CSI 5155 Assignment 3 Report

Name: Zhikun (Sam) Yuen Student ID: 300323972

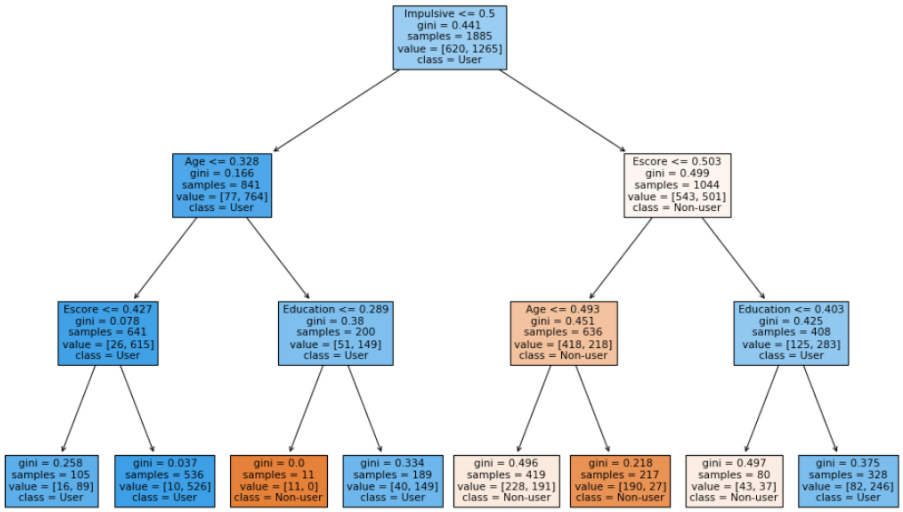
Q1

I select the label "**Cannabis**" as the dataset with decision tree. It is because this label is relatively balance than others. We can have some interesting observation from it. For some extremely imbalance labels, the tree may have depth 1 only and all nodes in the tree are the same class.

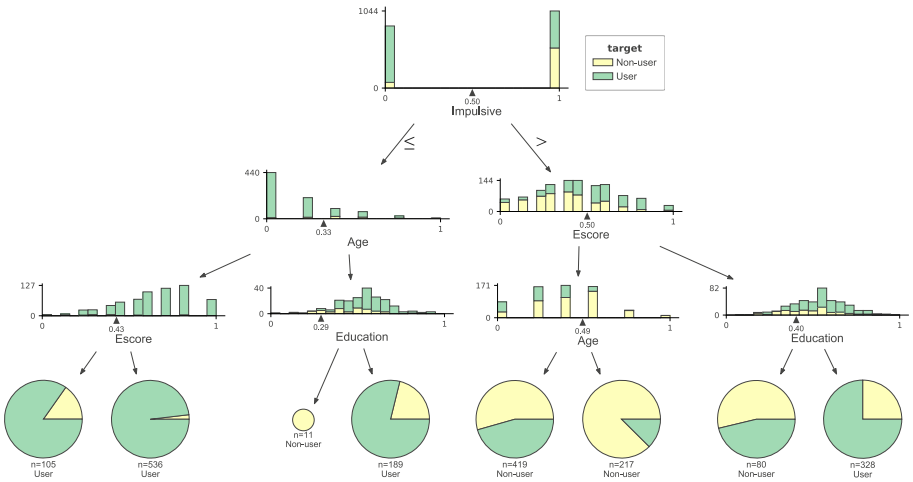
I use two visualizations for the tree. One is from sklearn.tree.plot_tree() and other one is from dtreeviz.trees.dtreeviz().

I use restricted the depth of the decision within depth 1 to 3. It is because when I do grid search from depth 1 to 10, the best depth is 5. However, the difference of the accuracy is just 0.02. I decide to restrict the depth of the tree because a simple model has higher interpretability if the performance is similar.

sklearn.tree.plot_tree():



dtreeviz.trees.dtreeviz()



The two plots show different insights for the tree so I will use both of them to explain the tree.

Q2

In this decision tree, only Impulsive, Age, Escore and Education will be used to make a prediction. The decision tree is use the feature with largest information gain (lowest gini index, according to sklearn's document) to split in every node because this can separate most of the same class data to one side first.

To make a decision, the tree will first look at what is "impulsive" (a binary feature) of the data. From the second plot, almost all training data with feature value 0 is "User", i.e. lower gini index/highest information gain. If the " impulsive" <= 0.5, we will then look at the "age" feature. If the "age" is <=0.328, we will look at its "Escore" (nearly all are "User"). Otherwise, we will look at its "Education". Although the tree also split one more time in "Escore", both path will be predicted as "User" because the majority class in this two split is "User". You can see the ratio from the pipe charts in the second plot. If the age is > 0.328, we need to look at "education", almost all of "non-user" are separated into this path. This is why the model uses a feature to split a node with the lowest gini index. If the education is <= 0.289, the tree will classified it as "non-user". You can see all training data in this leaf is "non-user" and most of the data in the "education" > 0.289 is "User". Again, the split with the lowest gini index will put the features/value with more same class data together, i.e. it carries more information to classify the data.

If the "impulsive" is > 0.5, we will look at its "escore" first. If "escore" is <=0.499, we will look at "age" then. Most of the training data in here is "non-user". Although there is a further split with "age", the majority class in both way is "non-user". The data will be classified as "non-user" in this path (impulsive->escore->age), no matter what value of the age is. If "escore" is > 0.499, most of the data is "User" and we will then look at "education". If "education" <=0.403, the data will be classified as "non-user". Otherwise, it will be classified as "User".

Q3

The constructure of the decision tree is base on an algorithm with gini index as the information gain measure. During building the tree, the model compute the gini index as the impurity measurement for each features in the data, and then use the gini index to compute the information gain and select the features with the highest information gain. This is because the lower gini index/higher information gain the feature is, the more information that feature carries, i.e. we can separate the most data with the same label together. The algorithm stops until some stop criteria, e.g. maximum depth or all data is split well.

During the test stage, given an unseen/test data, the decision tree will follow the splitting criteria in each node to make a decision, just like a binary tree, until reach a leaf node. Then, the algorithm will base on the majority class of the training data in that leaf to predict the data as the majority class. In training and test a decision tree, the algorithm follows these rule. This is why the algorithm didn't do something else.

However, using the majority class in the leaf node to predict the data is not always correct. For some cases, the ratio of two classes is very close, i.e. 50% vs 50%. Always predicting the majority class makes the performance become lower. A better way is to make a prediction based on the probability of classes in the leaf. In this case, given a same data, the decision of the model is different in every prediction.

Q4

If we look at the leaves of the tree, we can see some leaves have similar ratio for user vs non-user, e.g. 46% user vs 54% non-user (impulsive>0.5 -> escore<=0.503 -> age<=0.493). The ratio of user and non-user is really close, but the leaf is "Non-user". It is very likely that a "User" data is classified as "Non-user" in this leaf. In this case, our algorithm is failed. (impulsive>0.5 -> escore>0.503 -> Education > 0.403) is also similar to this cases.

For other leaves, the majority class occupies ¾ of the ratio. It is very high chance that a data is predicted as the majority class is correct. At this case, our model is succussed.

Also, all the above assumption is based on no overfitting on our decision tree model. Decision tree model has very high variance so overfitting is a very common problem in it. If the data distribution on train data and test data is not the same, the model cannot predict well on the unseen data. In this case, our model is also failed.

Q5

The model should work well in a unseen test data first. Also, "User" class is more important in this case. I need to make sure the performance, e.g. recall, precision, F1 and accuracy of "User" class is good. I will also check the features is make sense to classify whether a person is a user or non-user of "**Cannabis"** with some past data in survey or expert knowledge. E.g. using "age" and "education" to predict the data is very make sense because these are some important factors for people with drug addiction, e.g. teenage or low education. Also, comparing the splitting value with the past survey or academic papers, if the split value matches the value on research or survey, then the model is trustworthy.

Q6

In some splits, e.g. (Impulsive->age->escore), both ways make the same prediction. Actually, we can stop the split because no matter what the value of that feature is, the prediction is the same, i.e. the performance of the tree will not be changed if we . We can prune the tree of these nodes, this can make the tree even predicts faster but no changes in the performance. Also, we can use a test data to prune the tree, because decision can be overfitting very easily. Pruning the tree can solve this problem and improve the performance in unseen data.

Also, we can use bagging and boosting approaches to reduce variance and bias for decision tree. They are the variants of decision tree, random forest and Adaboost. They usually have better performance than single decision tree.