COMP 551 Assignment 1 Write-up

Faith Ruetas, Joey Marten, and Gary Zhang

October 11, 2022

Abstract

We implemented the K-Nearest Neighbours (KNN) and Decision Tree (DT) supervised learning algorithms, then ran our models on the Hepatitis and Diabetic Retinopathy Debrecen (Messidor) datasets. After pre-processing the data and training the models, both algorithms performed best on the smaller-sized Hepatitis dataset. This occurred for the KNN algorithm with the hyperparameters of K = 5 and the Cosine distance function. For the DT model, the hyperparameters of max depth = 1 and the Entropy cost function yielded the best result. In general, KNN was more accurate across both datasets whereas DT was more consistent when comparing its training, validation, and testing accuracies.

1 Introduction

In machine learning projects, it is common to rely on publicly available classes such as Sklearn's KNeighborsClassifier and DecisionTreeClassifier. For this assignment, by contrast, our task was to implement the K-Nearest Neighbours (KNN) and Decision Tree (DT) supervised learning algorithms ourselves. We used the Hepatitis and Diabetic Retinopathy Debrecen (Messidor) datasets to evaluate the accuracy of our models. The first dataset tracks the survival rate of hepatitis patients based on features such as age and sex; the second dataset tracks signs of diabetic retinopathy in Messidor images alongside features such as quality assessment level and optic disk diameter.

Overall, both models performed best on the smaller-sized Hepatitis dataset: the KNN model attained 96.15% accuracy using K = 5 and the Cosine distance function, and the DT model attained the same accuracy using max depth = 1 and the Entropy cost function. In general, the KNN model performed better across both datasets whereas the DT model was more consistent in its training, validation, and testing accuracies.

There is ample prior work on classifying health-related data using supervised learning algorithms. In "Applying k-Nearest Neighbour in Diagnosing Heart Disease Patients," Shouman et al. find that KNN is more accurate than neural network ensemble at diagnosing heart disease patients [2]. In another article, "Classification Tree Analysis: A Statistical Tool to Investigate Risk Factor Interactions with an Example for Colon Cancer (United States)," Camp and Slattery posit DT as a particularly effective model for investigating multilevel interactions, such as risk factors that combine to induce disease [1]. In short, our project follows pre-existing scholarship on KNN and DT, though with different datasets.

2 Methods

2.1 K-Nearest Neighbours (KNN)

The KNN algorithm is an exemplar-based, non-parametric method: given a new data point, it predicts the label (or classification) of that instance based on the labels of the K most similar examples in the training set. The model's hyperparameters are thus the value for K and the function used to evaluate the distance (or similarity) between instances.

First, the algorithm calculates the distance from some instance in the testing data to each instance in the training data. The indices of the K smallest distances are saved and the labels of these closest data points are used to calculate the probability of the test point belonging to each class. To predict a data point's label, the model chooses the class with the highest probability.

2.2 Decision Tree (DT)

The DT algorithm uses the training data to create a full binary tree. The model's hyperparameters are thus the maximum depth at which the binary tree stops splitting the data and the cost function used to determine the purity of each split.

First, the algorithm divides the data using a condition on a feature and a threshold at each node. The portion of the data satisfying the condition enters one of the child nodes and the remaining data enters the other. Conditions are chosen by minimizing the split cost with respect to the choice of feature and threshold. For each node, predictions are made by taking the average or max argument of the labels of the data points in that region.

3 Datasets

3.1 Hepatitis

The Hepatitis dataset comprises 155 instances spanning 19 features (excluding the classes, or labels). Based on traits such as age and histology, the data tracks whether a patient with hepatitis survives.

Aside from standardizing the data for categorical and continuous features (e.g., using the boolean values 0 and 1 instead of 1.0 and 2.0 for the former), we pre-processed the data using two methods to remove malformed instances. On the original dataframe, we deleted 75 rows with occurrences of '?' as feature values. For our second approach, we used imputation to keep as many instances as possible: we created a copy of the original dataframe on which we replaced each occurrence of '?' with the relevant column's mode for categorical columns and mean for continuous-valued columns. To remove outlier data points, we removed 1 instance in the original dataframe and 10 in the imputation dataframe with feature values exceeding 3 standard deviations above the respective feature mean.

3.2 Messidor

The Messidor dataset comprises 1151 instances spanning 19 features (excluding the labels). Based on traits in the Messidor image set, such as quality assessment level and optic disk diameter, the data tracks whether instances contain signs of diabetic retinopathy.

To pre-process the data, we did not need to remove any rows due to malformed data. That being said, we once again mapped categorical value instances of b'0' and b'1' to the boolean values 0 and 1, respectively, in the labels. To remove outlier data points, we removed all instances with feature values exceeding 3 standard deviations above the respective feature mean.

3.3 Analysis

For the Hepatitis data, the SGOT, ALK Phosphate, and Protime values were most varied whilst it was Exudates CL 0.5, MA Detection CL 0.5, and MA Detection CL 0.6 for the Messidor data. We also plotted the feature distributions to better visually understand the datasets (see Fig 1 and 2).

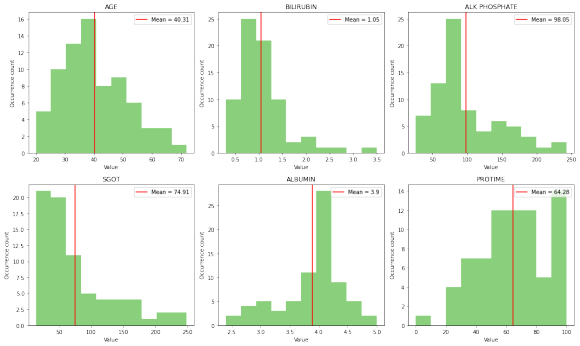

Figure 1: Histograms for the continuous-valued features in the Hepatitis data mostly cluster towards the lower/middle regions of the value range.

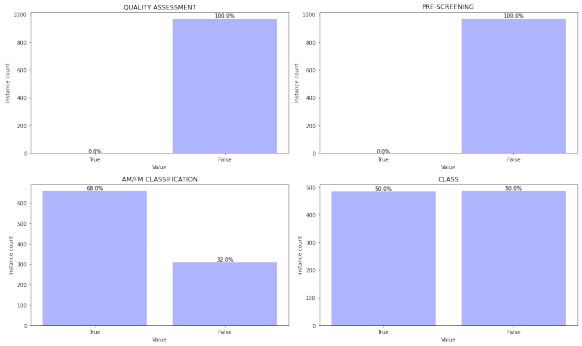

Figure 2: Categorical feature distributions for the Messidor data show that Quality and Pre-screening are all composed of one value.

4 Results

4.1 Baseline tests

For our baseline tests, we employed a KNN model with K = 3 and the Euclidean distance function, and a DT model with max depth = 3 and the Misclassification cost function. The data was split into 2/3 training and 1/3 testing.

	KNN (K=3) training accuracy	KNN (K=3) test accuracy	DT training accuracy	DT test accuracy
Hepatitis	85.42%	88.46%	95.83%	88.46%
Messidor	82.45%	62.73%	69.44%	62.42%

4.2 Testing different K values

As expected, the training accuracy for both datasets begins at 100% when K = 1.

For the Hepatitis dataset (Fig 3), this accuracy drops as K increases, then plateaus for the rest of the values. The inverse occurs for the testing data: after starting below 60%, the accuracy increases as K increases before plateauing as well. This plateauing effect for higher K values could be attributed to the small size of the Hepatitis dataset.

For the Messidor data (Fig 4), the testing set performs on par with the validation set but results in a higher test accuracy than the baseline test (K = 3).

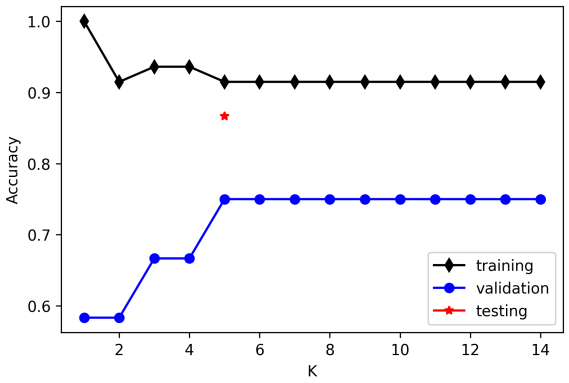

Figure 3: K values for the Hepatitis data.

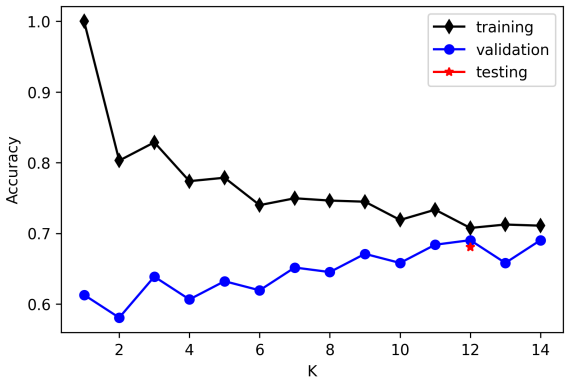

Figure 4: K values for the Messidor data

4.3 Testing different maximum depth values

After running DT on a range of maximum depth values, the testing accuracy demonstrated a clear pattern: as the maximum tree depth increased, the training accuracy increased while the validation accuracy stayed relatively the same. This makes sense as the greater the maximum depth, the more the data is split into nodes, creating smaller data regions and likely leading to overfitting.

For the Hepatitis data (Fig 5), the best value (max depth = 1) performs slightly better than our baseline accuracy (max depth = 3). For the Messidor data (Fig 6), the test performance on the best value (max depth = 10) is, again, on par with the accuracy of the validation set.

4.4 Testing KNN on different distance functions

When experimenting with different distance functions, the highest overall testing accuracy occurs with the Cosine distance function. The Manhattan distance performs lower overall on the testing data, but higher overall on the training data than both of the other functions.

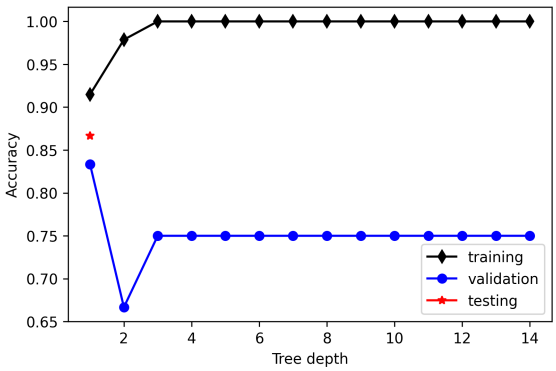

Figure 5: Best depths for the Hepatitis data.

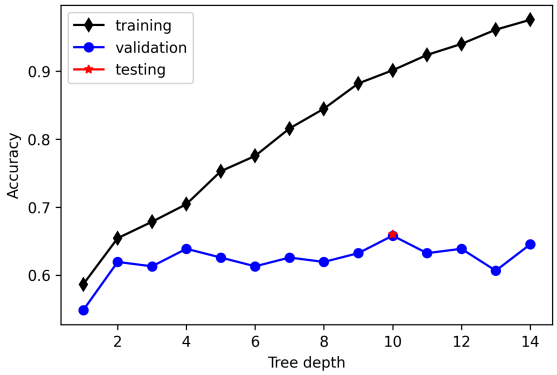

Figure 6: Best depths for the Messidor data.

Hepatitis data

	Euclidean (training)	Euclidean (testing)	Manhattan (training)	Manhattan (testing)	Cosine (training)	Cosine (testing)
K=1	100%	84.62%	100%	84.62%	100%	88.46%
K=3	85.42%	88.46%	89.58%	88.46%	87.5%	92.31%
K=5	85.42%	92.31%	85.42%	88.46%	85.42%	96.15%

Messidor data

	Euclidean (training)	Euclidean (testing)	Manhattan (training)	Manhattan (testing)	Cosine (training)	Cosine (testing
K=1	100%	59.7%	100%	61.21%	100%	63.33%
K=3	82.45%	62.73%	80.72%	63.94%	82.13%	63.94%
K=5	77.12%	64.55%	78.06%	63.64%	77.12%	66.06%

4.5 Testing DT on different cost functions

Hepatitis data and reporting testing accuracy

Though the cost functions perform relatively similarly, the Entropy cost function performs better at lower maximum depth values and the Misclassification cost function at greater maximum depth values.

Max Depth	1	2	3	4	5
Misclassification	92.31%	92.31%	88.46%	92.31%	92.31%
Gini cost	92.31%	92.31%	84.62%	88.46%	88.46%
Entropy	96.15%	92.31%	88.46%	88.46%	88.46%

Messidor data and reporting testing accuracy

The Gini and Entropy cost functions perform similarly while the Misclassification cost function performs marginally worse.

Max Depth	1	2	3	4	5
Misclassification	58.48%	61.21%	62.42%	61.82%	61.21%
Gini cost	58.48%	61.52%	63.03%	63.03%	63.64%
Entropy	58.48%	61.52%	63.33%	63.03%	62.73%

4.6 Decision boundaries

To plot the decision boundaries of our experiments, we ran our algorithms on the top two features correlated with the labels for each dataset (calculated using pairwise column correlation). In the figures below, we see that the boundaries drawn by KNN (Fig 7) allow for more pockets of classification, such as the yellow areas within the purple around Optic Disc Diameter = 2 that DT (Fig 8) classifies entirely as purple.

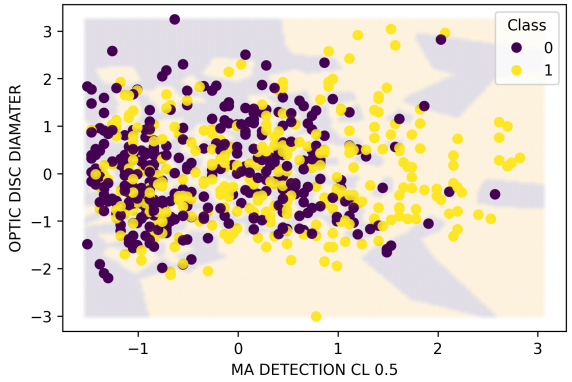

Figure 7: KNN on the Messidor data (54.55% testing accuracy)

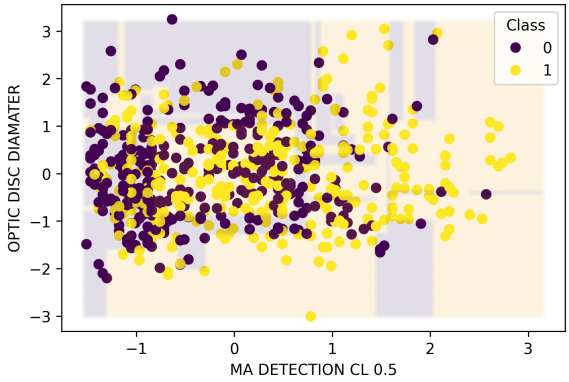

Figure 8: DT on the Messidor data (58.79% testing accuracy)

4.7 Key features

To determine key features for KNN and DT, we evaluated how each algorithm weighs the input features when predicting its outputs (the labels):

$$knn/dt.fit(x_{train}, y_{train}).predict(x_{train}) = \hat{y}, Importance = RandomForest.fit(x_{train}, \hat{y}).featureImportances$$

For the Hepatitis data, the key features are Protime and Albumin based on the KNN-predicted labels, and Protime and Bilirubin based on the DT-predicted labels. Both models demonstrate that Protime is a key feature in making predictions.

For the Messidor data, the key features are MA Detection CL 0.5 and Exudates CL 0.5 based on the KNN-predicted labels, and MA Detection CL 0.5 and Exudates 0.93 based on the DT-predicted labels. Both models demonstrate that MA Detection and Exudates are key features in making predictions.

5 Discussion and conclusion

In summation, we find that the KNN algorithm generally performs more accurately on both datasets while the DT algorithm performs more consistently between its training, validation and testing accuracies. We reason that this may be due to KNN's tendency to split data more locally, allowing for pockets of one class within another, as well as its tendency to overfit training data with low K values.

As K increases, KNN's training accuracy plateaus or decreases while its testing accuracy plateaus or slightly increases. As the maximum depth value increases, DT's training accuracy increases while its testing accuracy plateaus or slightly decreases.

In the future, possible directions of exploration include implementing weighted KNN and comparing its results with the standard KNN algorithm, using dimensionality reduction to graph experiments in 2D, and employing linear regression instead of the Random Forest algorithm to extract key features. Since the Hepatitis dataset was particularly small, we repeatedly obtained certain accuracy percentages; to avoid this on similarly restricted datasets, K-fold cross-validation could present an alternative method of experimentation.

6 Statement of contributions

Our work was generally divided as follows:

- Faith: pre-processing data, KNN algorithm code, encapsulating code, merging work on Github, final proofreading and editing for write-up
- Gary: class code analysis, helper functions, code quality checking, LaTeX editing for write-up
- Joey: DT algorithm code, experiments, code quality checking, Google Colab table of contents organization

References

[1] Nicola J. Camp and Martha L. Slattery. "Classification Tree Analysis: A Statistical Tool to Investigate Risk Factor Interactions with an Example for Colon Cancer (United States)". In: *Cancer Causes Control* 13.9 (2002), pp. 818–823.

[2] Mai Shouman, Tim Turner, and Rob Stocker. "Applying k-Nearest Neighbour in Diagnosing Heart Disease Patients". In: *International Journal of Information and Education Technology* 2.3 (2010), pp. 220–223.

A Extra experiments

A.1 Testing normalized and standardized data

	Hepatitis data (normalized)	Hepatitis data (standardized)	Messidor data (normalized)	Messidor data (standardized)
KNN	96.15%	96.15%	51.52%	51.52%
DT	65.38%	84.62%	61.82%	64.24%

A.2 Testing the models on important features

A.2.1 Top 2 important features based on Random Forest

	Hepatitis data -Protime -Bilirubin	Messidor data -MA Detection CL 0.5 and -Optic Disc Diameter
KNN	96.15%	54.55%
DT	96.15%	58.79%

A.2.2 Top 2 correlated features (using pairwise column correlation)

	Hepatitis data -Albumin -Histology	Messidor data -MA Detection CL 0.5 -Exudates CL 0.57)
KNN	92.31%	52.12 %
DT	96.15%	60.91%

A.2.3 Top 2 important features for Random Forest with KNN-predicted labels (see section 4.7)

	Hepatitis data -Protime -Albumin	Messidor data -MA Detection CL 0.5 and -Exudates CL 0.5
KNN	96.15%	62.12 %

A.2.4 Top 2 important features for Random Forest with DT-predicted labels (see section 4.7)

	Hepatitis data -Protime -Bilirubin	Messidor data -MA Detection CL 0.5 and -Exudates CL 0.93
KNN	96.15%	60.91 %