**DM Project 2 : Classification models**

1. **Dataset:**

**Handmake data**:

*Sandwich\_easy*: 6 rules, one constrint per each, ordered data

*Sandwich*: 8 rules with at most two constraints per rule

Rules can be found in sandwich\_data\_rule.csv

11 features with a binary classified label (if the sandwich is edible or not)

Each data has 2000 rows for training, 500 rows for testing

You can adjust data generation with datagen code

**From Kaggle:**

*Heart:* Whether the patient has heart disease or not

13 features, train 242 rows and an addition 20% split to test

The features, **cp** and **thal**, are unordered categorical data

<https://www.kaggle.com/ronitf/heart-disease-uci>

*Bank:* Predict whether the client will subscribe a term deposit

16 features, train 45211 rows and test 4521 rows (10% split)

Mostly categorical feature data

<https://www.kaggle.com/sonujha090/bank-marketing>

1. **Result:**

Here I use 4 kinds of models: **Decision Tree, Decision Tree with pruning, KNN, Naïve Bayes.**

Model constructed by calling package from sklearn

Code is written in dm\_hw2.py (for DT and DT pruned), dm\_knn\_hw2.py, dm\_bayes\_hw2.py

Pruned is determined whether the minor class data is below 2% of training data, if true, then stop the pruning.

You need to modified code to get different dataset good to work.

Two charts for accuracy and AUC evaluation

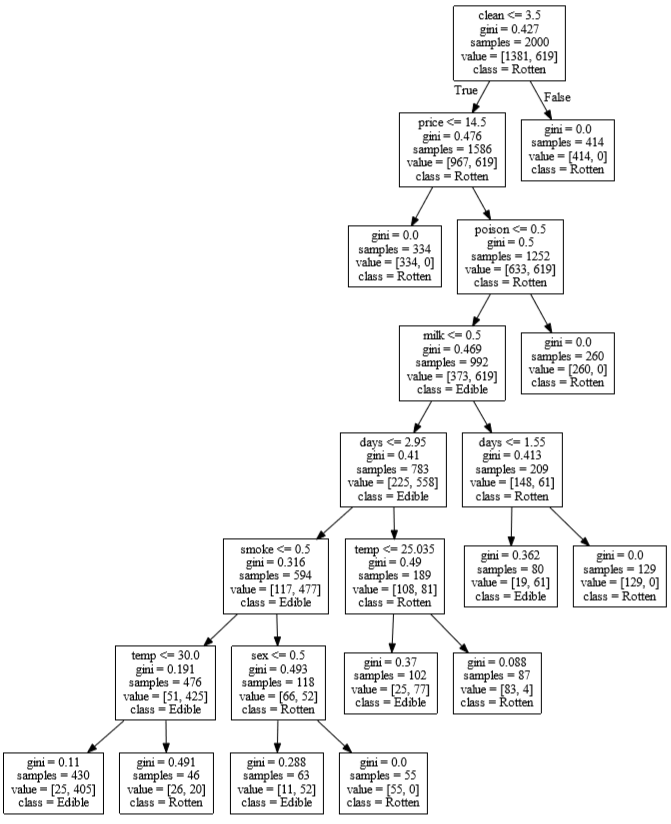
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuarcy | Sw\_easy | Sandwich | Heart | Bank |
| DT | 1 | 0.998 | 0.787 | 0.897 |
| DT\_prune | 1 | 0.942 | 0.787 | 0.893 |
| KNN | 0.834 | 0.762 | 0.672 | 0.880 |
| NBayes | 0.582 | 0.54 | 0.869 | 0.865 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AUC | Sw\_easy | Sandwich | Heart | Bank |
| DT | 1 | 0.998 | 0.790 | 0.749 |
| DT\_prune | 1 | 0.945 | 0.792 | 0.577 |
| KNN | 0.768 | 0.738 | 0.673 | 0.571 |
| NBayes | 0.737 | 0.644 | 0.869 | 0.636 |

Noticed that bank dataset has been preprocessed by changing categorical data into numeric value (roughly to all columns)

1. **Discussion**
2. Rules comparison (sandwich dataset)

Decision tree (pruned 2%)



There are 7 rules results to rotten (compared to 8 rules defined)

Another point, when the training data size is small, decision tree may not precisely get the attribute and correct value for classification, thus the rule may be totally different than the rules we defined.

1. Pruned or not

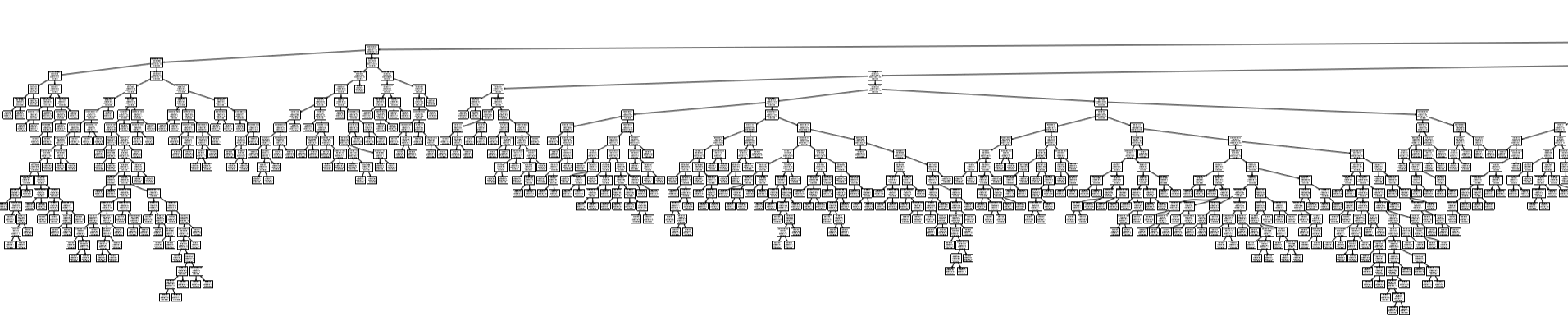
Here is another chart of pruned 0.1%, 0.2%, 1%, 2%, 5% comparison

Using bank data as example

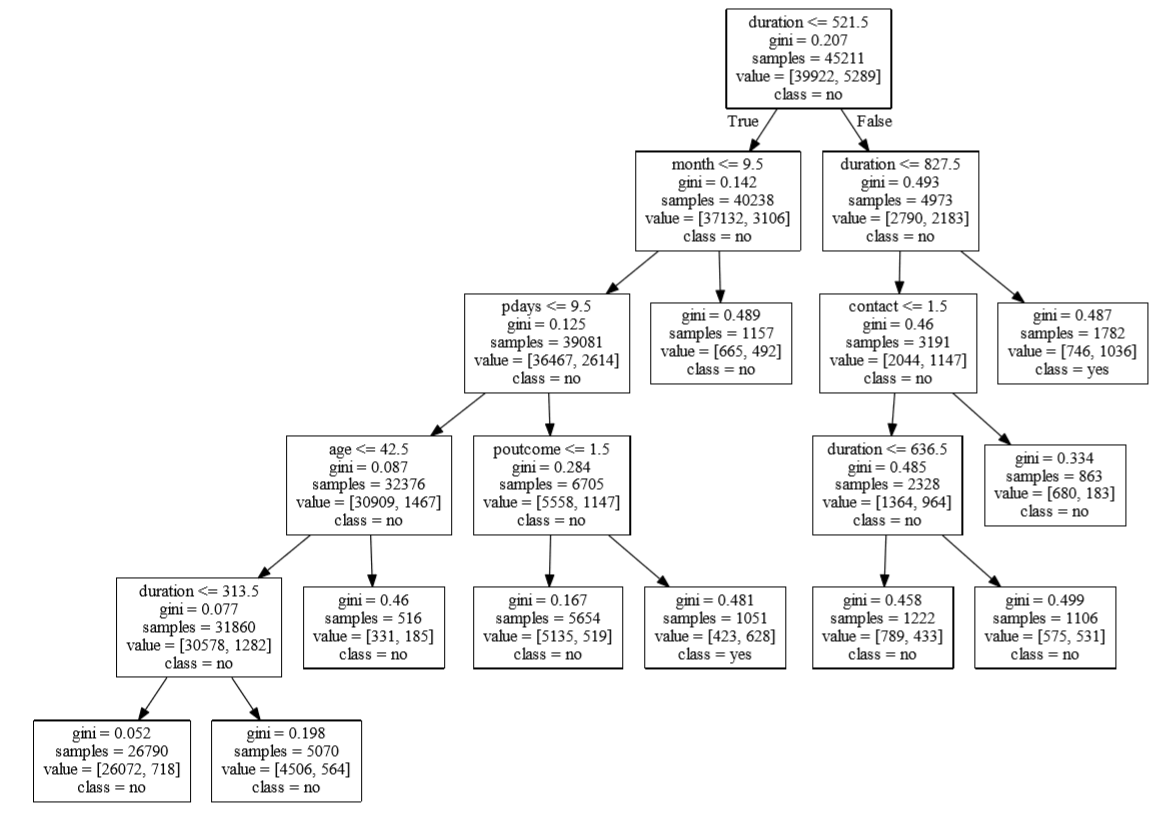
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRUNED | 0.1% | 0.2% | 1% | 2% | 5% |
| ACC | 0.903 | 0.901 | 0.896 | 0.893 | 0.885 |
| AUC | 0.684 | 0.650 | 0.650 | 0.577 | 0.5 |

As you see, accuracy and AUC is dropping when over pruned, however, such pruning is needed especially when data is not ordered and furthermore, discrete categories. And moderate pruning may not dropping the accuracy and can help user clearly understand how the data should be split based on important features.

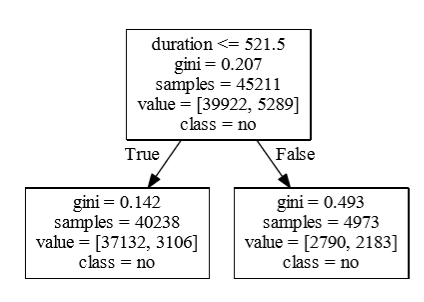
No pruned



Pruned 2%



Pruned 5%



1. Categorical data

Notice that in heart dataset, there are two attributes that is not ordered. Thus decision tree method performs worse than applying on other datasets.

1. **Conclusion (Combining Result and Discussion)**

Decision tree outperforms other methods on these four dataset, but slightly performed poor on heart dataset. We usually assume decision tree is not capable for handling categorical data. And pruning is effective to prevent overfit on testing data.