MSBA Team 7 Homework 1b

Janvier Nshimyumukiza, Ashish Sangai, Sherraina Song, Wenqi Zhai

Introduction to Business Analytics

September 9, 2022

Part 1: Decision Tree in Python

The main steps our team follows for the decision tree in Python is as below:

1) EDA (Exploratory Data Analysis)
2) Model Induction
3) Tuning the Model
4) Model Evaluation
5) Review and Result Discussion

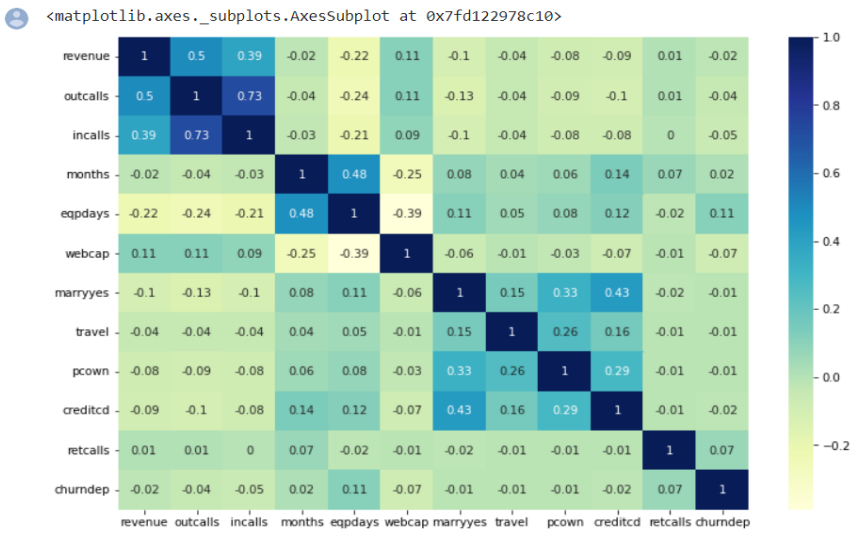
1. EDA (Exploratory Data Analysis)

We explored the dataset before building the model by identifying the data size, null values, missing values, and statistical descriptions. On a high-level overview, the dataset contains 12 variables and 31891 observations. The target variable *churndep* is binary and balanced with 16036 for value 0 and 15855 for value 1. The data is relatively clean since there are no missing values for any variable that needs to be manipulated.

Below are the statistical descriptions(summary table) of our data:

	revenue	outcalls	incalls	months	eqpdays	webcap	marryyes	travel	pcown	creditcd	retcalls	churndep
count	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000	31891.000000
mean	58.665179	24.951385	8.065277	18.761908	391.222633	0.894704	0.363175	0.057163	0.184817	0.676931	0.044088	0.497162
std	44.163859	34.790147	16.610589	9.548019	254.998478	0.306939	0.480922	0.232158	0.388155	0.467656	0.224552	0.500000
min	-5.860000	0.000000	0.000000	6.000000	-5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	33.450000	3.000000	0.000000	11.000000	212.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	48.380000	13.330000	2.000000	17.000000	341.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
75%	71.040000	33.330000	9.000000	24.000000	530.000000	1.000000	1.000000	0.000000	0.000000	1.000000	0.000000	1.000000
max	861.110000	610.330000	404.000000	60.000000	1812.000000	1.000000	1.000000	1.000000	1.000000	1.000000	4.000000	1.000000

We then created a correlation matrix to look at target variables and explanatory variables.

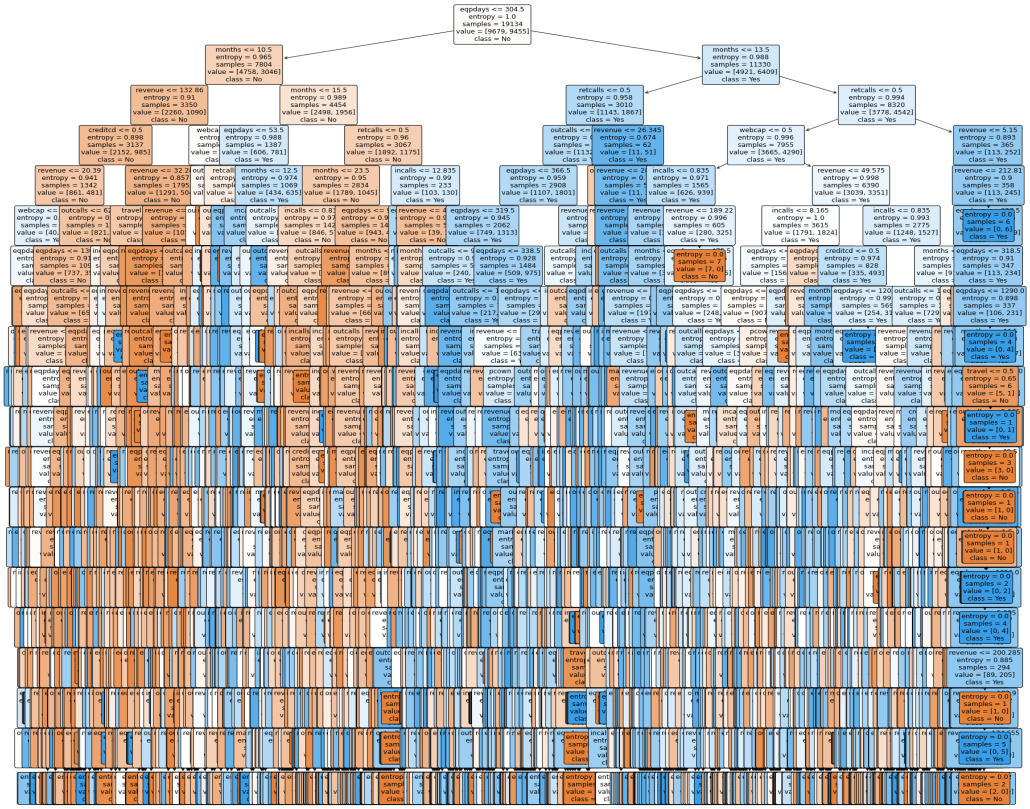


Why is the decision tree an appropriate model for this problem?

According to the correlation matrix, the maximum correlation with the target variable *churndep* is only 0.11, indicating a weak linear relationship. In scenarios like this, we prefer using a non-parametric machine learning algorithm like a decision tree.

2. Model Induction

We generated the tree using the training dataset with *max_depth* set to None and *entropy* as the criterion and anticipated an overfitted decision tree. The predictive accuracy of the model is only around 53.60% which is almost like flipping a coin. We must go through the process of tuning and training the model to give us better predictive statistics.

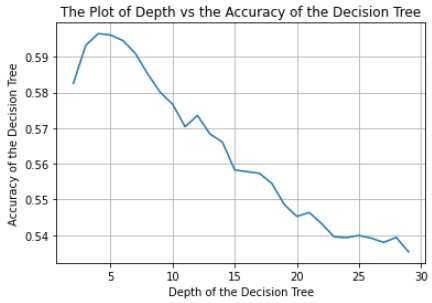


	precision	recall	f1-score	support
0	0.53	0.53	0.53	6357
1	0.54	0.54	0.54	6400
accuracy			0.53	12757
macro avg	0.53	0.53	0.53	12757
weighted avg	0.53	0.53	0.53	12757

3. Tuning the Model

1）Optimizing the **max_depth** parameter

To tune the model, We first iterated the max_depth parameter from 2 to 30 to resolve the overfitting issue for better performance. We also ran the same model with standardised data and there was no improvement in the results. From the graph below, our model shows peak accuracy at around a max_depth of 5.



2）Using **gini or entropy** as the criterion

We chose the cross-validation technique GridSearchCV to find the optimal hyperparameter values: to help us tune the depth of the tree and then select the criterion at the same time to find the model with the best accuracy.

```
parameters = {'criterion':('gini','entropy'), 'max_depth':[1, 2, 3, 4, 5, 10],'class_weight':['balanced']}
```

To tune the model, we created a function with the input of training_table and training_labels, Every combination of Gini/Entropy with Max_depth equals 1, 2, 3, 4, 5, 10 respectively were passed to fit the decision tree model and evaluated accordingly. As a result, after comparing the function returned the model with **Gini as the criterion** and **max_depth set to 5** yielded the best result with an accuracy of 0.60.

4. Model Evaluation

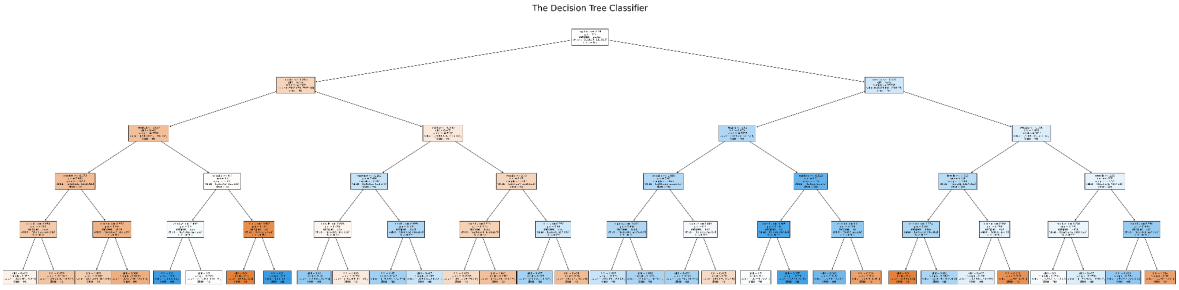
How can we evaluate the predictive ability of the decision tree?

We evaluated the predictive ability of the decision tree with **accuracy, precision, recall, f-score tests, and confusion matrix** and generated the below results.

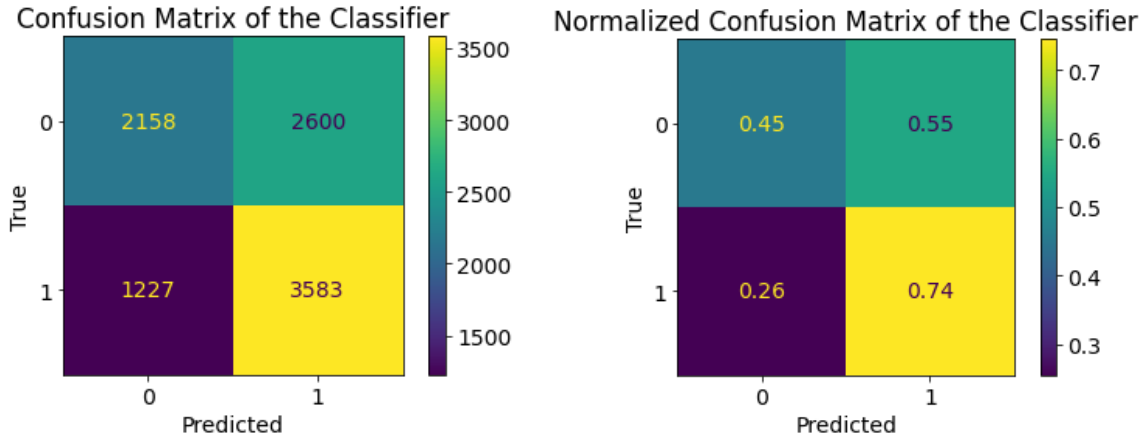
```
The optimal hyperparameters are [class_weight, criterion, max_depth] =  ['balanced', 'gini', 5]
```

The parameters that yield to the best decision tree are:

- Criterion: **gini**
- Max_depth: **5**
- (With random_state: **42)**



#Confusion Matrix (Original & Normalized)



#Performance Statistics (Precision, Recall, F1-measure, Accuracy)

```
The optimal hyperparameters are [class_weight, criterion, max_depth] =  ['balanced', 'gini', 5]
Precisions score:  0.5794921559113699
Recall score:  0.7449064449064449
F1-score score:  0.6518693714181751
Test Accuracy:  0.6000209030100334
```

#Feature Importance

Feature importance refers to the scores assigned to input features based on their usefulness in predicting a target variable, in our project churndep. We also examined the feature importance for all the predicting variables. The result shows that marryyes, travel and pcown do not contribute to the model at all, while eqpdays, months and revenue are the variables contributing to the model the most.

THE FEATURE IMPORTANCES
==========

{'eqpdays': 0.5097375870224564, 'months': 0.2775153561374177, 'revenue': 0.05724097752757458, 'retcalls': 0.05633084002185661, 'incalls': 0.030438734511832787, 'webcap': 0.02790173329620919, 'outcalls': 0.024305994002632867, 'creditcd': 0.01652877748001992, 'marryyes': 0.0, 'travel': 0.0, 'pcown': 0.0}

5. Review and Results Discussion

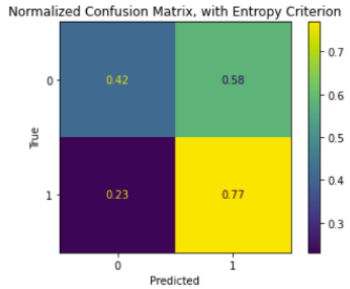
If you build decision trees with different splitting criteria, which decision tree would you prefer to use in practice?

Overall the decision tree model with gini as the criterion and max_depth = 5 has outperformed all the others in terms of accuracy. Comparing the best model with entropy and gini as the criterion and max_depth = 5 when both of the models reached the best performance according to the graph, interestingly even though gini has outperformed the entropy model in terms of accuracy, however, the entropy model shows better performance for predicting the churn.

If we look closely, according to the classification report, The gini model has 0.01 better precision to predict 0 (not churn) than the entropy model. While the entropy model has 0.01 better precision and 0.01 better f1-score to predict the churn group than the gini model and a better f1-score for the positive class 1 (churn group) overall. Precision for the positive class is calculated using TP/TP+FP, and high precision indicates relatively smaller type I error possibilities. The interpretation is that in the business situation when making a type I error is costly, that is to identify users who won't churn as churn users could be a big problem, we should deploy the model with entropy as the criterion in production. For instance, if the action of the prediction is to offer those groups that may churn the promotional price and upgrade for free, doing so for the wrongly predicted churn users could impose higher costs.

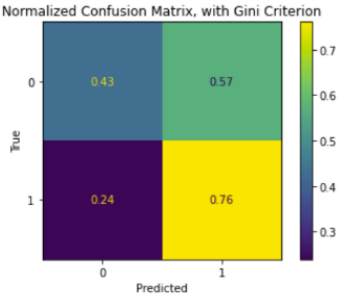
However, if the goal is to maximize the overall accuracy rate, then our team would still suggest sticking to the gini model as the top choice.

#Model Comparison: Gini & max_depth = 5 v.s. Entropy & max_depth = 5



THE RESULTS FOR ENTROPY:
==========

	precision	recall	f1-score	support
0	0.42	0.64	0.51	4146
1	0.77	0.57	0.66	8611
accuracy			0.59	12757
macro avg	0.59	0.61	0.58	12757
weighted avg	0.66	0.59	0.61	12757



THE RESULTS FOR GINI:
==========

	precision	recall	f1-score	support
0	0.43	0.64	0.51	4259
1	0.76	0.57	0.65	8498
accuracy			0.60	12757
macro avg	0.60	0.61	0.58	12757
weighted avg	0.65	0.60	0.61	12757

Part 2: Decision Tree in Rapidminer

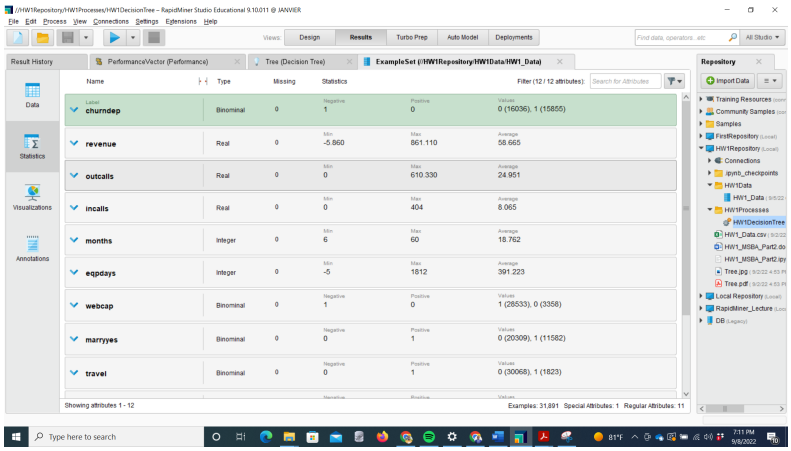
In RapidMiner, we created the process containing the following operators:

1) Import the data
2) Filter Examples
3) Validation
    a) Fitting the decision tree model
    b) Applying the model
    c) Evaluating the performance

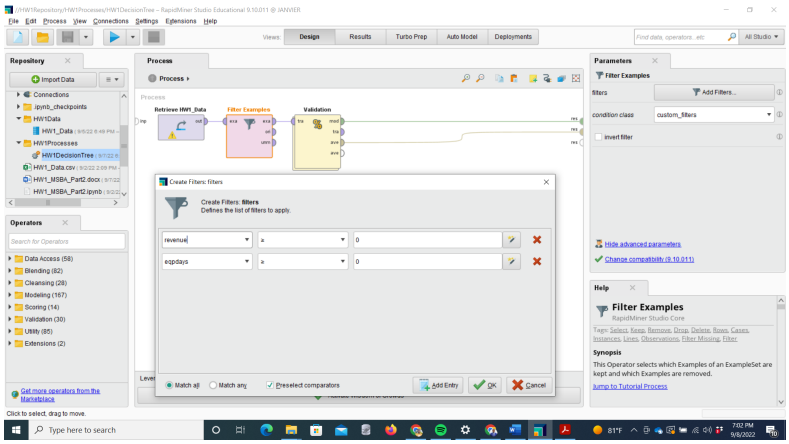
1. Import and filter the data

The first step is to import the data and examine if there are any abnormal data points with each predicting variable. **revenue** and **eqpdays** are associated with abnormal negative values that we considered as noise. The filter for those two variables was created according to the data visualization to eliminate the noise.

#Examine the summary statistics for the input dataset

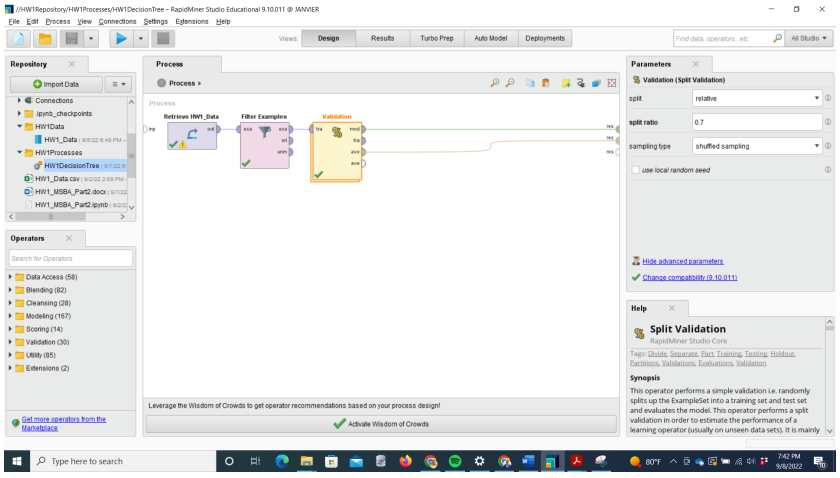


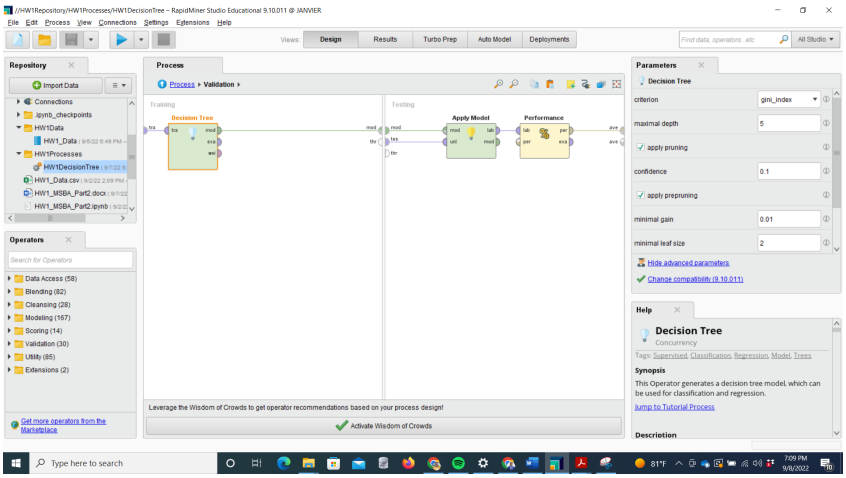
#Create the filter



2. Validation

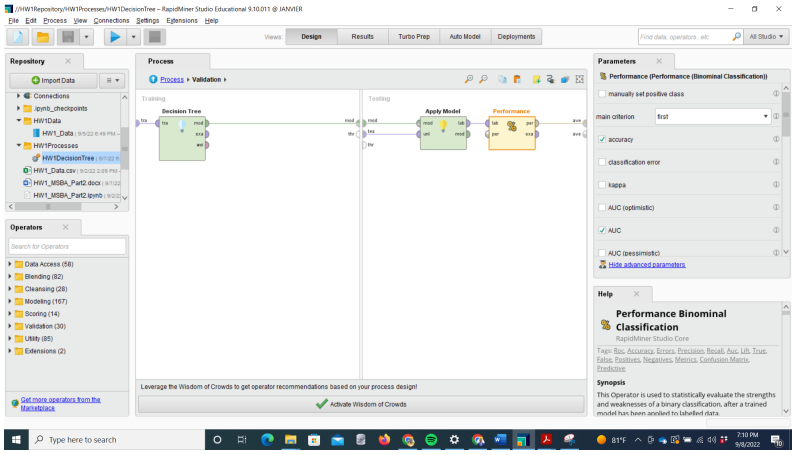
Just like what we did in Python, after comparing different split ratios, the split ratio of 0.7/0.3 gave us the best result. Here we set the split ratio to 0.7.





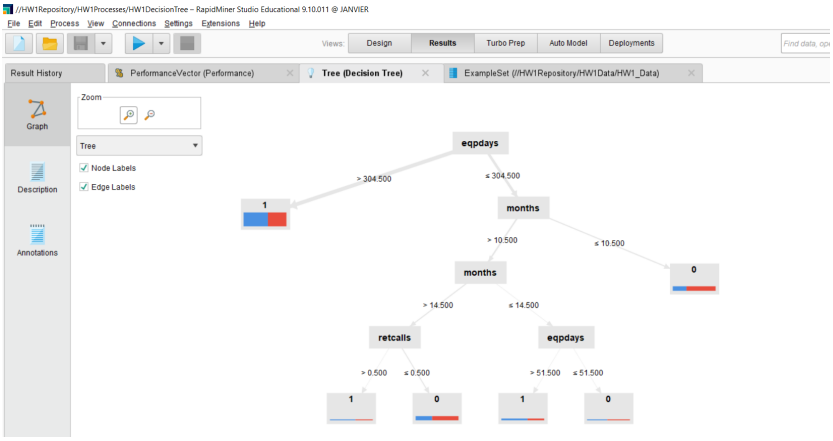
Moving on to setting the parameters for the tree, we have also figured out the best model in Python. So that the criterion was set to Gini, max depth is 5. And we leave all the rest as default.

For performance evaluation, we selected the parameters for model “goodness” including accuracy, AUC, precision, recall and f-measure. We saw the same results as in the Python model we built with the same parameters.



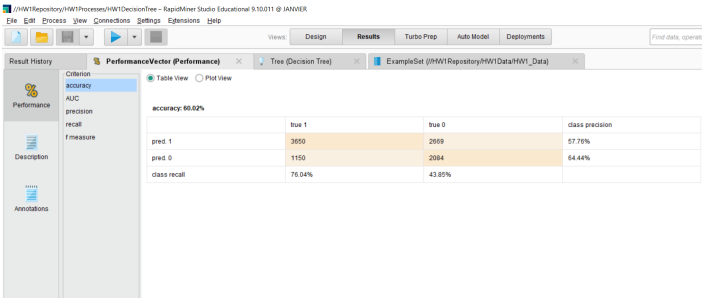
After setting up our model, we then run the process and the plot of the tree was generated along with the evaluation statistics for the model performance.

Part 1 Decision Tree Plot:

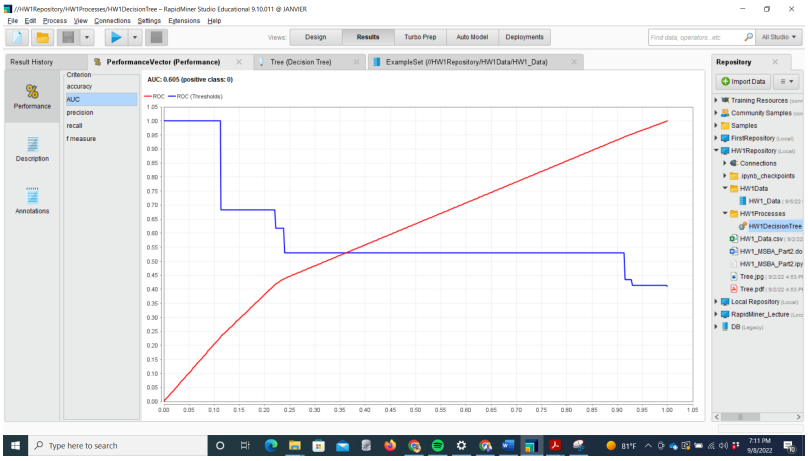


Part 2 Evaluation of the Model:

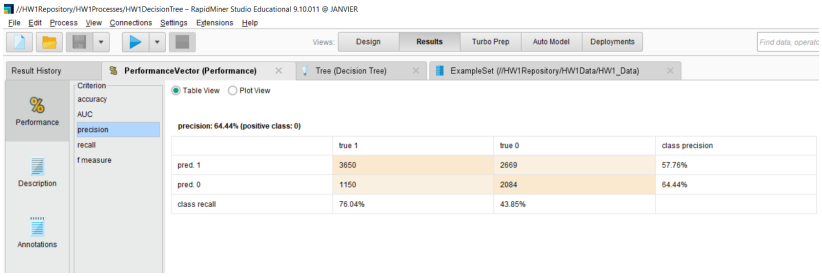
1) Accuracy: overall accuracy = 60.02%



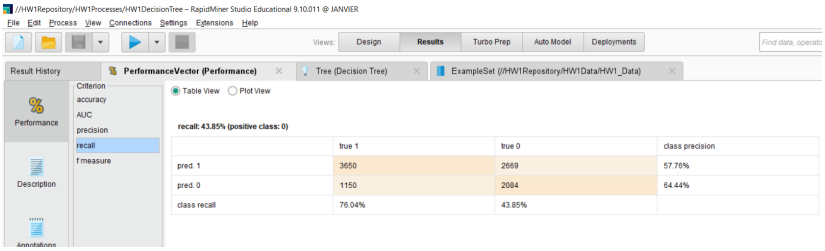
2) AUC & ROC Curve



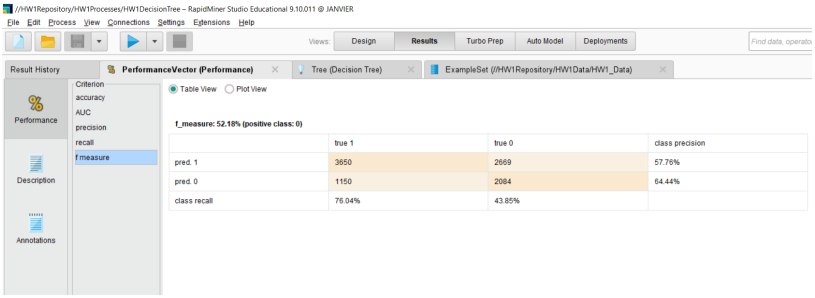
3) Precision



4) Recall



5) F-measure



Conclusion:

For this analysis, we had balanced data. The percentage of positive and negative examples was pretty close in this dataset. Therefore accuracy is a good metric to distinguish the models in this case. In this work, we have compared the accuracy of the models built on two splitting criteria. Gini impurity and entropy. The results have shown that the model is more accurate when it is built with the Gini criterion. Although the entropy-based model exhibits a better precision score, the accuracy is enough to choose the best model for the balanced dataset. With more time and more holistic data, we could improve the model further by testing a wider variety of methods to raise the confidence level.