Assignment 1: Getting Started with Machine Learning

Terrance Yan, Yunjia Zheng, Annie Kang

COMP 551 Winter 2024, McGill University

Abstract

Binary classification has always been one of the major machine learning problems popular in the academic and industrial community and has been receiving full research.[1] Many textbooks and lectures select it as the outset for learning ML. In this assignment, we applied two classic methods, KNN and Decision Tree (DT), to two unique datasets, enhancing them based on detailed statistical analysis. We modified KNN by adding a Gaussian weighting structure and used the Mahalanobis Metric for similarity measurement. Our DT approach was improved by incorporating diverse cost functions like Misclassification Cost, Entropy, and Gini Index for better splits and feature selection. These modifications led to improved adaptability and performance of the models on different datasets. Our results showed that while KNN excelled in accuracy under certain conditions, DT was more stable across various scenarios, particularly excelling in AUROC performance in both NHAMES and Cancer datasets. The study also highlighted the impact of specific hyperparameters on model performance and confirmed the effectiveness of our feature importance evaluation method.

1 Introduction

In this assignment, we focus on implementing and comparing two classification techniques: K-Nearest Neighbour (KNN) and Decision Trees (DTs), applied to two health datasets. These datasets include the National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset and the Breast Cancer Wisconsin (Original) dataset.

The NHANES 2013-2014 Age Prediction Subset [2], created by the Centers for Disease Control and Prevention (CDC), aims to assess the health and nutritional status of the U.S. population. This subset utilizes a range of physiological measurements, lifestyle choices, and biochemical markers to predict respondents' ages. It represents a comprehensive effort to understand age-related health dynamics. A study on this dataset using Adaptive Boosting (AdaBoost) and Support Vector Machines (SVM) achieved an accuracy of 98.54% in predicting the severity of cardiovascular diseases [3].

The Breast Cancer Wisconsin (Original) dataset [4], a multivariate dataset for breast cancer classification, consists of 699 instances with 9 features. Collected by Dr. Wolberg through clinical case reports, this dataset is chronologically grouped and serves as a significant resource for developing and testing predictive models for breast cancer. A study utilizing the k-means clustering algorithm with variations in centroid initialization and distance algorithms significantly improved the Positive Predictive Value (PPV) of classification up to 92% accuracy [5].

We aim to creatively enhance the basic KNN and DT algorithms to assess the efficacy of different models on real-world data. We preprocess these datasets by cleaning the data, computing basic statistics, and understanding feature interactions using tools like NumPy or Pandas. Graphs such as KDE graphs and violin graphs will illustrate our dataset findings.

To analyze the results, we employed a multi-dimensional approach. We evaluated the models based on various performance metrics, including accuracy, the Area Under the Receiver Operating Characteristic (AUROC), and feature importance measures. We also explored the impact of different hyperparameters such as the number of neighbors in KNN and the depth of DTs on model performance. This comprehensive analysis allowed us to gain insights into the strengths and limitations of each model in handling specific types of data, as well as to draw comparisons between them.

This assignment seeks to deepen our understanding of Machine Learning programming, encompassing data handling, algorithm implementation, and model performance evaluation.

2 Methods

2.1 Gaussian Weighted - Mahalanobis Metric based KNN

KNN is a typical non-parametric learning method without an explicit learning progress(the fitting is merely a storage of the existing dataset). There are three major components in this learning method: the size of the neighborhood(measured by 'K'), the similarity measure, and the decision rule(usually majority voting)[6], where the selection of 'K' and the similarity measure plays a significant role.

A higher 'K' (size of the neighborhood) represents a larger search area; it means checking more specificity factors from the examples. This constitutes a trade-off since we want higher accuracy ability but we only want the universal pattern to be learned instead of the peculiar pattern for generalization. As we will see, we use the validation set to decide this hyper-parameter.

As for the metric, it's intuitively a direct measure of similarity. Here, after scrutinizing our dataset, we decide to use the Mahalanobis metric with Gaussian weighting. In comparison to the Euclidean metric, the Mahalanobis metric displays good properties in two major aspects: 1. convenience - offers inherent standardization to a feature vector, 2. greater tolerance to violation of the i.i.d. assumption - eliminates the covariation factor between features by an inherent semi-PCA process. Besides, we decided to apply more weight to the near point to utilize the similarity information revealed in distance (intuitively). The Gaussian function is selected here due to its non-negativity on the entire real set and well-behaved property in not being too biased on the nearest point(which may be fatal if surrounded by noise).

2.2 Decision Tree

The Decision Tree (DT) is a fundamental machine-learning algorithm used for classification tasks. It works by recursively splitting the data into subsets based on feature values, creating a tree-like model of decisions.[7] In our implementation, we have focused on enhancing the traditional DT approach by incorporating various cost functions and optimizing the feature selection process.

The DT implementation is characterized by a structured approach where each node represents a data subset, storing essential details such as data indices, class probabilities, and child node information. The nodes identify the best split in the data using three distinct cost functions: Misclassification Cost, Entropy, and Gini Index, each evaluating splits differently based on class probabilities and data impurity. A greedy algorithm selects the optimal split at each node by minimizing the cost. The tree growth halts when a node reaches the maximum depth or contains fewer instances than a set threshold. For predictions, the DT navigates from the root to a leaf node, using the test instance's features, and uses the class probabilities at the leaf for final decision-making.

Some advanced features are incorporated on top of the provided reference code. Functions like count features and feature cost are essential for analyzing the usage frequency and associated costs of different features within the decision tree. They offer valuable insights into feature importance and efficiency, enhancing the tree's analytical capabilities. Additionally, the code efficiently calculates and stores the cost at each leaf node. This is a critical component for informed decision-making during the tree's construction. In summary, these enhancements improve the decision tree's functionality and decision-making process, providing a more insightful analytical tool.

The Decision Tree implementation provides a robust framework for binary classification, enhanced by strategic feature analysis and customization options. This approach not only improves the model's accuracy but also provides insights into the feature dynamics of the datasets.

3 Datasets

We analyzed our dataset in a clear route: obtain general characteristics by descriptive statistical analysis, investigate detailed features of the data by utilizing visualization strategy and intuition(appealing to intuitionism), investigate the dynamics between features(carrying out feature engineering), and finally use our previous analysis as the feature filter to reduce the dimensionality of the data.

We firstly carried out the data cleaning process through the embedded function in pandas. By utilizing '.describe()' and '.dropna()', we can verify whether there's missing values by checking if all the features has the same size. After removing the missing values, we also did some class modification based on the source of the dataset. Duplicates are removed after that by employing '.drop_duplicate()' function. We have also set the meaningful index of the dataset, though it does not affect further data process. After all these preliminaries, our dataset is cleaned and finally able to carry on analysis.

After removing the missing values and the duplicates, in the descriptive statistical part, we utilized the ‘.describe()‘ function in pandas to acquire a glimpse. For the general distribution of the data, we were tilted to think data_set_1 is more skewed(actually bimodal) by observation and the KDE-graph we created, which suggests P-R curve should be a better measure of data_set_1; data_set_2's distribution shape is more balanced. In degree of deviation, we noticed that data_set_1 tends to reveal more volatility since the standard deviation in some features is usually high ($\sigma/\overline{x} > 0.4$), which also reflects in some features of data_set_2. Apart from that, both features include outliers, implying that further outliers should be analyzed. We did descriptive analysis for each class in each data set subsequently. The result suggests we should make a comparison between the distribution of different classes on each feature since some basic statistics vary greatly.

The analyses mentioned above justify the importance of carrying out comparison and outlier analysis. Since our purpose for learning the data is for cleaning and utilizing them, we concomitantly implemented the idea of feature engineering here, that is, investigating the dynamics between features and labels. The existence of outlier and the unknown distribution of the population suggests using a violin graph since it intuitively shows the outliers, the distributions, and the median-central tendency measure which is not greatly influenced by outliers[8]. The violin graph of data_set_1 indicates there are massive outliers existing. Outlier frequency comparison between different classes indicates outlier should not be excluded since it also contains differential information. Thus, by this and the shape comparison, we selected some features. In data_set_2, we did the same but found many significant features in explaining the differences in labels. We further carry out variance analysis, generating a list of features significant in predicting the label. This variance analysis is carried out in the version of calculating the squared mean difference(SMD), whose rationale is higher correlation corresponds to higher deviation. And it shows that RIAGENDR and Clump_thickness are the features with highest SMD. Detailed results can be found in Task1 section from our code. By computing them, we ranked them, with generating a list of features considered ought to be included in our feature space.

Finally, we concluded our analysis by conducting a covariation analysis using correlation through heat maps. Treating the previous analyses as filters, with the covariation analysis, we finally selected the features. One noticeable point is that data_set_2 indicates high covariation between many features, supporting our strategy for using Mahalanobis distance as the similarity measure.

4 Results

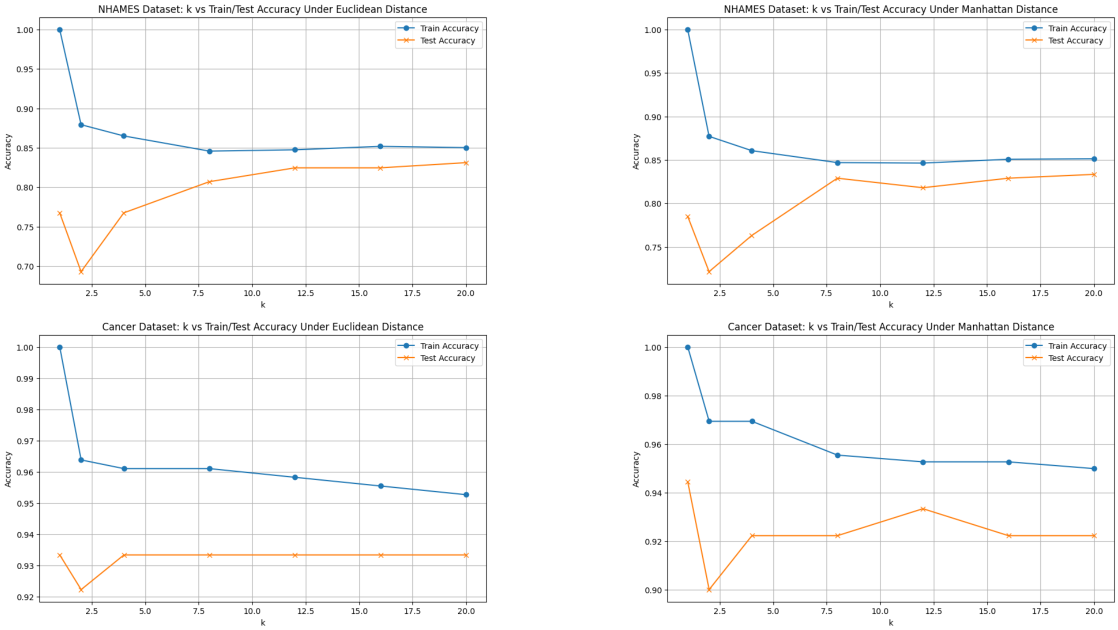


Figure 1: Influence of K values on two datasets

Firstly, as a bold trial, we tested out the KNN model with K = 11 and the Decision Tree model with depth = 4 (which is around the optimal choices of K and depth shown later). For the NHAMES dataset, the accuracy for this KNN model is 84.2, and for the DT model is 83.6, while the AUROCs are 0.56 and 0.68 for KNN and DT respectively. This result shows although KNN gives out more true positives, DT is more stable for all thresholds for this dataset. Similar behavior is observed in the cancer dataset, with (93.3,0.936) and (91.2,0.962) as (accuracy, AUROC) pair for KNN and DT respectively.

Then we checked out the influence of different K values for KNN with Euclidean distance and Manhatten distance on both datasets, which is shown in the figures above. The trends are similar in all four figures: as K increases, the accuracy of the training set decreases, while the accuracy of the test set increases in general and becomes stable. Notice for the NHAMES dataset, Manhattan distance gives slightly better performance, which might imply that the geometry of the natural grouping of data is more aligned with axes. The accuracy under different K for weighted KNN mentioned in section 2.1 is shown in Figure 2. The trends using the two datasets are opposite, suggesting that NHAMES might have a tighter cluster while clusters in the Cancer dataset are interleaving.

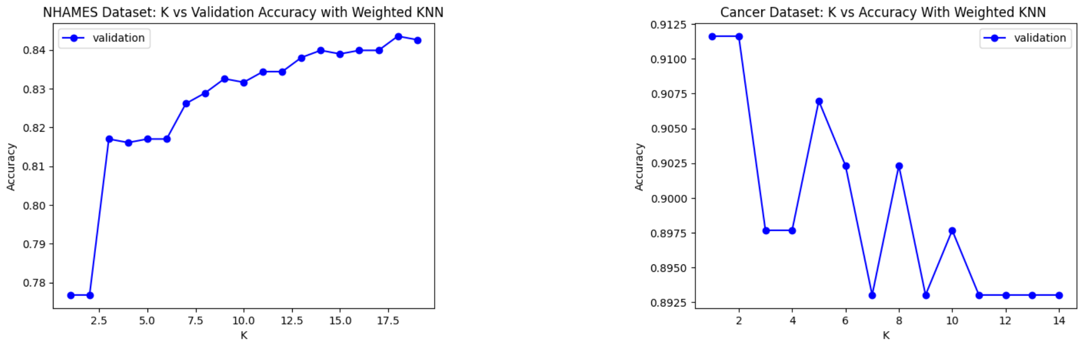


Figure 2: Influence of K values using weighted KNN

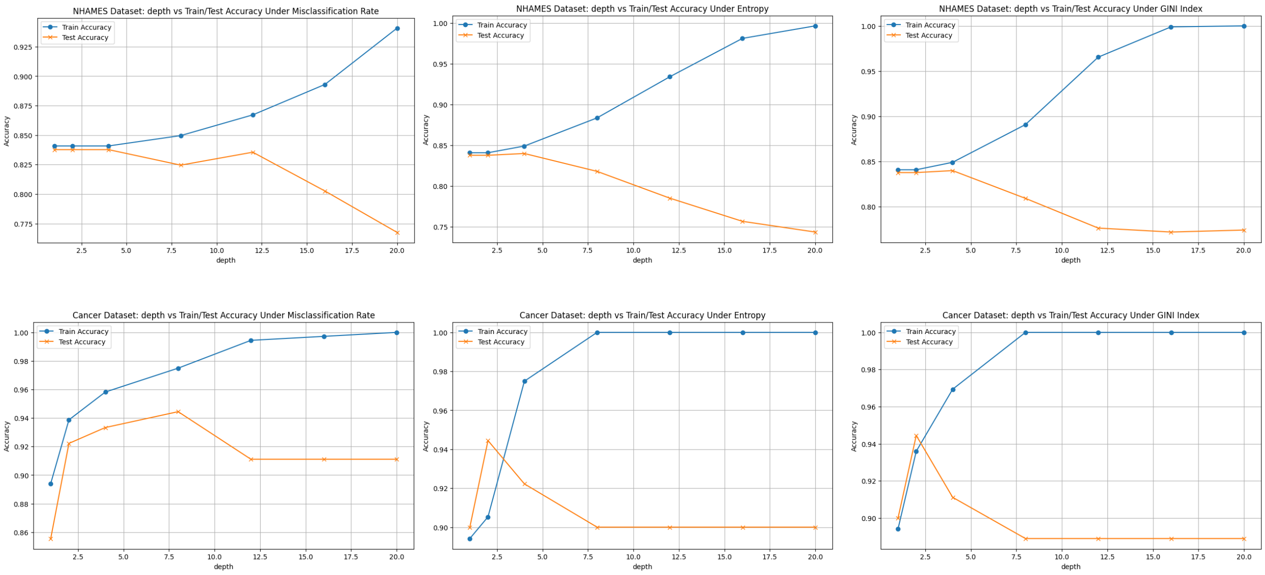
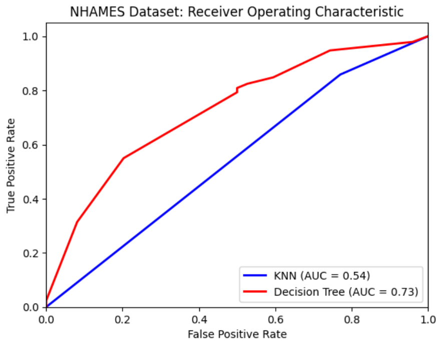
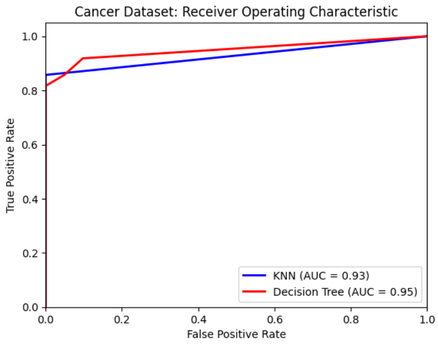


Figure 3: Influence of depth values on two datasets

We also checked the influence of different depths for DT with various loss functions on both datasets, as shown below. With a deeper tree, it is more likely to overfit as the classification is more fine-grained, which causes training accuracy to increase but test accuracy decreases after some threshold. It also shows that the GINI index is more sensitive to tree depth since a more complicated tree causes the purity of output on test data to drop more dramatically.

Next, we first split half of the training data into a validation set to find the best K resp. depth for KNN resp.DT. Using the hyperparameters found, we generate the ROC for both datasets as shown in Figure 4. For both datasets, DT outperforms KNN with a higher AUC, suggesting it's more stable for both tasks. Another observation is both models' AUCs for NHAMES are just slightly above 0.5, indicating the imbalanced characteristics of NHAMES affect the model to identify the minority class instances less accurately.





Lastly, regarding feature importance, the strategies we applied for KNN and DT are different. For KNN, external feature selection are carried out by calculating the correlations, described in Section 3. For DT, we first calculate the rough feature importance score for each feature by counting the number of non-leaf nodes where feature d is used and then calculate the weighted sum of the reduction. In addition, the latter approach will indicate the rough feature importance, since only if a feature is used as a threshold, it will show up in the weighted sum, so the following result is reported from the weighted sum method. For the NHAMES dataset, the top five important features are RIAGENDR, BMXBMI,LBXGLU,LBXGLT and LBXIN ;Clump_thickness, Bland_chromatin, Uniformity_of_cell_size and Uniformity_of_cell_shape(only four are used) for Cancer Dataset. Most important features rank high in the simple mean difference calculation. The only exception is Bland_chromatin. It could be because the greedy algorithm finds a local optimum and this feature outperforms the others in such a combination of features already decided higher on the tree.

5 Conclusion and discussion

The study's results show that both KNN and DT models perform well, but their effectiveness varies with different datasets. Decision Trees showed more consistent performance across various thresholds in the NHANES dataset, while KNN achieved higher accuracy in the Breast Cancer Wisconsin dataset, indicating its adaptability. These findings highlight the significance of choosing the right algorithm based on specific dataset characteristics. Additionally, experimenting with hyperparameters such as the number of neighbors in KNN and tree depth in DTs provided crucial insights for optimizing these models.

The key takeaways from this assignment revolve around experiencing the entire data pre-processing lifecycle, encompassing data cleaning and analysis of feature correlations within two real-world datasets: the NHANES Age Prediction Subset and the Breast Cancer Wisconsin Dataset. We also delved into the implementation and refinement of two fundamental machine learning algorithms, K-Nearest Neighbour and Decision Trees, and applied these enhanced models for comprehensive data analysis. This process not only enhanced our technical skills in machine learning model implementation but also deepened our practical understanding of handling and interpreting complex datasets.

Potential directions for future investigation could include exploring more advanced machine learning techniques to improve predictive capabilities on these datasets with the integration of feature engineering techniques to further enhance model performance. Additionally, the development of hybrid models that combine the strengths of both KNN and DTs could potentially achieve higher accuracy and stability. These explorations would not only advance our understanding of model applicability but also contribute to the broader field of predictive analytics in healthcare.

Besides, we have noticed that the second dataset reveals a strong inter-correlation property, indicating some formal dimensional reduction process may be useful such as PCA, SVD, etc.. More statistical descriptive methodology could also be considered to implement for better intuitiveness and discovering major features.

6 Statement of Contributions

This section outlines the contributions of each team member to the project.

Terrance Yan

- Carried out Data analysis and cleaning (Task 1)
- Implemented Weighted KNN and Wrote the corresponding part in the report

Yunjia Zheng

- Suggest team roles and responsibilities
- Implementation and writing report for Task 3.

Annie Kang

- Implemented DT (Task 2) and wrote the corresponding part of the report. Wrote Introduction and Conclusion
- Report template set up and Grammar correction

References

[1] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[2] *National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset*, UCI Machine Learning Repository, `https://doi.org/10.24432/C5BS66`, 2023. DOI: `10.24432/C5BS66`.

[3] A. Dinh, S. Miertschin, A. Young, and S. Mohanty, "A data-driven approach to predicting diabetes and cardiovascular disease with machine learning," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, pp. 1–15, 2019. DOI: `10.1186/s12911-019-0918-5`.

[4] W. Wolberg, *Breast Cancer Wisconsin (Original)*, UCI Machine Learning Repository, `https://doi.org/10.24432/C5HP4Z`, 1992. DOI: `10.24432/C5HP4Z`.

[5] A. Dubey, U. Gupta, and S. Jain, "Analysis of k-means clustering approach on the breast cancer wisconsin dataset," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 11, pp. 2033–2047, 2016. DOI: `10.1007/s11548-016-1437-9`.

[6] H. Li, *Machine Learning Methods*, trans. by L. Lu and H. Zeng. Springer Singapore, 2023. DOI: `10.1007/978-981-99-3917-6`. [Online]. Available: `https://doi.org/10.1007/978-981-99-3917-6`.

[7] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, 1991. DOI: `10.1109/21.97458`.

[8] J. Jia, X. He, and Y. Jin, *Statistics, Seventh Edition*. Renmin University of China Press, Jan. 2018, ISBN: 9787300253510.