Using K-Fold Cross Validation on Decision Tree and Logistic Regression Models to Classify Iris Species

Lillian Mueller lmuelle1@umd.edu
Regina Hong rhong@umd.edu

***Abstract*— Determining if a logistic regression model or a decision tree model is more effective in classifying Iris species is an example of a case where performing k-fold cross validation may be appropriate. After evaluating each model individually as a classifer for the iris dataset, the performance of each models was then examined via cross validation. It was found that the logistic regression model consistently performed with a higher accuracy score. The effects varying k values also had effects on the projected accuracy of the two models. Being able to validate developing models is very important for model implementation and application. Examining the behavior of the cross validation method will aid future model development and applications.**

I. Introduction

Being able to evaluate the effectiveness of a prediction model is just as important as the development. One method that is conventionally used to determine the skill of a model is cross validation. This method involves breaking the dataset up evenly into a number of random sample groups, or folds, and systematically training a model of all groups but one of the groups and testing the model of the remaining group until all groups have been used as a separate testing set. The accuracy scores for each model are then used to evaluate the effectiveness of the model for that dataset. Cross validation is commonly used to determine the optimal model for a dataset because it is fairly easy to implement and understand [1].

Cross validation is a method that can be used to determine which classification model is a better predictor of the Iris dataset: the decision tree classification model or the logistic regression classification model. These models were evaluated separately in previous investigations where modifications of these methods were examined to optimize accuracy. The model with parameters that yielded the greatest accuracy for this dataset was selected for this investigation. Specifically, the decision tree model using entropy as a purity measure will be compared to a logistic regression model without any penalties [2], [3]. As in previous reports, these models will be predicting the class of each iris flower based on the following features: sepal length, sepal width, petal length, and petal width.

The Python `sklearn` library is used to perform cross validation in section II to conclude the optimal model between decision tree and logistic regression for the Iris dataset. The results are outlined in section III. The importance of this investigation is in section IV.

II. Methodology

This investigation entails using `SKLearns`'s `cross_validate` function from the `model_selection` module to compare the accuracy between a decision tree model and logistic regression model. Several Python packages were utilized including `sklearn`, `pandas`, and `numpy`. Specifically from `sklearn`, we use the following modules: `linear_model`, `preprocessing`, `model_selection`, `metrics`, and `tree`.

First, the Iris dataset was loaded as an `sklearn.utils.Bunch` object called `iris_data` via the `load_iris()` function. This object is similar to a Python dictionary. To make the data easier to read, it was transformed into the `df_iris` dataframe, where the data parameter was set as the `data` attribute of the dataset and the columns parameter as the `feature_names` attribute. A new column called "class" was added to this dataframe which contains the `target` variable of the Iris dataset; this is the class of the Iris plant - setosa, versicolor, or virginica. Since the `target` attribute contains an array with values from 0-2, the `.replace` function was used to map these numerical values to their corresponding classifications: setosa for 0, versicolor for 1, and virginica for 2.

After loading in and setting up the Iris dataset, cross validation was performed on both a decision tree and logistic regression model, which were built using the same `sklearn` functions as in our previous studies [2], [3]. The `criterion` parameter of the decision tree model was set to `entropy` and the `penalty` parameter in the logistic regression model was set to `None`.

The type of cross validation used is called k-fold cross validation. This kind of cross validation involves splitting the data into k groups or folds, using k-1 folds as training data, and using the remaining fold as test data. Some sort of measurement or calculation is done (in this case we chose to use accuracy) to validate the model. These steps are repeated until every fold has become test data and the mean accuracy value is computed. Figure 1 visualizes this concept with k=5 folds.

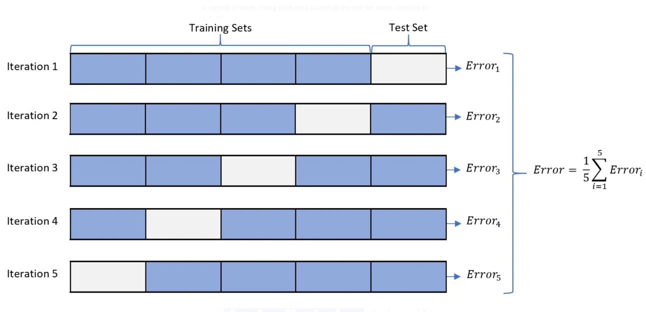

Fig. 1. Example of k-fold Cross Validation with k=5 folds [4]

In the cross validation process for both the decision tree and logistic regression models, the `cv` parameter was set to `None` meaning it uses the default of k=5 folds, and the `scoring` parameter was set to `accuracy` since that was the strategy chosen to determine the effectiveness of the models. The mean and standard deviation of the cross validation of both models was determined.

To validate the `cross_validate` function from sklearn, we also created our own cross validation function called `perform_manual_cross_validation` that goes through the same algorithm of splitting the Iris data into folds, using one fold as the test data, and iterating through until all the folds have been used as test data. We ran this function on the same decision tree and linear regression models; the accuracy each iteration was printed and the mean and standard deviation of those accuracies were compared to when the `cross_validate` function was used.

Returning to using the `cross_validate` function we changed the `cv` parameter, setting it to the values of 3, 7, and 10 to see how the accuracy of the cross validated decision tree and linear regression models would change. We were especially interested in seeing how setting k=10 would affect accuracy because k=5 and k=10 are the more common fold values.

III. Results

The mean and standard deviation accuracy values from performing 5-fold cross validation on our logistic regression and decision tree models are shown in Table I.

Measure	Logistic Regression	Decision Tree
Mean	0.973333	0.960000
Standard Deviation	0.036515	0.036515

TABLE I

5-Fold Cross Validation Statistics for Logistic Regression and Decision Tree Models

It can be seen that the logistic regression model is slightly more accurate than the decision tree model, at 0.973 compared to 0.960. This lines up with our previous study; when the results of that linear regression model were briefly compared to that of the decision tree model, the logistic regression model was more accurate. Here, we used a more robust method of evaluating effectiveness of the models and found the same conclusion to be true.

The results of our own k-fold cross validation function `perform_manual_cross_validation` produced similar results, seen in Table II.

Measure	Logistic Regression	Decision Tree
Mean	0.953333	0.933333
Standard Deviation	0.038006	0.0.062361

TABLE II

5-Fold Manual Cross Validation Statistics for Logistic Regression and Decision Tree Models

Although the values are not exactly the same as when the `cross_validate` function was used, it still confirms that the logistic regression model is slightly more accurate than the decision tree model. The higher standard deviation values may be attributed to the way the data is being split into the folds; it is just splitting the Iris dataset into fifths without implementing a random order.

The results of performing k-fold cross validation using sklearn and changing the number of folds are shown below. The additional k values of 3, 7, 10, 15, and 50 are used.

k	Measure	Logistic Regression	Decision Tree
3	Mean	0.973333	0.96
	Standard Dev.	0.023094	0.02
5	Mean	0.973333	0.96
	Standard Dev.	0.036515	0.036515
7	Mean	0.959802	0.939703
	Standard Dev.	0.074425	0.065443
10	Mean	0.98	0.953333
	Standard Dev.	0.044997	0.044997
15	Mean	0.980000	0.960000
	Standard Dev.	0.041404	0.063246
50	Mean	0.980000	0.946667
	Standard Dev.	0.104545	0.123443

TABLE III

Cross Validation Statistics for Logistic Regression and Decision Tree Models (k=3, 5, 7, 10, 15, 50)

For all k-values, the logistic tree models consistently has greater accuracy than the decision tree model. Between k=3 and k=5, the accuracy of both the models did not change. When k=7, the accuracy of both models decreased. This could be because the folds do not have the same amount of data, as 150 data points do not divide evenly between 7 groups, also causing more variance within the accuracy of the folds. When k=50, the accuracy for the decision tree decreases which indicates underfitting of the data, the number of data points in each fold must be too low to effectively train a decision tree model. The standard deviation also greatly increased for both models when k=50. When k=10 and 15, the accuracy of both models increases slightly compared to k=3 and k=5. Compared to k=50, the models (evaluated where k=10, 15) have much lower standard deviations. The methods with k=10 and 15 have the highest accuracies and relatively low standard deviations which indicates that 10-15 folds is the optimal number of folds to validate this dataset.

Evaluating the method at k=10 through 15, we concluded that k=14 would be the optimal k value for this validation procedure as it produced high accuracy and low variance indicating it was unbiased and balanced.

Measure	Logistic Regression	Decision Tree
Mean	0.980519	0.953896
Standard Dev.	0.038710	0.047901

TABLE IV
K=14 CROSS VALIDATION STATISTICS

IV. DISCUSSION

Through this investigation, we were able to confirm that the logistic regression model was more effective as a predictive model for the Iris dataset as compared with a decision tree model. Here, we used a more robust method of evaluating effectiveness of the models and found the same results. K-fold cross validation allows for a relatively quick confirmation of which model in question is more accurate as compared to other models.

K=10 is commonly used and may be found to work better for other datasets. The accuracy of both testing and training data could be observed to determine if overfitting or under-fitting is occurring. In general, lower k values can lead to overfitting and high variance whereas higher k values can lead to underfitting and higher bias [5]. This is why evaluating multiple k values is so important. For this study, we were able to determine a balance.

In the future, the Iris dataset could be split into test and train partitions, more similar to what was done in previous studies. Then, k-fold cross validation would only be performed on the training data and the results of that would finally be applied to the testing data. Additionally, rather than using k folds where each group consists of randomly selected samples, expanding this investigation to encompass stratified sampling and portioning could be beneficial and decrease standard deviation of the accuracy scores.

REFERENCES

[1] J. Brownlee, “A Gentle Introduction to k-fold Cross-Validation,” MachineLearningMastery.com. Accessed: Oct. 01, 2023. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/

[2] L. Mueller and R. Hong, “Investigating Decision Trees,” Sep. 18, 2023.

[3] L. Mueller and R. Hong, “Iris Classification Using Logistic Regression,” Sep. 25, 2023.

[4] R. Patro, “Cross Validation: K Fold vs Monte Carlo,” Medium. Accessed: Oct. 05, 2023. [Online]. Available: https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b

[5] “How do you avoid overfitting or underfitting when using grid search and k-fold cross validation?” Accessed: Oct. 06, 2023. [Online]. Available: https://www.linkedin.com/advice/3/how-do-you-avoid-overfitting-underfitting-when