COMP 551 Assignment 1 Report - Winter 2024

Group 17: Jimmy Sheng, Iris Wang, Keyu Wang

January 29, 2024

Abstract

In this assignment, we generated discussion around two essential machine learning algorithms: K-Nearest Neighbour (KNN) and Decision Trees (DT). We first loaded and cleaned the data from source provided in the handout, then implemented the two algorithms as required, then using different parameters and cost functions for each algorithm, dived deeper into the optimization of such algorithm. The process facilitated a detailed exploration of each algorithm's strength and weakness in different scenarios. Such is explored by measuring and contrasting the accuracy and AUROC (Area under the Receiver Operating Characteristic Curve) of each model. After careful qualitative and quantitative analysis, we reached the conclusion that DT and KNN both perform very well with CANCER data set and less as well with NHANES dataset.

Introduction

In this assignment, the two data sets we analyzed are both healthcare related.

1. The National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset is administered by the Centers for Disease Control and Prevention (CDC). The combined feature set represents the most comprehensive health and nutrition information acquired among a nationally representative sample of the US population. The subset not the full set is used in this project since though expansive, the full data set is too broad for analytical purpose. Therefore we use a narrowed down subset which includes the following features: respondent's gender; whether or not respondent engages in moderate or vigorous-intensity sports, fitness, or recreational activities in the typical week; respondent's Body Mass Index; respondent's Blood Glucose after fasting; whether or not the respondent is diabetic; respondent's Oral and respondent's Blood Insulin Levels. The data set also includes information about respondent's age, which is excluded from our analysis ; and respondent's age group, which is the target of our analysis. With KNN, we were able to reach a maximum accuracy of 84.11% and with Decision Tree, 83.49%.

2. The Breast Cancer Wisconsin data set is a 9-feature data set with 8 groups collected over a period of 2 years. The reason why data is collected over a 2 year period is that samples arrive periodically as Dr.Woldberg reports his clinical cases. Therefore, the database is provided in chronological grouping. This data set has missing values for the Bare_nuclei feature in some rows. Therefore we dropped the rows with missing data to resolve this issue. Since all 9 features have very large square difference, we included all of them when applying K-Nearest Neighbour algorithm. With KNN, we reached a maximum accuracy of 96.15%, and 92.60% with Decision Tree.

After comparison of both methods in both datasets, we reached the conclusion that both methods performs better with CANCER data set (both above 90%), and that KNN performs marginally better than Decision Tree on both datasets.

Methods

Both KNN and DT are supervised learning techniques used for classification and regression.

KNN is a dummy learner. It's a straightforward, intuitive machine learning algorithm based on the principle of proximity. It involves identifying the 'k' closest data points, known as neighbors, to a new, unseen instance based on distance metrics such as Euclidean, Manhattan, or Hamming distance. The KNN algorithm make prediction based on the which center is closest to the point we're trying to classify. As a non-parametric and instance-based method, KNN does not explicitly learn a model. Instead, it memorizes the entire training dataset and performs computations at the time of prediction, making it particularly sensitive to the choice of 'k' and the distance metric used. Therefore at implementing the algorithm, we experienced with multiple k values until over-fitting occurs to find the most optimal one. We also tried multiple distance function and ended up finding out that euclidean distance is the optimal function for both dataset.

Decision Trees algorithm constructs a tree-like model of decisions, where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a continuous value (in the case of regression). Rules together form a path from root to leaf. In building the tree, the algorithm selects the attribute that best splits the dataset into subsets based on learning from cost function. This process is recursively applied to each subset until the tree is fully constructed. Tree-depth is an essential hyper-parameter for DTs. In-order to find the ultimate tree depth, we also experimented with multiple tree depth. Though decision Trees are easy to understand and interpret, it can suffer from over-fitting if the tree becomes too deep or complex. Therefore we are very careful when determine the hyper-parameters. We tried multiple cost functions and found out that CANCER dataset performs most optimal using gini index as cost function; while NHANES dataset prediction accuracy is similar using misclassification cost, entropy and gini-index.

Datasets

After loading the datasets into Colab, we first examined if there are any cleanings that needs to be done for each set. Using the command

$$display(NHANES\_X.isna().sum())$$

, we were able to identify that there's no missing data in NHANES data set, and some missing data in Breast Cancer Wisconsin data set. We decide to drop the rows with missing Bare_nuclei data using the command

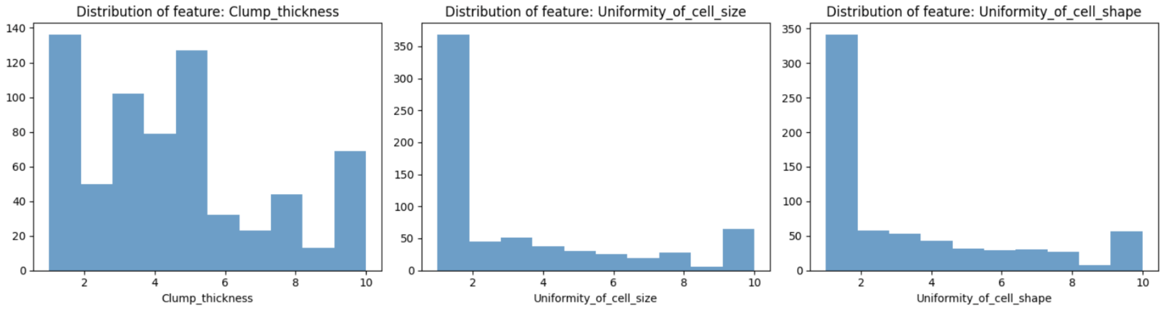
$$CANCER\_X.dropna()$$

, then repeated the command before to double check if all missing data scenarios have been resolved.

In order to understand the dataset better, we plotted the histogram of each feature to see if there is a malformed feature. We noticed the DIQ010 (Doctor told you have diabetes) seems odd to have unevenly distributed into 3 values. We then looked into it and noticed that the three labels corresponds to: 1 as "yes", 2 as "no", 3 as "borderline".

At the next step, we calculated basic statistics like mean, standard deviation, percentiles and a few other statistics. We then ranked the square difference of each features of each dataset and came up with the conclusion that: Tops 3 square difference features for NHANES dataset are LBXGLT (respondent's Oral), LBXGLU (respondent's Bloog Glucose after fasting) and LBXIN (respondent's Blood Insulin Levels), which are the features included in the KNN algorithm we constructed later. For CANCER dataset, we included all features since after experiment, we noticed that the accuracy drops if we exclude some features. The reason why we decide to keep all features is that they all have large square difference. We also drew histogram to help visualize the data better.

Figure 1: fraction of histograms



Results

Using self-implemented *evaluate_acc* and *calculate_auroc* functions, we compare across different hyperparameters like k, and maximum depth and performed visualization of the result.

Our first attempt is to compare AUROC and accuracy for different depths to find the optimal number of centers for KNN. After generating AUROC and accuracy for k from 1 to 10, we found that for both NHANES and CANCER datasets, the best combination of AUROC and accuracy happens when K = 8.

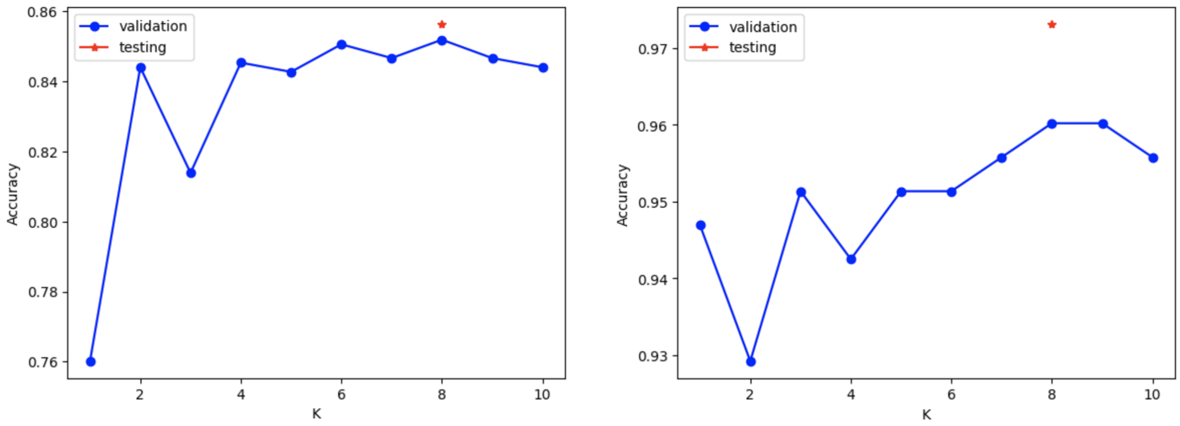
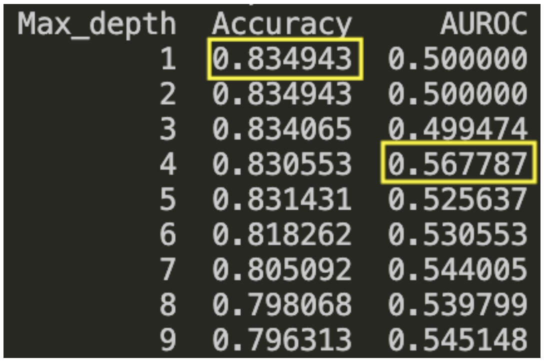


Figure 2: Left: Different k for NHANES ; Right: Different k for CANCER

Then similarly using AUROC and accuracy with different depths, we tried to find the best depth for each dataset. We saw through experiments, in the CANCER dataset, the best depth in 1 to 9 with the best accuracy and AUROC seems to be at depth 2 or 4, and 3 or 4 for NHANES, which raised a question: which metrics should we base on? To answer this question, we decide to proceed with the exploration with the following measures.

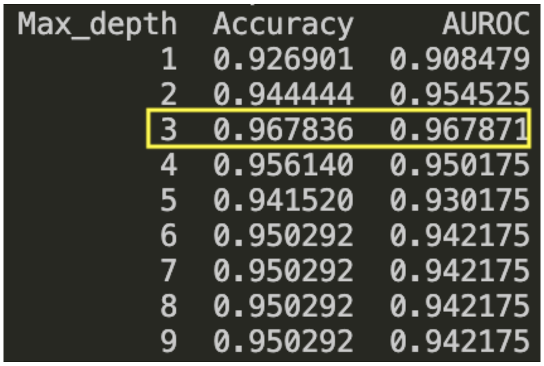
1. Introduce the PR curve, which for class-imbalanced datasets like NHANES, could be a better metric.

2. Split the data into training, validation and testing and use the validation set to select the best K and the best tree depth and evaluate the best choice based on the test set.



```
Max_depth  Accuracy     AUROC
        1  0.834943  0.500000
        2  0.834943  0.500000
        3  0.834065  0.499474
        4  0.830553  0.567787
        5  0.831431  0.525637
        6  0.818262  0.530553
        7  0.805092  0.544005
        8  0.798068  0.539799
        9  0.796313  0.545148
```

Figure 3: Accuracy and AUROC on class-imbalanced dataset, NHANES



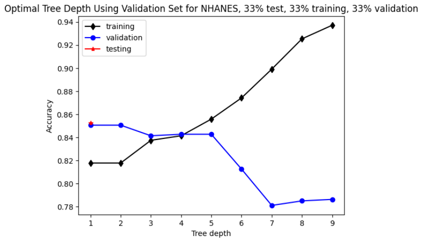
```
Max_depth  Accuracy     AUROC
        1  0.926901  0.908479
        2  0.944444  0.954525
        3  0.967836  0.967871
        4  0.956140  0.950175
        5  0.941520  0.930175
        6  0.950292  0.942175
        7  0.950292  0.942175
        8  0.950292  0.942175
        9  0.950292  0.942175
```

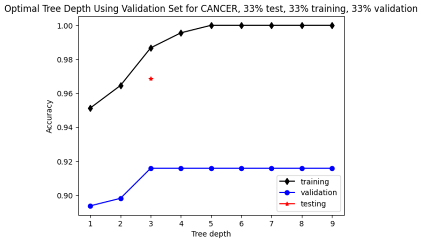
Figure 4: Accuracy and AUROC on class-balanced dataset, CANCER

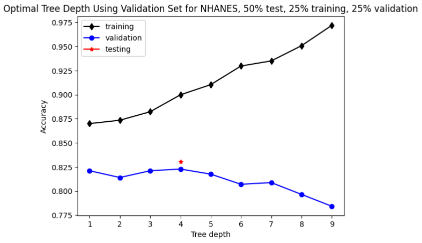
By calculating AUPRC and drawing PRC graphs for different depths, we reached the surprising conclusion that AUPRC did not improve as lot as we thought for the class-imbalanced NHANES dataset. For NHANES dataset, using a depth of 1 seems to be optimal from the result that it has the highest AUPRC; however depth 4 has the best AUROC but a much lower AUPRC. We believe for a class-imbalanced dataset, AUPRC matters more. We decide to confirm that with more explorations. As for the CANCER dataset, it became clear that depth 3 is optimal since it has the highest accuracy, AUROC and AUPRC.

We then dived deep into finding the best k and best depth based on the validation set. Using a 33.3% split for training, testing and validation set, we concluded that for NHANES set, the best k for KNN is 8, consistent with the result above, and the best depth for DT is 1.

We then decided to test with a different ratio, in which we used 50% for testing data, 25% for training data, and 25% for validation data, and found out that now the best depth for DT is 4 for the NHANES set. This finding shows that different split ratios will produce different results for the decision tree. As for the CANCER set, we found the best k as 8 and the best tree depth at 3.

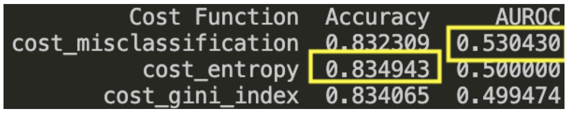






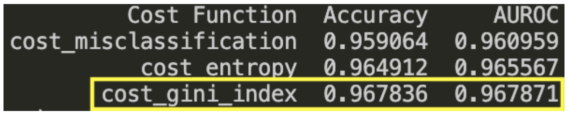
We also explored different options for the distance function in KNN and the cost function in DT. We compared the Manhattan distance and Euclidean distance in both datasets and concluded that Euclidean distance performs better in both datasets. We also compared the accuracy, AUROC, AUPRC for 3 cost functions: misclassification, entropy and Gini index. We found no obvious difference in the 3 cost functions using depth 1 for NHANES dataset and at depth 4, misclassification rate has better accuracy but worse AUROC and AUPRC, while entropy has the best AUROC and AUPRC. As for the CANCER dataset, the choice is very clear: Gini index performs best. Different cost function seems more consistent for the CANCER dataset which always produces better accuracy, whereas for NHANES dataset, different cost functions affect the 3 metrics a lot.

As mentioned in class, k-fold cross-validation can sometimes help improve accuracy. Therefore we decided to perform 5-fold cross-validation for both datasets. However, the result showed that with or without k-fold cross-validation doesn't create much difference in accuracy. We believe it's because k-fold works better with small datasets, like the example of 50 data given in class. Here, both datasets are big



```
         Cost Function  Accuracy     AUROC
cost_misclassification  0.832309  0.530430
          cost_entropy  0.834943  0.500000
       cost_gini_index  0.834065  0.499474
```

Figure 5: Cost function comparison of NHANES dataset, at max depth 3



```
         Cost Function  Accuracy     AUROC
cost_misclassification  0.959064  0.960959
          cost_entropy  0.964912  0.965567
       cost_gini_index  0.967836  0.967871
```

Figure 6: Cost function comparison of CANCER dataset, at max depth 3

and have enough data to build the model and prediction. Therefore the attempt of k-fold cross-validation didn't help improve the model accuracy.

Discussion and Conclusion

ROC VS. PRC Curves for class-imbalanced data

In theory and in fact, ROC works poorly on class-imbalanced data (such as NHANES age dataset), where the highest AUROC is just 0.57 which is slightly better than a random coin toss. Therefore, we also implemented the Precision-recall curve. However, the AUPRC isn't necessarily better, and at some k/depths, even performs worse. We suspect that it's due to the NHANES data having features that are not contributing enough to the prediction, as compared to CANCER where all features have big mean differences between the 2 classes.

Rough Feature Importance in DT

To answer the question of whether the top 5 features in DT are the same as the simple mean difference approach, they mostly alignly, but not entirely. On average 3.5/5 top features are also top 5 using mean difference. We think this is because in a Decision Tree, feature importance is typically assessed based on how much a feature contributes to reducing the impurity (e.g., Gini impurity) in the nodes, and this calculation is NOT solely based on the differences between positive and negative groups for a feature. One thing important to note is that `Clump_thickness` doesn't have a high mean difference but it gets the top ranking in CANCER's feature importance count. It is likely because it has a non-skewed distribution (as shown in Figure 1), unlike the other features where they are all right-skewed, so it gives the DT more opportunity to set it as a tree split.

Conclusion

In sum, we found that for KNN, the best parameters are K = 8 and Euclidean distance for both datasets. We obtained 84% accuracy and AUROC = 0.54 for NHANES, 96.15% accuracy and AUROC = 0.96 for CANCER. As for DT, the best parameters are depth = 1 for NHANES and depth = 3 for CANCER, and using gini-index as the cost function. We obtained 83.49% accuracy and AUROC = 0.50 for NHANES, 92.60% accuracy and AUROC = 0.91 for CANCER.

Statement of Contributions

All 3 of us did the data loading and cleaning individually. Then we all attempted to write KNN and DT, Keyu and Jimmy successfully made it work. Keyu then wrote the DT section for task 3, Jimmy wrote KNN section for task 3 and Iris wrote cross-validation. Iris wrote the abstract, introduction, methods, datasets and results section of the result, with the help of Keyu's graph. Then Keyu and Jimmy finished up the report with the discussion and conclusion section.

1 Citations

1. O. L. Mangasarian and W. H. Wolberg: ”Cancer diagnosis via linear programming”, SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

2. William H. Wolberg and O.L. Mangasarian: ”Multisurface method of pattern separation for medical diagnosis applied to breast cytology”, Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.

3. O. L. Mangasarian, R. Setiono, and W.H. Wolberg: ”Pattern recognition via linear programming: Theory and application to medical diagnosis”, in: ”Large-scale numerical optimization”, Thomas F. Coleman and Yuying Li, editors, SIAM Publications, Philadelphia 1990, pp 22-30.

4. K. P. Bennett & O. L. Mangasarian: ”Robust linear programming discrimination of two linearly inseparable sets”, Optimization Methods and Software 1, 1992, 23-34 (Gordon & Breach Science Publishers).