Assignment 1: Getting Started with Machine Learning

Anson Ng Howard Qin Aybuke Ekiz

January 31, 2024

Abstract

In this assignment, we implemented a K-Nearest Neighbour model and a Decision Tree model from scratch, and observed their performances on two datasets: National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset and Breast Cancer Wisconsin. Furthermore, we investigated hyper-parameter selections for the models, comparing their accuracies and Areas Under the Receiver Characteristic Curve. We also experimented with different cost and distance functions, as well as feature selection methods. In terms of test set accuracies, our K-Nearest Neighbour and Decision Tree models performed very close to each other, for both of the datasets, but the K-Nearest Neighbour model had a slightly better AUROC score in both cases.

Introduction

As highlighted in the abstract, the major task for this project was the implementations of the the KNN and the DT models. The other tasks included prepossessing and analyzing the two datasets, comparing the models' performances on the datasets, experimenting with the hyper-parameter choices for the models (K values for the KNN model and the maximum depth parameter for the DT model), investigating the use of different distance and cost functions for the models, and exploring the key feature selection processes for the models.

Methods

We implemented KNN and DT models, which are both supervised learning algorithms, and can both be used for regression and classification. In our case, both of the datasets' target values were categorical, therefore our task was classification.

The main idea of the KNN is to predict the value of a test data point by looking at the K nearest neighbors in the training set. To do this, it calculates the distance between the new test data point and all the points in the training set, and selects the K nearest points (based on a distance function, like Euclidean distance for example). In classification, the most common value of those K nearest points is predicted as the value of the test point. KNN is said to be a non-parametric algorithm, because the only "parameter" it has is a hyper-parameter, which is the value K itself, which is to be determined by the user, not by the model itself.

The main idea of the DT is that it recursively splits the training data into regions, to minimize some cost function (Gini Index or Entropy for example). A hyper-parameter of the DT model is the maximum tree depth, which again, is a parameter need to be set by the user, not by the model itself. The model stops splitting the training data into regions when the maximum tree depth is reached, and then predicts the value of the test point by traversing the tree, starting from the root and based on the features selected to make the region splits at each non-leaf node. The value of the test point is predicted to be the average/mode (based on if the task is regression or classification) value of the lead node it ends up at.

Datasets

National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset Dataset contains 2278 real data instances. It has some health information of the responders' as features, and the age category (Adult or Senior) of the responder as the target value. Breast Cancer Wisconsin Dataset contains 699 total, and 499 unique real data instances. During the prepossessing of the dataset, we chose to remove the duplicate data points. It has some health information as features, and the benign or malignant class categories as the target value.

As a basic analysis of our two datasets, we computed the mean of each feature for the different classes, and computed the squared differences of the means. For the NHANES dataset, the features with the most variances among the Adults and Seniors were, in order, LBXGLT, LBXGLU and LBXIN. And as expected, the feature with the least variance among the Adults and Seniors was the gender. All the three features with the most variance among the age groups are about to the responder's glucose levels, which would make sense to be associated with their age. For the Breast Cancer dataset, the features with the most variances among benign and malignant classes were, in order, Bare Nuclei, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli and Marginal Adhesion, though in this dataset, all the features' variances among groups were relatively close to each other.

We employed slightly different data prepossessing methods for our two models. Since KNN is sensitive to scaling, we standardized our datasets for it. Since DT does not have the same concern, we used the datasets as they were in the DT model, without standardization. For both models, we split both our datasets into equal training, validation and test sets. For the KNN model, we explored both feature selection, and just inputting the dataset with all the features, with no feature selection. The performance differences among those two methods will be discussed further in the Results section. We used correlation matrix to find out the features that correlate the most with the target values.

Results

Experiment 1

Compare the accuracy and AUROC of KNN and DT algorithm on the two datasets.

As we will discuss below in Experiment 2 and Experiment 3, we will apply K value selection and max tree depth selection. To compare the performances of the KNN and DT, we use the models with the best performances of each model to compare them. Using the K value to be 5, as discussed in Experiment 2, and using the max tree depth to be 3, as discussed in Experiment 3, we found the accuracy of the KNN model on the Dataset 1 to be 0.80 and its AUROC to be 0.55. As a further exploration, we created the confusion matrix of KNN on the Dataset 1, and found out that its False Positive rate is quite high. This can shed some light on the poor AUROC value.

KNN model on the Dataset 2: Accuracy: 0.95 AUROC score: 0.94

DT model on the Dataset 1: Accuracy: 0.84 AUROC: 0.53 Similarly also created the confusion matrix, and saw that the DT model also has a high False Positive rate.

DT model on the Dataset 2: 0.93 AUROC: 0.93

Experiment 2

Test different K values and see how it affects the training data accuracy and test data accuracy of KNN.

We split our datasets into training, validation and test sets, so in order to test the different K values, we used validation accuracy values. As it can be seen from the figure below, for Dataset 1, the best validation accuracy is reached when the K value is 5, and after that there is a slight trend downwards. For Dataset 2, the best validation accuracy is reached when the K value is 11, and after that there is a slight trend downwards similarly. Note that we select our training, validation and test data indices randomly, so at each run, the validation accuracies change slightly, but in most cases it peaks at some K value and after that there is a slight trend downwards.

Experiment 3

Similarly, check how maximum tree depth can affect the performance of DT on the provided datasets.

As similar to the K value for the KNN, maximum tree depth has a similar effect on the validation accuracies. With a similar approach to the K values, we found that max tree depth of 3 for the dataset 1 and a max tree depth of 4 for the dataset 2 results in the best validation accuracies.

Experiment 4

Try out different distance/cost functions for both models.

For KNN, we tried the Hamming distance. Here is the ROC curve comparison of the KNN model with Euclidean and Hamming distance functions, on Dataset 2.

For DT, we tried entropy cost function, instead of the Gini Index, but there was no observable difference in terms of the model performance.

Experiment 5

Plot the ROC for KNN and DT on the test data

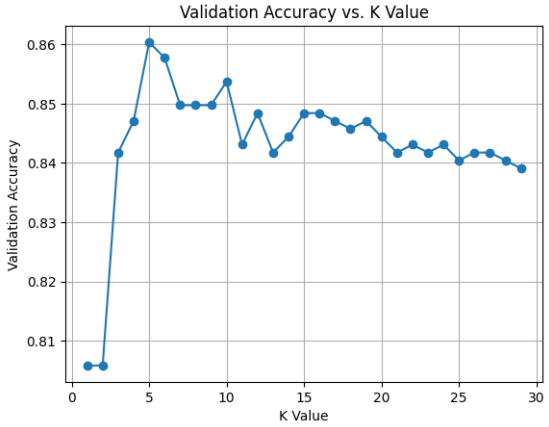

Figure 1: Experiment 2: Dataset 1

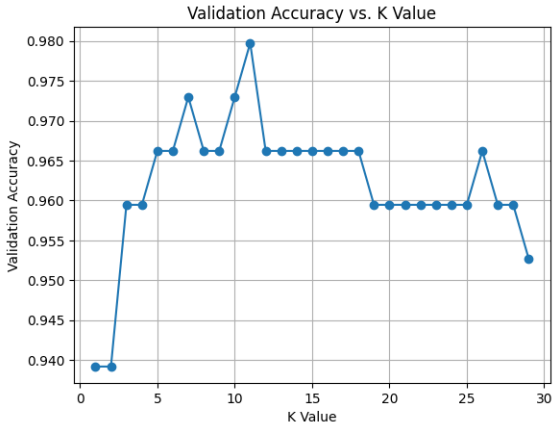

Figure 2: Experiment 2: Dataset 2

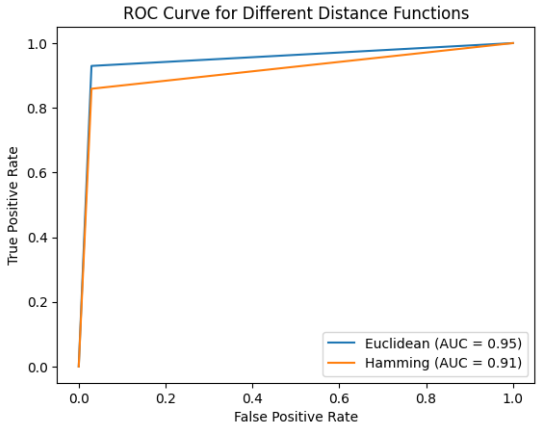

Figure 3: Experiment 4: Dataset 2

Similarly to Experiment 1, to compare the models to each other, we used the best performing model for each. One important thing to note here is that even though both the KNN and DT had high validation test accuracies, their ROC scores are not good. It is an important reminder that simply looking at the accuracies may not be the best evaluation of the performance of a model. As seen on the graph, both models performed way better on Dataset 2. The fact that they both performed poorly on one dataset and well on one dataset may suggest that it could have been due to the distribution of the dataset. Dataset 2 is a more balanced dataset compared to Dataset 1 in terms of class labels, and both KNN and DT are models that might perform poorly on a not-balanced dataset.

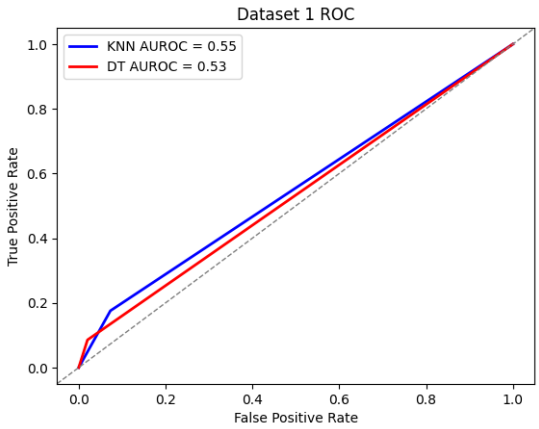

Figure 4: Experiment 5: Dataset 1

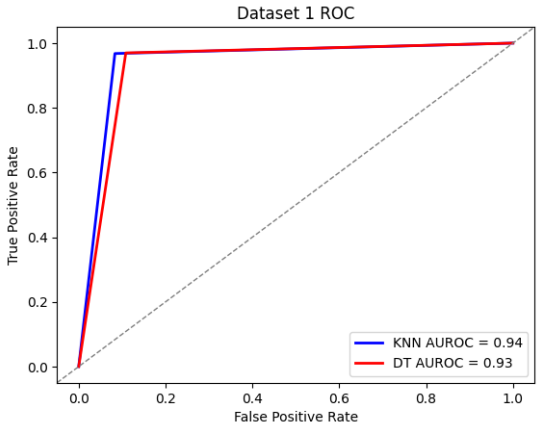

Figure 5: Experiment 5: Dataset 2

Experiment 6

Describe how you obtain the key features used in KNN (e.g., external feature selection by correlation with the labels).

As described briefly in the Datasets section, we used feature correlation to select the important features for KNN. Using a correlation matrix, we looked at what features correlate the most with the class labels. By only selecting the features that correlate the most, we are able to overcome the KNN sensitivity to noise features. However, in our case, this key feature selection did not result in a considerable model performance improvement. It only increased the accuracy of the model on Dataset 1 from 0.8 to 0.83 and the AROUC from 0.55 to 0.58.

Experiment 7

For DT, you can compute a rough feature importance score for each feature d by counting the number of non-leaf nodes where feature d is used. By keeping track of the feature that is selected to be the feature determining the split at each non-leaf node, when our DT is done fitting the training data, we can have a count of the features used for splits. For Dataset 1, 'LBXGLT': 5, 'LBXIN': 4, 'PAQ605': 1, 'LBXGLU': 2, 'BMXBMI': 1, 'DIQ010': 1 were the counts of the features that were the deciding factor of the split. The numbers given are the counts of how many times that feature were the deciding factor. Top 6 is reported as there is a tie.

For Dataset 2, 'Uniformity of Cell Shape': 1, 'Clump Thickness': 2, 'Uniformity of Cell Size': 2, 'Bland Chromatin': 1, 'Mitoses': 1' were the counts of the features that were the deciding factor of the split.

Going back to our discussion at the Datasets section regarding the simple mean difference of the features for positive and negative target groups, For Dataset 1, the features with the most variances among the Adults and Seniors were, in order, LBXGLT, LBXGLU and LBXIN.

For the Dataset 2, the features with the most variances among benign and malignant classes were, in order, Bare Nuclei, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli and Marginal Adhesion.

Comparing these, we see that there are some overlaps, especially for the Dataset 1, but still they are not all the same. It would make sense that there are some overlaps, because the more different values a feature has for the different target groups, it is more likely that it is a feature that can offer a deciding threshold value. However, this is not necessarily true. A feature might have different means for two different classes, but it may not have clustered separation, in which case it would not be a good candidate for the deciding feature of a split in the DT. Furthermore, when deciding the splits, DT considers the current structure of the tree as well, which is not true for just simple mean difference calculation.

Discussion and Conclusion

As conclusion, our implemented-from-scratch KNN and DT models were not so successful on Dataset 1, but they performed well on Dataset 2. The two models performed around the same level, no significance difference was observed. We got to experiment with hyper-parameter tuning, explored different distance and cost functions and analyzed the key features of the datasets.

For future investigation: During the experiments, we speculated that the reason our models are performing badly on Dataset 1 and performing well on Dataset 2 might be due to the distribution of the datasets, i.e. if they are balanced in terms of the class groupings or not. This could be a further point to investigate, we could do another experiment using precision-recall which suited better to measure performance on imbalanced data.

Statement of Contribution

Anson: Datasets prepossessing and the KNN model implementation
Howard: Datasets prepossessing and the DT model implementation
Aybuke: Running Experiments and the Write-up