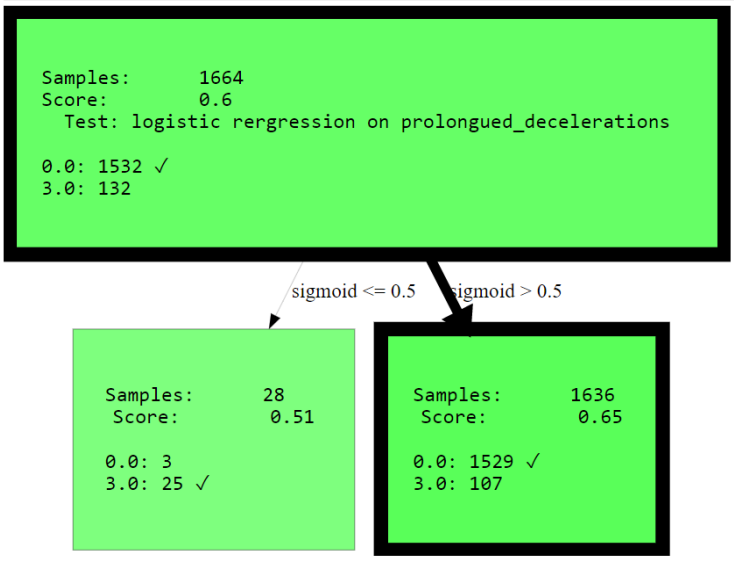


Train Accuracy for 3 vs Rest: 0.9338942307692307

Test Results for 3 vs Rest:

```
Accuracy =  0.9302884615384616
Macro Precision =  0.9646341463414634
Micro Precision =  0.9302884615384616
Weighted Precision =  0.935219277673546
Macro Recall =  0.5857142857142857
Micro Recall =  0.9302884615384616
Weighted Recall =  0.9302884615384616
Macro F1 =  0.6280102371188061
Micro F1 =  0.9302884615384615
Weighted F1 =  0.9069122788500082

Classification Report
              precision    recall  f1-score   support

         0.0       0.93      1.00      0.96       381
         3.0       1.00      0.17      0.29        35

    accuracy                           0.93       416
   macro avg       0.96      0.59      0.63       416
weighted avg       0.94      0.93      0.91       416
```

Tree Visualization for 3 vs Rest:



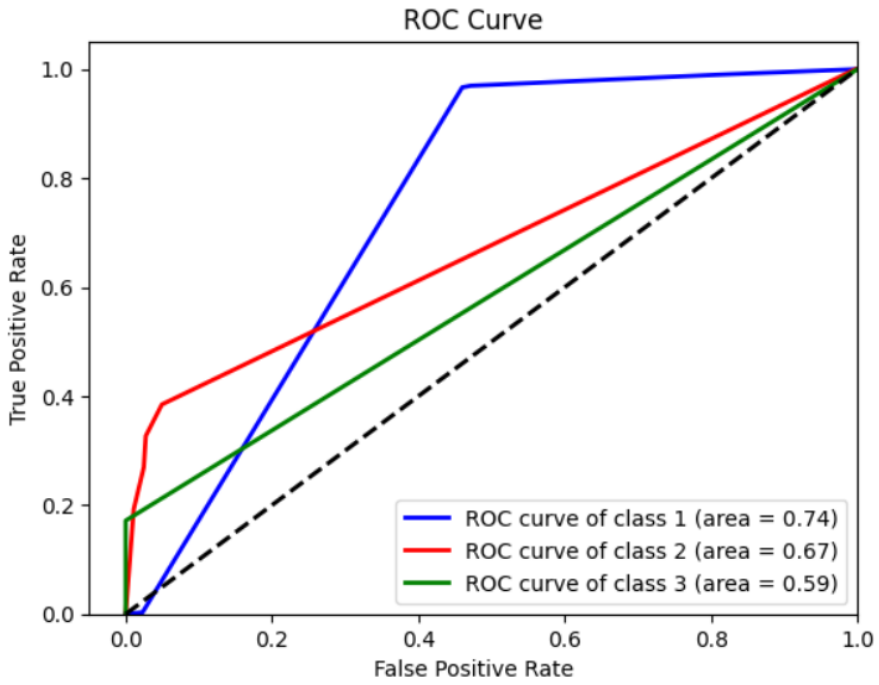
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Final Metrics on test data:

```
Accuracy =  0.8317307692307693
Macro Precision =  0.7869875222816399
Micro Precision =  0.8317307692307693
Weighted Precision =  0.8275401069518716
Macro Recall =  0.4987686072792456
Micro Recall =  0.8317307692307693
Weighted Recall =  0.8317307692307693
Macro F1 =  0.5392826015419083
Micro F1 =  0.8317307692307692
Weighted F1 =  0.8002523207112423

Classification Report
              precision    recall  f1-score   support

         1.0       0.86      0.98      0.92       329
         2.0       0.50      0.35      0.41        52
         3.0       1.00      0.17      0.29        35

    accuracy                           0.83       416
   macro avg       0.79      0.50      0.54       416
weighted avg       0.83      0.83      0.80       416
```

Auc ROC Measure:



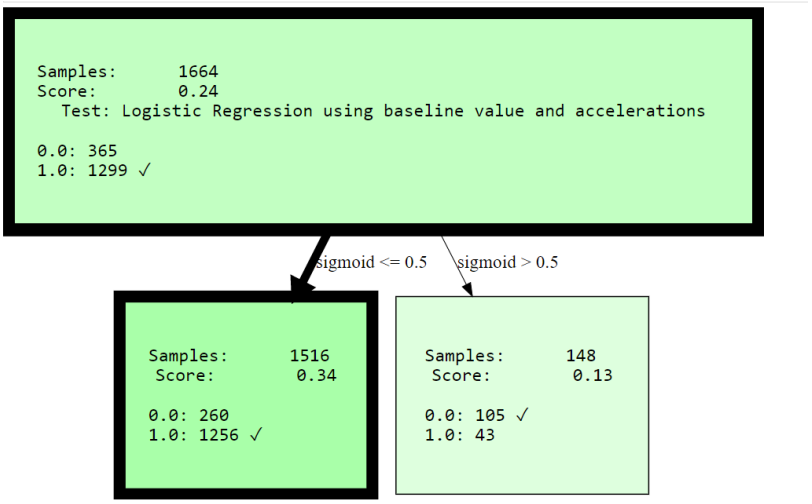
Dataset 2 Pairwise Split:

Train Accuracy for 1 vs Rest: 0.8179086538461539

Test Results for 1 vs Rest:

```
Accuracy =  0.8197115384615384
Macro Precision =  0.7975311490539917
Micro Precision =  0.8197115384615384
Weighted Precision =  0.8119603271094389
Macro Recall =  0.5901023652307585
Micro Recall =  0.8197115384615384
Weighted Recall =  0.8197115384615384
Macro F1 =  0.6040960828352812
Micro F1 =  0.8197115384615384
Weighted F1 =  0.7740600576286175

Classification Report
               precision    recall  f1-score   support

         0.0       0.77      0.20      0.31        87
         1.0       0.82      0.98      0.90       329

    accuracy                           0.82       416
   macro avg       0.80      0.59      0.60       416
weighted avg       0.81      0.82      0.77       416
```

Tree Visualization for 1 vs Rest:

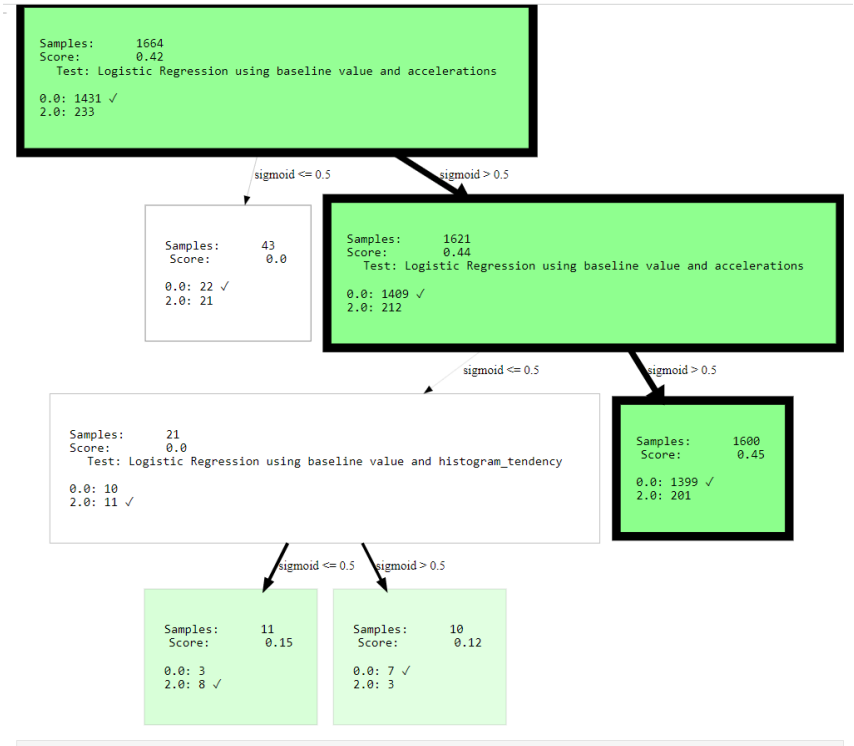


Train Accuracy for 2 vs Rest: 0.8629807692307693

Test Results for 2 vs Rest:

```
Accuracy =  0.8774038461538461
Macro Precision =  0.9385542168674699
Micro Precision =  0.8774038461538461
Weighted Precision =  0.8924698795180722
Macro Recall =  0.5096153846153846
Micro Recall =  0.8774038461538461
Weighted Recall =  0.8774038461538461
Macro F1 =  0.4861336498171337
Micro F1 =  0.8774038461538461
Weighted F1 =  0.8224320003875311

Classification Report
              precision    recall  f1-score   support

         0.0       0.88      1.00      0.93       364
         2.0       1.00      0.02      0.04        52

    accuracy                           0.88       416
   macro avg       0.94      0.51      0.49       416
weighted avg       0.89      0.88      0.82       416
```

Tree Visualization for 2 vs Rest:



Train Accuracy for 3 vs Rest: 0.9338942307692307

Test Results for 3 vs Rest:

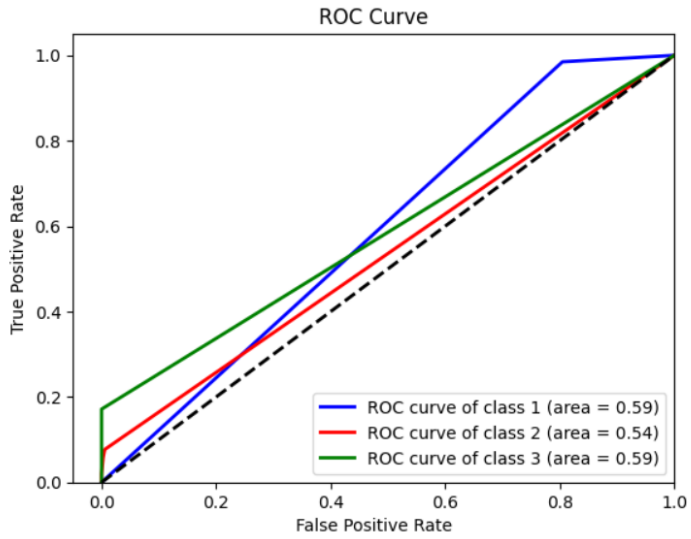
```
Accuracy =  0.9302884615384616
Macro Precision =  0.9646341463414634
Micro Precision =  0.9302884615384616
Weighted Precision =  0.935219277673546
Macro Recall =  0.5857142857142857
Micro Recall =  0.9302884615384616
Weighted Recall =  0.9302884615384616
Macro F1 =  0.6280102371188061
Micro F1 =  0.9302884615384615
Weighted F1 =  0.9069122788500082

Classification Report
              precision    recall  f1-score   support

         0.0       0.93      1.00      0.96       381
         3.0       1.00      0.17      0.29        35

    accuracy                           0.93       416
   macro avg       0.96      0.59      0.63       416
weighted avg       0.94      0.93      0.91       416
```

Tree Visualization for 3 vs Rest:

Final Metrics on test data:

```
Accuracy =  0.8125
Macro Precision =  0.8041152263374486
Micro Precision =  0.8125
Weighted Precision =  0.8015906932573599
Macro Recall =  0.40970695970695975
Micro Recall =  0.8125
Weighted Recall =  0.8125
Macro F1 =  0.43146795013052675
Micro F1 =  0.8125
Weighted F1 =  0.7467600758514011

Classification Report
              precision    recall  f1-score   support

         1.0       0.81      1.00      0.90       329
         2.0       0.60      0.06      0.11        52
         3.0       1.00      0.17      0.29        35

    accuracy                           0.81       416
   macro avg       0.80      0.41      0.43       416
weighted avg       0.80      0.81      0.75       416
```

Auc ROC Measure:



Dataset 3 (Simple numerical conversion of categorical features) Single Split:

Train Accuracy: 0.9070721185977277

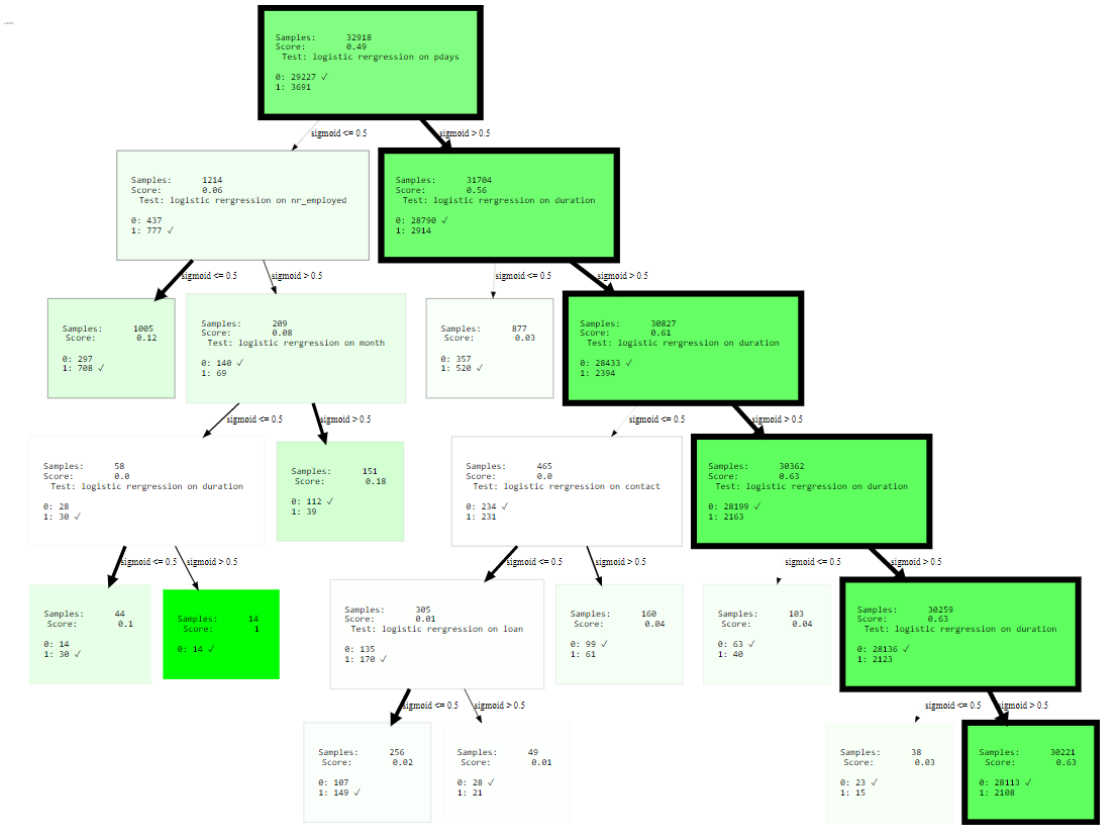
Test Results:

```
Accuracy =  0.9029161603888214
Macro Precision =  0.7765820409898279
Micro Precision =  0.9029161603888214
Weighted Precision =  0.8878497504660305
Macro Recall =  0.6574581842047383
Micro Recall =  0.9029161603888214
Weighted Recall =  0.9029161603888214
Macro F1 =  0.6946287563985198
Micro F1 =  0.9029161603888214
Weighted F1 =  0.8897085877656318

Classification Report
              precision    recall  f1-score   support

           0       0.92      0.97      0.95      7298
           1       0.63      0.34      0.44       932

    accuracy                           0.90      8230
   macro avg       0.78      0.66      0.69      8230
weighted avg       0.89      0.90      0.89      8230
```

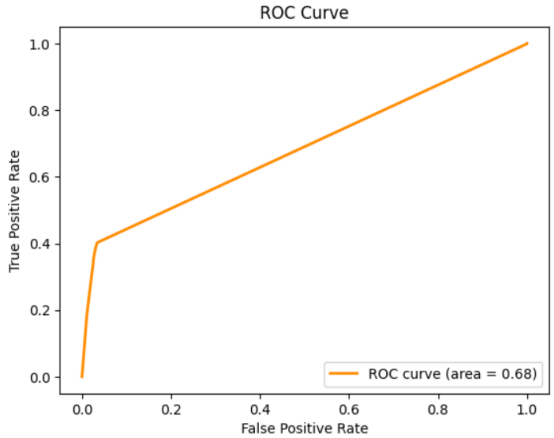
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Tree Visualization:



AUC ROC Measure:



AUC = 0.6849959333911225

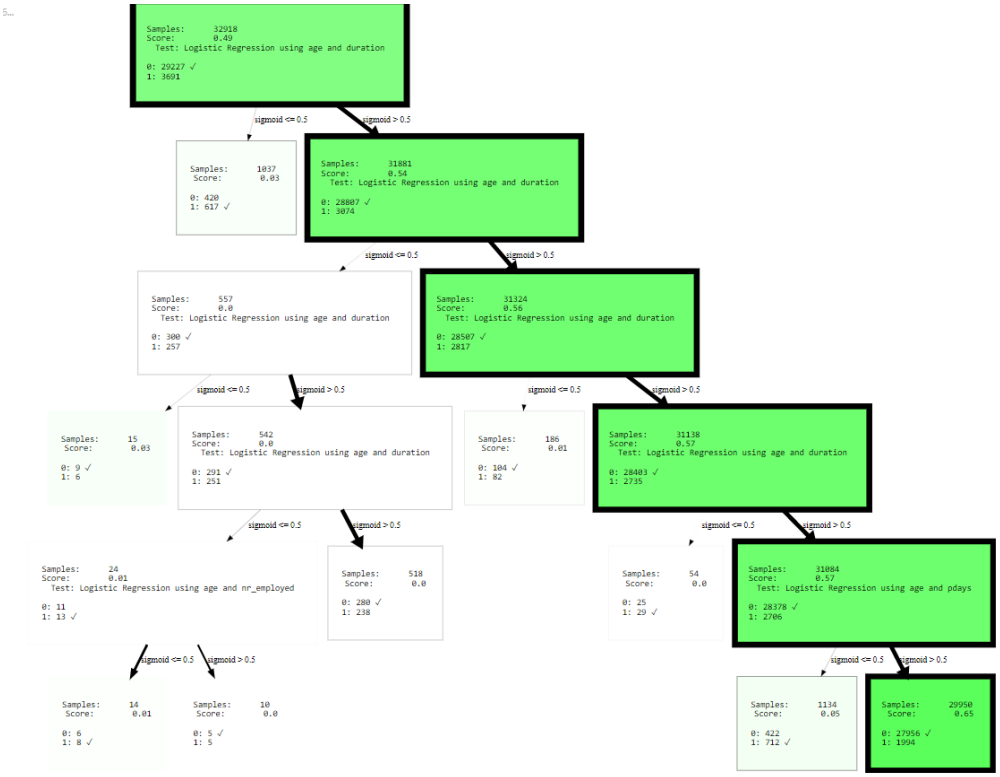
Dataset 3 (Simple numerical conversion of categorical features) Pairwise Split:

Train Accuracy: 0.9028495048301841

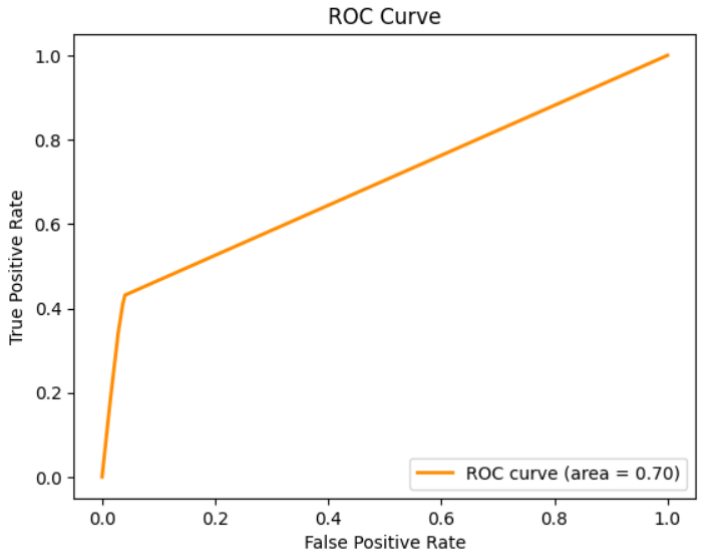
Test Results:

```
Accuracy =  0.8998784933171324
Macro Precision =  0.761001295131828
Micro Precision =  0.8998784933171324
Weighted Precision =  0.884374480123127
Macro Recall =  0.6571492924747446
Micro Recall =  0.8998784933171324
Weighted Recall =  0.8998784933171324
Macro F1 =  0.6911052419078603
Micro F1 =  0.8998784933171324
Weighted F1 =  0.8875360013626972

Classification Report
              precision    recall  f1-score   support

           0       0.92      0.97      0.95      7298
           1       0.60      0.34      0.44       932

    accuracy                           0.90      8230
   macro avg       0.76      0.66      0.69      8230
weighted avg       0.88      0.90      0.89      8230
```

Tree Visualization:



AUC ROC Measure:




```
AUC =  0.69632804037087
```

Dataset 3 (One Hot Encoded Data) Single Split:

Train Accuracy: 0.9068290904672216

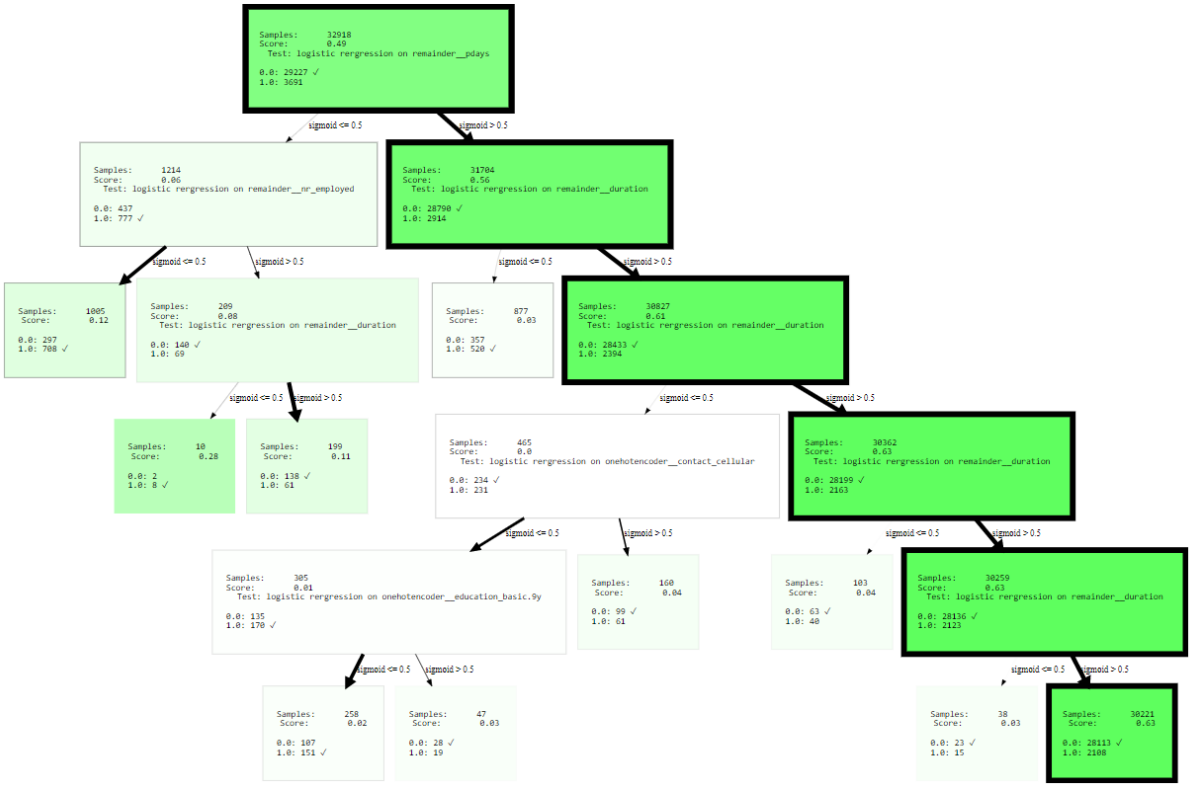
Test Results:

```
Accuracy =  0.903280680437424
Macro Precision =  0.7785033044750134
Micro Precision =  0.903280680437424
Weighted Precision =  0.888308775221033
Macro Recall =  0.6576637199679611
Micro Recall =  0.903280680437424
Weighted Recall =  0.903280680437424
Macro F1 =  0.6951979490901647
Micro F1 =  0.903280680437424
Weighted F1 =  0.890000106113202

Classification Report
              precision    recall  f1-score   support

         0.0       0.92      0.98      0.95      7298
         1.0       0.64      0.34      0.44       932

    accuracy                           0.90      8230
   macro avg       0.78      0.66      0.70      8230
weighted avg       0.89      0.90      0.89      8230
```

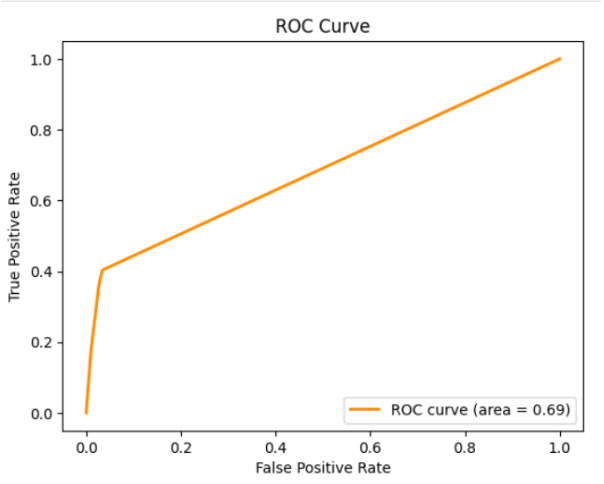
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Tree Visualization:



AUC ROC Measure:



AUC = 0.6859034223027769

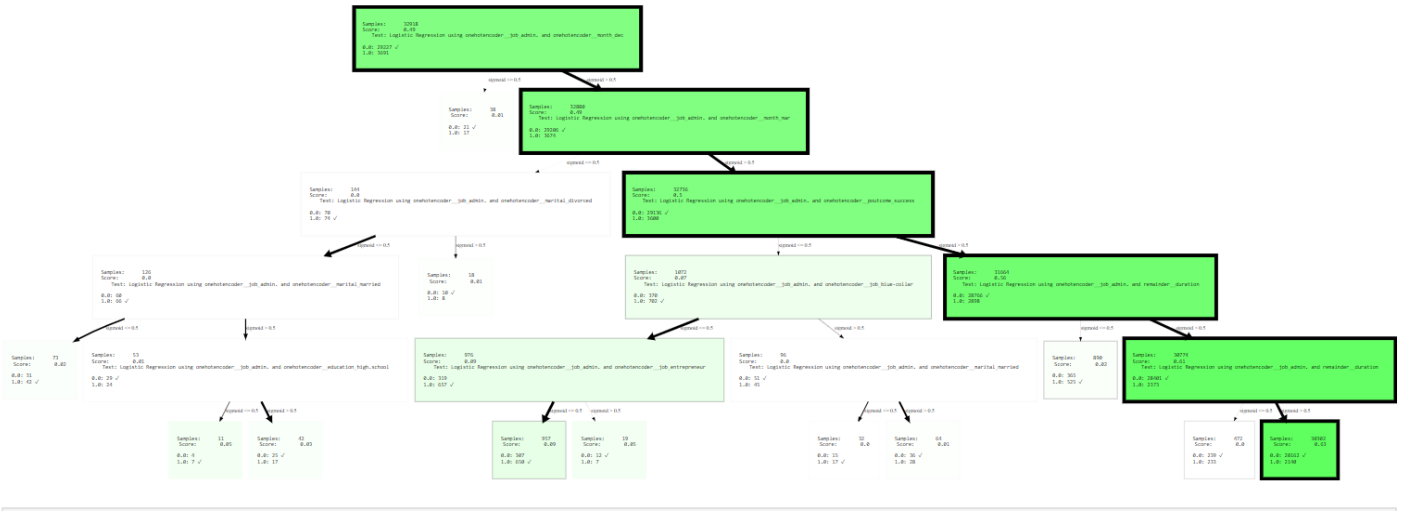
Dataset 3 (One Hot Encoded Data) Pairwise Split:

Train Accuracy: 0.903639346254329

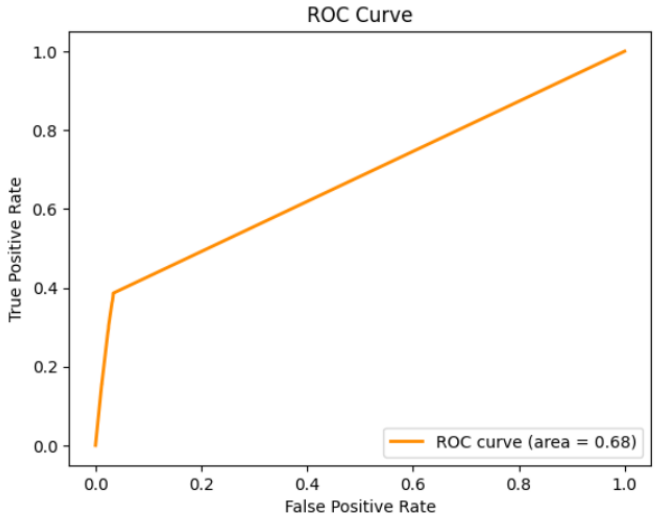
Test Results:

```
Accuracy =  0.8991494532199271
Macro Precision =  0.7633564941752378
Micro Precision =  0.8991494532199271
Weighted Precision =  0.881678144500791
Macro Recall =  0.6389554078547006
Micro Recall =  0.8991494532199271
Weighted Recall =  0.8991494532199271
Macro F1 =  0.6747500857537214
Micro F1 =  0.8991494532199271
Weighted F1 =  0.8837211760535847

Classification Report
              precision    recall  f1-score   support

         0.0       0.92      0.98      0.94      7298
         1.0       0.61      0.30      0.40       932

    accuracy                           0.90      8230
   macro avg       0.76      0.64      0.67      8230
weighted avg       0.88      0.90      0.88      8230
```

Tree Visualization:



AUC ROC Measure:




```
AUC =  0.6767862645653993
```

From the above analysis we can observe that even though single and pairwise split give almost similar results for all the datasets under different conditions, but we see that the recall and f1 of pairwise split is comparatively lesser which is also evident in the auc roc curves. Moreover we see that both mean and median modes of null value replacement yield similar results for dataset 1. We use the median mode of replacement for future analysis. For the third dataset, we see that the one hot encoded data yields slightly better results as compared to simple numerical conversion of categorical data. Since, it is also a more appropriate representation of the data, we use this version of the model fo future analysis.

We also modeled a normal (default parameter) sklearn decision tree for all the three preprocessed datasets. The following results were obtained on the same (next page):-

Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Dataset 1:

Resuts (for mean replacement of null values):

```
Accuracy =  0.9520958083832335
Macro Precision =  0.7934210526315789
Micro Precision =  0.9520958083832335
Weighted Precision =  0.9613614875512133
Macro Recall =  0.8898601398601399
Micro Recall =  0.9520958083832335
Weighted Recall =  0.9520958083832335
Macro F1 =  0.8331668331668332
Micro F1 =  0.9520958083832335
Weighted F1 =  0.9554696800205783
```

Classification Report

	precision	recall	f1-score	support
0.0	0.99	0.96	0.97	156
1.0	0.60	0.82	0.69	11
accuracy			0.95	167
macro avg	0.79	0.89	0.83	167
weighted avg	0.96	0.95	0.96	167

Resuts (for median replacement of null values):

```
Accuracy =  0.9461077844311377
Macro Precision =  0.7759103641456582
Micro Precision =  0.9461077844311377
Weighted Precision =  0.9534544356664821
Macro Recall =  0.8444055944055944
Micro Recall =  0.9461077844311377
Weighted Recall =  0.9461077844311377
Macro F1 =  0.8054368932038836
Micro F1 =  0.9461077844311377
Weighted F1 =  0.9490797046683331
```

Classification Report

	precision	recall	f1-score	support
0.0	0.98	0.96	0.97	156
1.0	0.57	0.73	0.64	11
accuracy			0.95	167
macro avg	0.78	0.84	0.81	167
weighted avg	0.95	0.95	0.95	167

Dataset 2:

```
Accuracy =  0.9302884615384616
Macro Precision =  0.8987363800014331
Micro Precision =  0.9302884615384616
Weighted Precision =  0.9283595991777973
Macro Recall =  0.8269737354843737
Micro Recall =  0.9302884615384616
Weighted Recall =  0.9302884615384616
Macro F1 =  0.8593144560357676
Micro F1 =  0.9302884615384615
Weighted F1 =  0.9278795999082884
```

Classification Report

	precision	recall	f1-score	support
1.0	0.94	0.98	0.96	329
2.0	0.88	0.73	0.80	52
3.0	0.87	0.77	0.82	35
accuracy			0.93	416
macro avg	0.90	0.83	0.86	416
weighted avg	0.93	0.93	0.93	416

Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

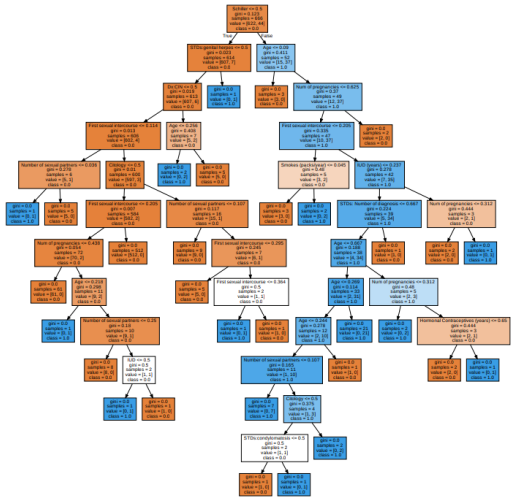
Dataset 3:

```
Resuts (for simple replacement of null values):
Accuracy =  0.8861482381530984
Macro Precision =  0.7168428428320432
Micro Precision =  0.8861482381530984
Weighted Precision =  0.8866283101625154
Macro Recall =  0.7186668227052624
Micro Recall =  0.8861482381530984
Weighted Recall =  0.8861482381530984
Macro F1 =  0.7177488852989543
Micro F1 =  0.8861482381530984
Weighted F1 =  0.8863866515832836

Classification Report
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      7298
           1       0.50      0.50      0.50       932

    accuracy                           0.89      8230
   macro avg       0.72      0.72      0.72      8230
weighted avg       0.89      0.89      0.89      8230
```

```
Resuts (for onehot replacement of null values):
Accuracy =  0.8852976913730255
Macro Precision =  0.7143769332744145
Micro Precision =  0.8852976913730255
Weighted Precision =  0.8851903627965114
Macro Recall =  0.7139755203671534
Micro Recall =  0.8852976913730255
Weighted Recall =  0.8852976913730255
Macro F1 =  0.7141759336395124
Micro F1 =  0.8852976913730254
Weighted F1 =  0.8852439471023805

Classification Report
              precision    recall  f1-score   support

         0.0       0.94      0.94      0.94      7298
         1.0       0.49      0.49      0.49       932

    accuracy                           0.89      8230
   macro avg       0.71      0.71      0.71      8230
weighted avg       0.89      0.89      0.89      8230
```

We also generated visualizations of the different sklearn decision tree classifiers for all the datasets. A preview can be seen below. Since, the visualizations are very big, a clear image can't be attached with the document, thus for complete visualization, please refer to the visualization subfolder.

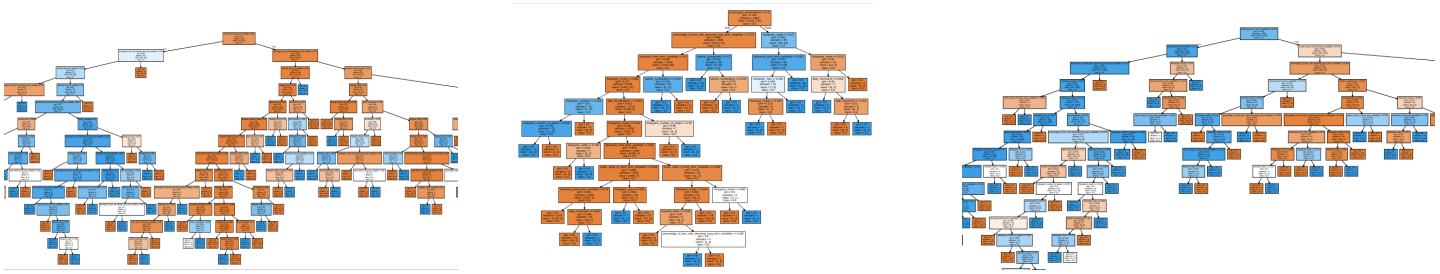
Dataset 1:



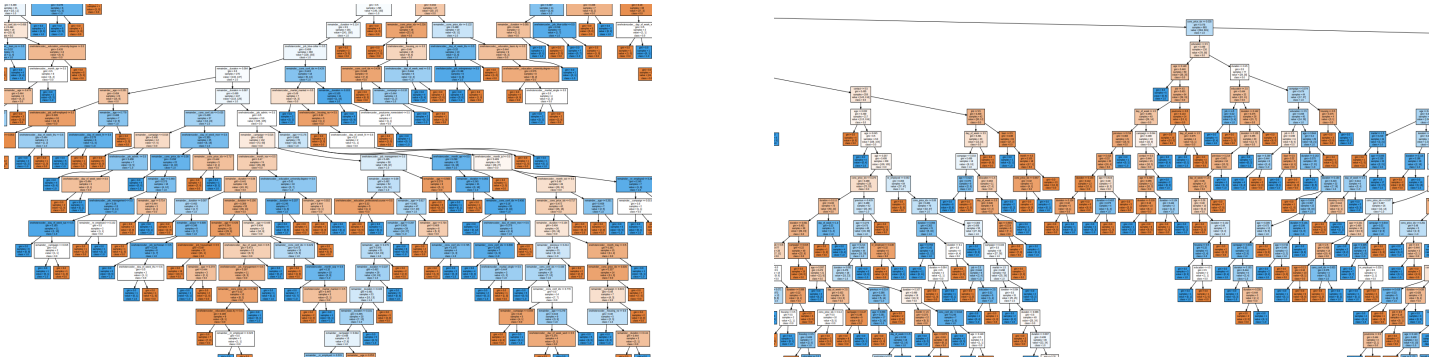
Data Mining Monsoon 2022 Offering
Assignment 1: Classification using Modified Decision Trees

Final Report 9th Oct 2022

Dataset 2:



Dataset 3:



After comparing the results and visualizations obtained from the modified decision tree and the standard sklearn decision tree we can observe that the major difference between the comparison is that the sklearn implementation chooses the attribute with the best entropy measure/ gini index and calculates the information gain for each sample. Based on that, it distributes the data accordingly in the child nodes. In our implementation, the tree does looks for the best attributes for the node, but the internal division of samples take place on the basis of the logistic regression values for the node. The same can be obbserved from the trees given above too. Goal is to apply logistic function with one feature/2 features and check metrics like info gain and decide the best feature to split. The depth of the resulting tree maybe less and the rules more interpretable. As a result the hyperparameters for max depth and the minimum number of samples per leaf have not been finetuned much for each different tree.

Postprocessing Steps

We perform 5 fold cross validation on the for both the single and pairwise split decision trees. The results for the same are as presented below (Average statistics over the folds):-

Dataset 1 (Median replacement of null values) Single Split:

```
Precision:  0.8402636497628009
Recall:  0.9215745544777804
F1:  0.8726566032081594
Accuracy:  0.9639780679604646
```

Dataset 1 (Median replacement of null values) Pairwise Split:

```
Precision:  0.8526173666502894
Recall:  0.7371005338747274
F1:  0.7661645534468129
Accuracy:  0.9507611283457182
```

We observe that the overall accuracy for single attribute split is better. The precision offered by the pairwise attribute split is slightly better but that is at the cost of low F1 and recall.

Dataset 2 Single Split:

```
Precision:  0.7380282494943191
Recall:  0.4893432040428185
F1:  0.5315554623060486
Accuracy:  0.8197115384615385
```

Dataset 2 Pairwise Split:

```
Precision:  0.6303684351344956
Recall:  0.40202834898012385
F1:  0.41584704813210943
Accuracy:  0.7923076923076924
```

We observe that single attribute split performs better overall in terms of all the metrics as compared to the pairwise attribute split.

Dataset 3 (One Hot Encoded Data) Single Split:

```
Precision:  0.7780456056956732
Recall:  0.668827072385594
F1:  0.7049257170217207
Accuracy:  0.9047827505102646
```

Dataset 3 (One Hot Encoded Data) Pairwise Split:

```
Precision:  0.763879372953405
Recall:  0.6569969445431458
F1:  0.6913703612080084
Accuracy:  0.901064507217237
```

We observe that even though there is not much difference between the two models but the single split attributes slightly outperforms the pairwise split model.
Based on the above observations, we find that the single split works better for the given datasets. We also performed different statistical tests to analyze the performance of our models. The given tests were performed:-

- t test
- wilcoxon test
- mann whitney u test
- kruskal wallis test
- chi squared test

For all the tests, we took the same null hypothesis, '*model1 and model2 are the same*'. Here, model 1 represents the single attribute split model and model 2 represents the pairwise attribute split model. The results for the same are reported below:-

Dataset 1 (median replaced):

```
t test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

wilcoxon test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

mann whitney u test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

kruskal wallis test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

chi squared test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis
```

Dataset 2:

```
t test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

wilcoxon test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

mann whitney u test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

kruskal wallis test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

chi squared test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis
```

Dataset 3 (one hot encoded):

```
t test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

wilcoxon test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

mann whitney u test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

kruskal wallis test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis

chi squared test results:
NULL HYPOTHESIS: model1 and model2 are the same
Fail to reject null hypothesis
```

We observe that for all the three datasets, all the tests yielded the same result and failed to reject the null hypothesis. Thus, they interpresented that model1 and model2 are indeed the same. All these tests work on

the null hypothesis that both the models are similar. Given the close results for each of the three models, these tests could not reject the null hypothesis in any of the cases, thus we mentioned that the test faile to reject the null hypothesis. Please note that this does not mean that the models are same, it's just that the tests failed to distinguish between them for the p value < 0.05, that is confidence interval of 10%.

Recreating Results Steps

The models for both our implementation and the ones obtained from the sklearn library are saved in the models folder. They can be loaded in the same object to make the predictions on the dataset.

References

- Medium Article (Andrzej Szymanski, PhD) - Resource Material Provided
- ListenData Article (Mr. Deepanshu Bhalla) - Weight of Evidence
- scikit-learn - user guide for understanding implementation of the different functions
- Medium Article (Mr. Ric. Hard) - Resource Material Provided
- seaborn - user guide for understanding implementation of the different functions
- pandas - user guide for understanding implementation of the different functions
- scipy - user guide for understanding implementation of the different functions
- LogitTree
- Decision Tree From Scratch
- HDTree
- Classroom Slides and Class notes