Classification of Leaves Based on Image Data

Elijah Horowitz, Hayden Spinos

CSCI 4520 Machine Learning

Dr. Hong Zhang

December 6, 2021

Table of Contents

1. **Data**

Our data set comes from the UCI Machine Learning Repository, it's called "Leaf Data Set" and it includes information about leafs that can be used to classify the leafs into their species. It has 340 different individual leaves each with 16 attributes of information. These attributes are,

Species - The actual species number of the leaf.

Specimen Number - The specimen of each leaf is of a certain species.

Eccentricity- The Eccentricity of the ellipse with identical second moments (1).

Aspect Ratio- Aspect ratio is defined as the quotient, values close to 0 indicate an elongated shape

Solidity- Measures how well I fits a convex shape.(1)

Stochastic Convexity- Extends the usual notion of convexity in topological scenes, using sampling to perform the calculation. Estimate the probability of a random segment.(1)

Isoperimetric Factor- The maximum value is 1 for circular regions, Curvy intertwined contours yield low values (1). Basically how round is the leaf.

Maximal Indentation Depth- Indentation function can be sampled at one degree intervals, maximal indentation depth is the maximum of this function.(1)

Lobedness- Calculated using the maximal indentation depth, characterizes how lobed a leaf is.(1)

Average Intensity- Defined as the mean of the intensity image(1)

Average Contrast- standard deviation of the intensity image(1)

Smoothness- Measures relative smoothness of the intensities in a given region, of constant intensity.(1)

Third Moment- Measure of the intensity histogram's skewness.(1)

Uniformity- Max value is reached when all intensity levels are equal.(1)

Entropy- A measure of intensity randomness. (1)

We only used 3rd through 16th attributes for our machine learning systems because the Species and the Specimen number were for humans to see when actually collecting and inputting data and weren't supposed to actually be a part of the machine learning systems. We also cut the data down because some species had a greater specimen number than others, so we got rid of any specimen number greater than 8 so that each species had the same number of specimens, we called this dataset leaf2.csv and used that for our machine learning systems, however the original dataset is still in the files if needed.

2. Objectives

Using the data from the publicly available Leaf dataset, our objective is to create a classification machine learning model that is able to correctly classify a leaf specimen based on its attributes.

3. Design

For our machine learning system, we used three different supervised learning algorithms since our data was labeled. For each of the three algorithms, we used two different model selection techniques to determine optimal performance for each model. Within each model selection technique, we also implemented flow control that would change a hyper-parameter of the algorithm being used. Each one of our models was implemented in its own script. The three models we used are listed below:

1. Gaussian Naive Bayes:
    a. 8-Fold Cross Validation

   b. Train_Test_Split (TTS) (test_size=.33)

2. Logistic Regression:
   a. Stratified K-Fold Cross Validation
   b. Train_Test_Split (TTS) (test_size=.33)
3. Support Vector Machine
   a. 8-Fold Cross Validation
   b. Train_Test_Split (TTS) (test_size=.33)

In order to view our results, we plotted the scores of each model against the respective tuned hyper parameter.

4. Training Process

Gaussian Naive Bayes (GNB)

For our GNB model, we used flow control to tune the hyper parameter "var_smoothing". We used a linspace of 100 values ranging from 0 to .001 to represent our var_smoothing values. These values were then used to generate a new model in each iteration where training and testing scores were recorded.

Logistic Regression (LR)

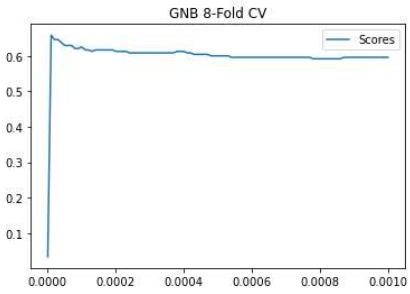
For our LR model, we chose to tune the C value hyper parameter. For our SKF cross validation set we used a linspace of 100 values from .1 to 5 and for our TTS set we again used 100 values but this time ranging from 50 to 60. Through trial and error of different ranges, these ranges highlighted the zones of optimal performance for the LR model.

Support Vector Machine (SVM)

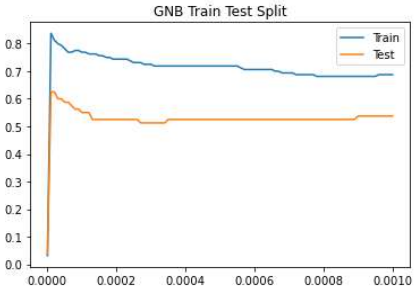
For our SVM models, we began by applying a standard scaler to the input dataset. On both model selection sets, we chose to also tune the C value hyper parameter. In the 8-Fold CV set, we used a linspace of 100 values ranging from 3.5 to 8 as well as the rbf kernel for all iterations. In the TTS model, we used a linspace of 100 values ranging from 1 to 50 to represent but a linear kernel instead. The linspace ranges were decided similarly to the previous model where the selected range best represented the optimal performance range. The kernel choices were also made through trial and error of score observations.

5. Results

Gaussian Naive Bayes (GNB)

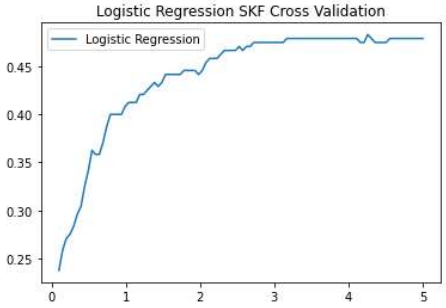


This is for GNB running with cross validation scoring, the peak score is just over .6 and is pretty constant.

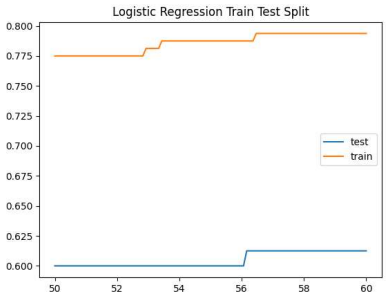


This is for GNB running with train test split, the training score peaks at .8 and the testing score peaks at 0.6, they both are pretty constant after the initial peak and drop off.

Logistic Regression (LR)

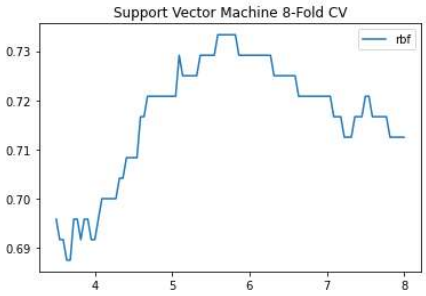


This is for LR running with SKF cross validation, the score is pretty low and peaks around 0.45. It stays pretty constant after 5.

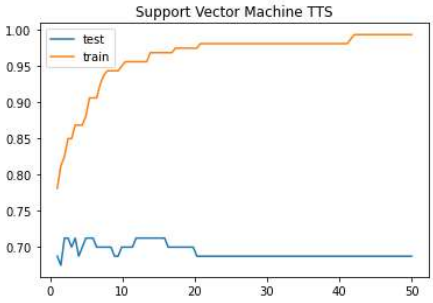


This is for LR running with train test split, the train scores peak at just above .775 and stay pretty constant. The Test scores peak at just over .6 and stay pretty constant.

Support Vector Machine (SVM)



This is for SVM with cross validation, the cross val scores peak around .73 and then taper off a little bit.



This is SVM with train test split, the train scores peak around 0.99 and the test scores peak around .72 they both stay pretty constant throughout.

Overall we found SVM to be the best machine learning model no matter which model selection we used, we found this because it had higher scores on average than the other machine learning models we used. We decided that GNB with train test split was our second best machine learning model and model selection combo, and after that there was a pretty stark fall off of the scores of all the other models and model selection combos. So it was pretty obvious to us that SVM was the best machine learning model to use for this data set.

Resources

(0)UCI Machine Learning Repository: Leaf data set. (n.d.). Retrieved December 5, 2021, from https://archive.ics.uci.edu/ml/datasets/leaf.

(1)Read Me- Data Set Description On UCI page