Assignment1COMP551- Group 19

Ridwanur Rahman, François Pagé, Isabella Chiaravallotti

January 2024

1 Abstract

In this assignment we explored the performance of two machine learning models, K Nearest Neighbor (KNN) and Decision Tree (DT), on two different datasets (NHANES dataset and the BC Wisconsin dataset). We found that of the two models, KNN was best for the BC dataset while DT was best for the NHANES dataset.

2 Introduction

We tested KNN and DT on two datasets: an age prediction set (henceforth referred to as NHANES) and a breast cancer dataset (henceforth referred to as BC) (NA,NA., 2023; Wolberg, 1992). Historically, these datasets have been used with machine learning algorithms for both machine learning investigations and medical investigations (For example, see: Vangeepuram,2021; Lopez-Martinex,2020;Li,2018;Sivapriya,2019). More specifically Li et al. (2016) explored 5 different classification models on the BC set and used AUC and prediction accuracy for assessing models as we have done in our experiments. They found random forest to be the most accurate model, however random forest was not a model investigated in this assignment. Similarly, a subset of the NHANES dataset was used by Vangeepuram et al. (2021) to predict youth diabetes risk using machine learning, exhibiting a practical, real-world application of machine learning on our datasets.

Our task was to evaluate model performance on these two datasets using either KNN or DT. On the NHANES set, the task involved classifying observations into one of two classes, adult or senior. For the BC dataset, our task involved classifying observations into one of two classes, benign or malignant. We found that the accuracy of KNN for the BC set was 0.95 and the accuracy of KNN for the NHANES set was 0.997, while the accuracy of DT for the BC set was 0.88 and the accuracy of DT for the NHANES set was 1.0.

3 Methods

KNN is a non-parametric method used to classify an observation based on its proximity to other observations used to train the model. They can also perform regression. During training we provide the model with the feature coordinates for each observation(x), and the corresponding label(y). Then, a test observation with an unknown label is mapped with the other data points. Using a distance function such as Manhattan or Euclidean distance, the K nearest neighbors of the test observation are identified. The ideal value for K can be determined by testing model performance on a validation set for a given set of potential K values. Then, we identify the most probable label for our test observation, according to the class probabilities of the neighboring labels.

A DT is comprised of a series of nodes that split the data according to a certain threshold of a given parameter. DTs can perform classification and regression. In other words, a DT divides data into a certain number of regions, with each region corresponding to a certain class or value. The ideal number of nodes, the feature used for splitting, and the threshold for making a split are all determined during model training. We optimize these parameters by minimizing a cost function such as entropy or Gini index.

4 Datasets

The NHANES dataset is comprised of 2278 observations and 10 features. Of the output labels 1914 were labeled 'Adult' and 364 were labeled 'Senior'. The BC dataset is comprised of 698 observations and 11 features. Of the output labels, 241 were malignant and 457 were benign. It is important to know how many observations take on each label, because we may assess an unbalanced dataset slightly differently than a balanced dataset during model evaluation. Both datasets are unbalanced, NHANES more so than BC.

We also took a look at the scale of each of the factors using a boxplot. This allows us to determine whether or not it will be necessary to normalize our features prior to model training. For example, we found that 10 of the 11 NHANES data exist on a similar scale, but one of the features is measured on a much larger scale than the others. This flags us to normalize the

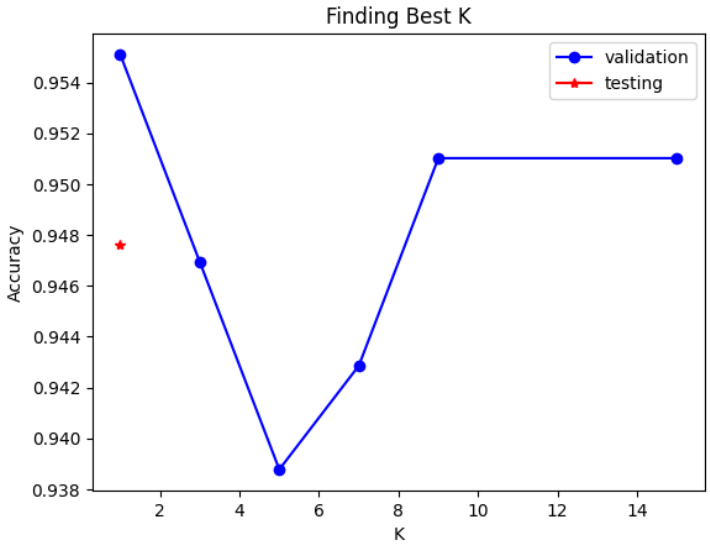


Figure 1: Finding the best K value for the BC set - 6 K values were assessed and the K with the highest accuracy was chosen. Best K was validated with holdout validation set (red star)

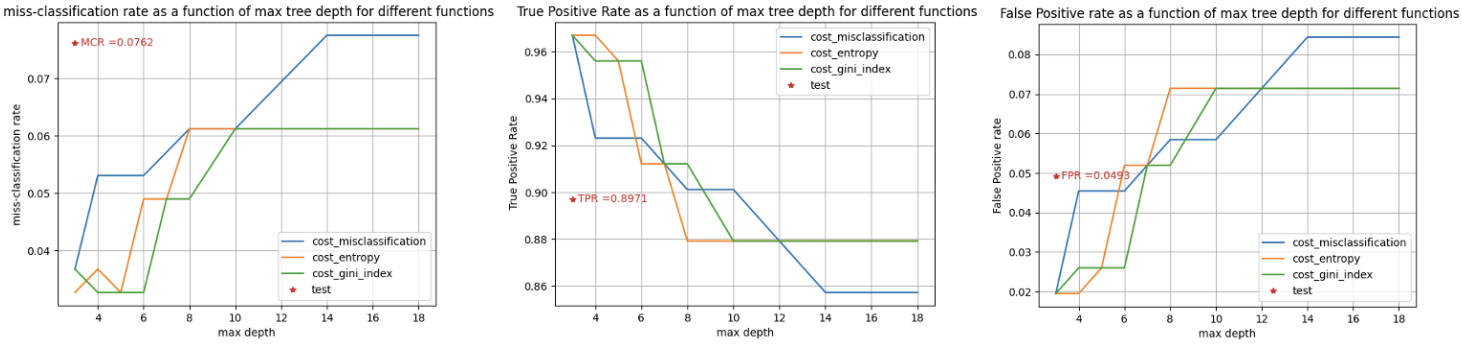


Figure 2: Finding the optimum tree depth for the BC set. Optimum depth was validated with holdout validation set (red star)

data before implementing the model. While analyzing the features, we also identified any missing data points and imputed the missing points using feature averages.

We also looked at some basic statistics for each feature including, mean, variance and correlation of the feature with the outcome label. This allows us to get a better feel for the data, make decisions about pre-processing and identify which features correlated the highest with the label. These features will become most important during model training and we may choose to simply eliminate variables that add noise to the model.

5 Results

(1) We found that the accuracy of KNN for the BC set was 0.95 and the accuracy of KNN for the NHANES set was 0.997, while the accuracy of DT for the BC set was 0.88 and the accuracy of DT for the NHANES set was 1.0. The AUROC for the NHANES set using KNN was 0.97. The AUROC for the NHANES set using DT was 1.0. The AUROC for the BC dataset was 0.95 for both KNN and DT.

(2) We tested six different K values and observed the effect on accuracy (Figure1). To accomplish this, we split the data into a training, testing, and validation set. The model was trained with the training data, and assessed with the validation set, for each K. The test set was used to evaluate this K value on unseen data. The K yielding the highest accuracy was selected for running experiments.

(3) We investigated the impact of tree depth on the performance of DT (Figure 2). This was also achieved by splitting the data into a training, testing, and validation set. Training was done with the training data, and assessed using the validation set, for each tree depth. The performance was determined by calculating individually the Miss-classification Rate, True Positive Rate, and False Positive Rate as a function of the max tree depth for different cost functions. The optimum tree depth that minimized the cost was then selected to run experiments on the test set.

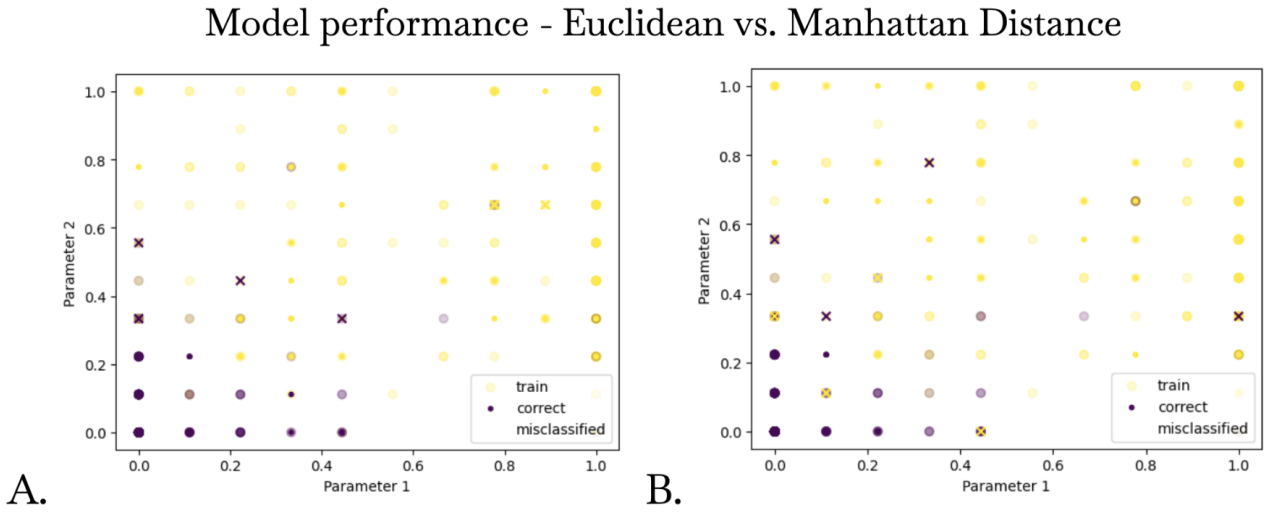


Figure 3: Model performance for the BC dataset using Euclidean distance (A) and Manhattan distance (B). Model accuracy in 2A was 0.93. Model accuracy in 2B was 0.94. The different distance functions resulted in different correctly classified points.

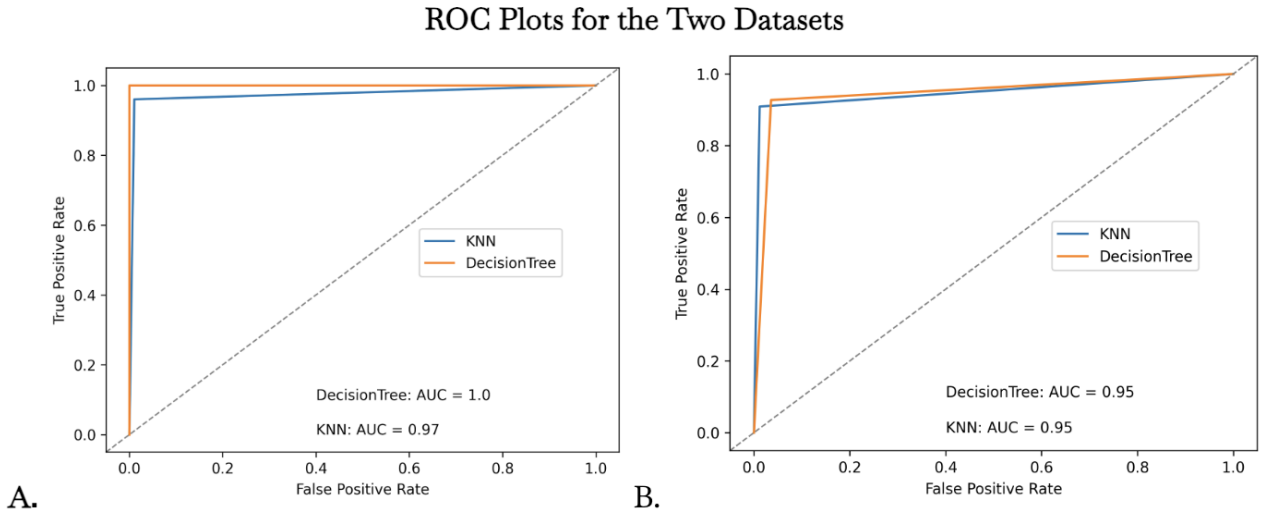


Figure 4: ROC plot and AUROC for (A) the NHANES dataset and (B) the Wisconsin BC dataset.

(4) We tested two different distance functions for KNN, Euclidean distance (the square root of the squared sum of the distance between 2 points) and Manhattan distance (the sum of the absolute difference between 2 points) . We can observe that the distance function implemented has a differential impact on our test observations. For example, we used Euclidean distance and Manhattan distance. There was a numerical (but not significant) difference between the prediction accuracy between the two distance functions on the BC set (0.93 with Euclidean distance and 0.94 with Manhattan distance). However, if we plot the correctly classified versus misclassified points, we will observe that the distance functions accurately predict a different sets of individual observations (Figure 3).

(5) We also tested different cost functions for DT (Figure 2). For the BC dataset, we found the best cost function to be Gini Index. For the NHANES set, it was the Miss-classification Cost function.

(6)We plotted the ROC for KNN and DT for both sets (Figure 4). For the NHANES dataset, we can see that the decision tree is slightly superior. For the BC dataset, we found that the ROC curves were identical.

(7) We determined key KNN features by calculating the correlation between the features and the labels and found that feature selection is highly significant. For example, we implemented the model using the two most highly-correlated features, and two random features for the NHANES dataset (Figure 5A and Figure 5B). We can observe that when we carefully select features the decision boundary is clearly visible, and if we choose two random features, the decision boundary is less distinct and does not describe the classes. The accuracy in Figure 5A is 1.0, while the accuracy in Figure 5B is 0.827. An accuracy of 0.827 may appear relatively high, but we can recall that this dataset is unbalanced and roughly 84 percent Adult labeled and 16 percent Senior labeled so an accuracy of 84 percent could be achieved by simply putting all observations into the

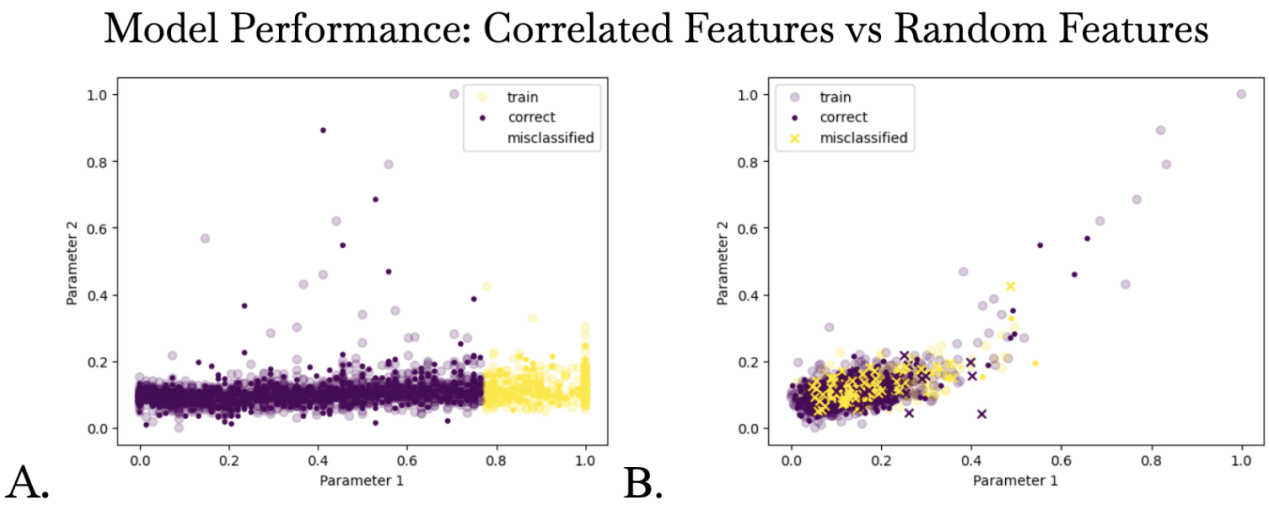


Figure 5: A comparison of model performance under (A) feature selection based on correlation and (B) random feature selection. Model accuracy in 3A is 1.0. Model accuracy in 3B is 0.827

Adult category.

(8) We computed feature importance score for each feature in our datasets. This was computed by calculating the correlation of each of the features of the datasets with respect to the target variables. The two features with the highest correlation scores (aka feature importance scores) were then selected to train, validate, and test our models with.

(9) In order to assess model performance on every data point, we implemented K-Fold cross validation. The results of the K-Fold cross validation were used to plot the ROC curves for each model on all points in each dataset.

6 Discussion and Conclusion

These experiments demonstrate the importance of feature selection, data processing, and hyperparameter selection for two machine learning models, KNN and DT. We found that for KNN, optimizing K is essential for maximizing model performance. Similarly, we found that optimizing tree depth, cost function, and calculating feature importance is essential for maximizing model performance.

7 Statement of Contribution

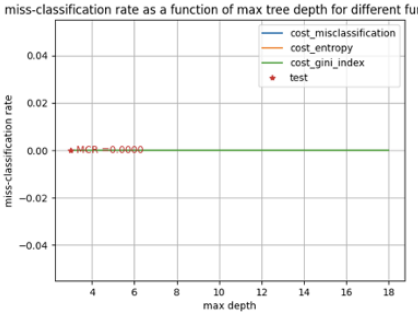
Ridwanur implemented DT and contributed to the report. Francois optimized DT hyperparameters and contributed to the report. Isabella implemented data processing, implemented KNN, assessed model performance and contributed to the report.

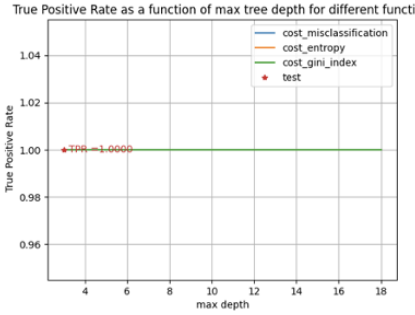
8 References

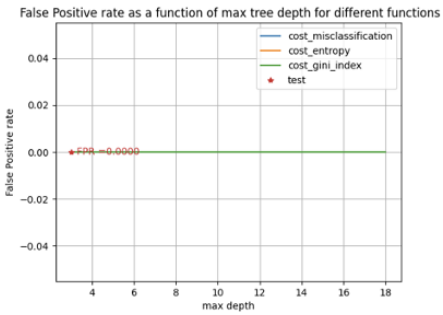
Li, Yixuan, and Zixuan Chen. "Performance evaluation of machine learning methods for breast cancer prediction." Appl Comput Math 7.4 (2018): 212-216. López-Martínez, Fernando, et al. "An artificial neural network approach for predicting hypertension using NHANES data." Scientific Reports 10.1 (2020): 10620. NA,NA. (2023). National Health and Nutrition Health Survey 2013-2014 (NHANES) NHANES Subset. UCI Machine Learning Repository. https://doi.org/10.24432/C5BS66. Sivapriya, J., et al. "Breast cancer prediction using machine learning." International Journal of Recent Technology and Engineering (IJRTE) 8.4 (2019): 4879-4881. Vangeepuram, Nita, et al. "Predicting youth diabetes risk using NHANES data and machine learning." Scientific reports 11.1 (2021): 11212. Wolberg,WIlliam. (1992). BC Wisconsin (Original). UCI Machine Learning Repository. https://doi.org/10.24432/C5HP4Z.

9 Addendum

Same as shown in the report, but for the NHANES data







Best K for NHANES data