Comparing KNN and Decision Tree

Kaiwen Xu, Byron Chen, Kevin Li

Abstract

In this project we investigate the performance of two classification models, KNN and decision trees on two datasets. One is "Adult" dataset, where we predict if an adult's income exceeds $50K/yr from various factors. Another one is "Bank" dataset, where we predict if the customer is subscribed to the bank term deposit plan. We find that the decision tree classifier achieves better accuracy than KNNs, and runs significantly faster than KNNs. We also find that choosing a good hyper-parameter and feeding more data can result in better accuracy. In addition, we explore how dropping a feature from dataset can improve the performance.

1 Introduction

Inspired by Haojun Zhu, who did a thorough but traditional statistical analysis on the "Adult" dataset[1], we want to use this dataset to compare the performance of two models we've learnt recently, namely KNN and decision tree. The goal of the data is to predict whether an adult's income exceeds $50K/yr from various factors, such as age, sex, nationality, etc... After careful analysis of the result, the decision tree has a better performance compares to KNN. And both hyper-parameters and data size can affect the result. To validate our findings, we find another dataset "Bank" to perform experiments. In this dataset, we predict if a customer is subscribed to the bank term deposit plan from various factors such as age, education, martial, etc.

2 Datasets

2.1 Adult dataset

This dataset includes the information acquired in a census. When we process the data, we find that around 2000 instances have missing data under columns *Workclass*, *Occupation* or *Nativa_country*. But all other columns are correctly filled. Therefore, these instances are too useful to be dropped. So, we go beyond and implement the "data imputation" technique. Since these features contain categorical values, we fill the missing entries with the most frequent category (e.g. "United-States" for missing native country). For *Occupation*, which has 3 categories with almost equal cardinality, we fill its missing entries with a random choice among these 3 categories.

We also find some completely duplicate instances in the dataset. Because only 32,561 instances are collected, we believe it is unlikely to have 2 people with the exactly same values for all features. We believe they are duplicated by mistake. So we decide to also remove these instances.

2.2 Bank dataset

This dataset includes the information from some customers of a Portuguese bank institution. During the processing of the data, we find that there is no missing data or duplicate instance. By testing each feature's influence on the result, we find that the attribute "Contact" influences little. So, it is dropped for better prediction. The "Bank" dataset has not been split into training set and test set. Therefore, we split the data into 75% training and 15% test with a random seed of 111.

3 Results

In this part, we report our results of the experiments. Since the model training and tuning processes are the same for both "Adult" and "Bank" datasets, we will explain our observations using only the results from "Adult" dataset.

3.1 Performance of KNN and Decision Tree

In our experiments, KNN and decision tree both get cross validation accuracies between 80% to 85%. But the decision tree generally has a higher accuracy than KNN. Decision tree also runs much faster than KNN (3 seconds compare to 35 seconds for each cross validation).

3.2 Changing hyper-parameters

For KNNs, the hyper-parameter is clearly the integer K. We try all the integer values between 1 to 25 for K (as shown below). The training accuracy is 100% when K=1, since the near neighbour is itself for every data point in the training set. It decreases as K gets larger. The validation accuracy increases as K gets larger, because small K (less than 5) tends to cause overfitting. After K=10, accuracy stabilizes to around 80%.

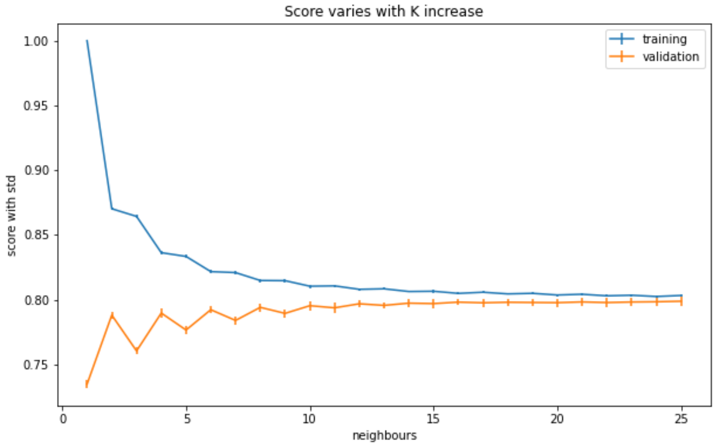

Figure 1: KNN with different K

For decision trees, the hyper-parameter we choose is the "min_impurity_decrease" (referred to as MID later). A node will be split if this split induces a decrease of the impurity greater than or equal to this value. We try the a list of exponentially spaced values from 0.000005 to 1. The training accuracy, benefited by overfitting, is high for small MIDs. The validation accuracy peaks at MID=0.0005, and drops significantly both when MID is less than 0.00005 or above 0.01. Since in these two cases, either overfitting or underfitting appears.

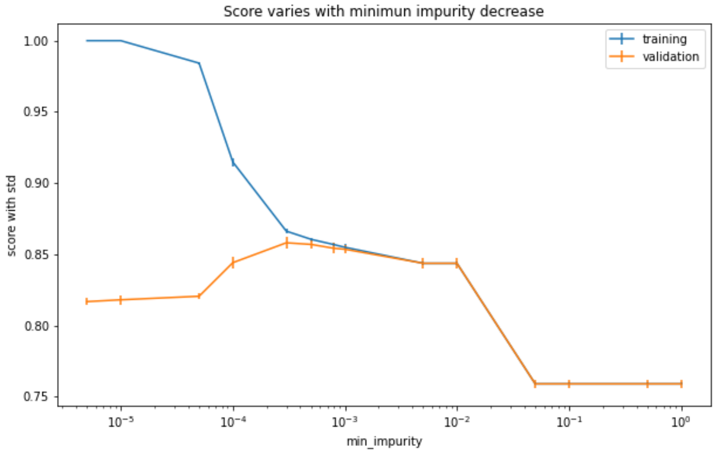

Figure 2: Decision Trees with different MID

3.3 Reducing the amount of data

We also research on how reducing the amount of data impacts results. We train both models on different percentage of the original dataset (from 10% to 100%). We observe both models to perform worse when the amount of data is reduced [See Figure 3 and Figure 4 in Appendix]. Notice that the decision trees with less training data actually predicts better on training set. We think this is because on less amount of data, it is easier to find a tree split that overfits the training data. But it certainly cannot generalize well.

4 Discuss and Conclusion

Finally, we can conclude that decision tree model has a better performance over KNN, at least on our two datasets. We also find that the hyper-parameter in a model should be neither too high nor too low. By grid searching, we can find an optimal hyper-parameter to select. Besides, by reducing the amount of data, the validation accuracy for both models are found to increase.

4.1 Go beyond

Going beyond, we also investigate how dropping a feature can affect the results for both models. For KNN, we observe a significant increase in accuracy (from 80% to 85%) after dropping the feature "Fnlwgt", which is a weight given by the U.S. Census Bureau to each person. "Fnlwgt" is considered noise here because its values are very large numbers that are not closely related to a person's income. And we know that KNN is extremely sensitive to noise with large scale. On the other hand, decision tree's accuracy does not change too much after dropping "Fnlwgt". We believe that's because decision tree inherently skips unimportant features. (See Figure 5 and 6 in Appendix)

5 Statement of Contributions

Kaiwen implemented the "cross-validation" function and tuned both models on the "adult" dataset. Byron and Kevin both did the preprocessing for the “Bank” dataset and Byron tuned KNN model on it while Kevin did the decision tree.

6 Appendix

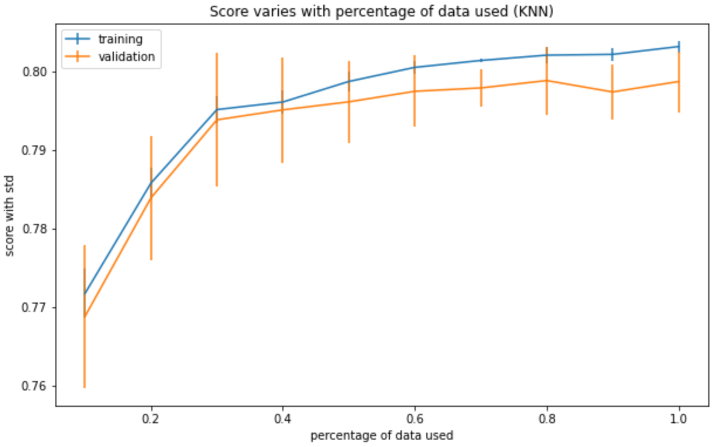

Figure 3: KNN with different % of data

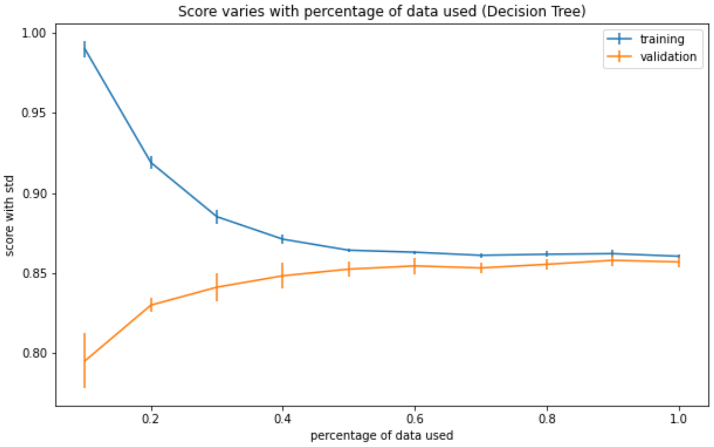

Figure 4: Decision Trees with different % of data

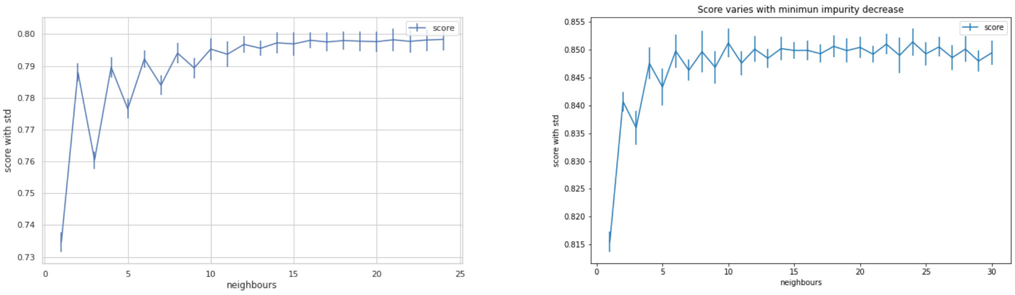

Figure 5: L: KNN with "Fnlwgt" R: KNN without "Fnlwgt"

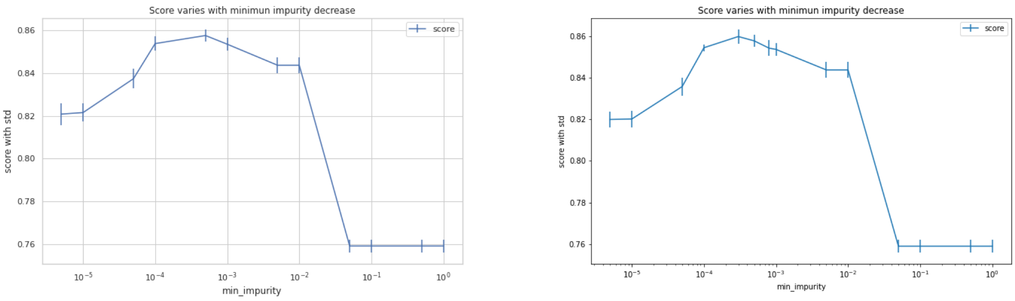

Figure 6: L: decision tree with "Fnlwgt" R: decision tree without "Fnlwgt"

References

[1] Haojun Zhu. *Predicting Earning Potential using the Adult Dataset*. 2016. URL: `https://rpubs.com/H_Zhu/235617`.