Leaf Classification

Team Name: TTU\_RashNParth

Team Members:

Partha S Satpathy

Rashmita Rout

**Problem Discussion:**

There are various species of plants in our ecosystem. To classify these plants, we must identify them. The best way to differentiate the plants is by identifying the leaves as they have unique features. If the correct choice is made, the specimen will be correctly identified. Our work with leaf classification will include binary leaf images and extracted features, including shape, margin & texture, as an integral part of the identification process to accurately identify 99 species of plants.

**Significance:**

Since a very long time the classification of plant species have been problematic and do not give accurate identification results. Accurate identification of a plant can be very helpful in knowing how it grows (e.g., size shape, texture, etc.) as well as how to care and protect it from pests and diseases.

**Exploratory Analysis / Data Cleaning:**

This dataset consists of 990 records in the training set and 594 records in the test set. It also contains 1584 images of the leaf specimens. There are 16 samples each of 99 species which have three sets of features i.e. shape, texture and margin per image. There is an Id assigned uniquely to each image.

The dataset provided contains no missing values, hence there was no need to process any missing values.

However, we tried two main analysis on the data: -

1. **Feature Selection: -** We tried to use backward selection process. In this process, the program tries to fit models first with all the input variables and continues to fit by decreasing the number of independent variables. In each step, it calculates the error and then finds the independent variables that gives the best results. It assigns a value of 1 to important variables and a large value to variables that do not give good results.

Code Snippet:

|  |
| --- |
|  |

1. **Feature Scaling: -** ML models work best when the input variables are normally distributed. StandardScaler makes the data as Gaussian with mean 0 and unit variance. It then normalizes the train labels using LabelEncoder which converts the categorical variable to 1s and 0s.

Code Snippet:

|  |
| --- |
|  |

**Type of Models:**

We have used various models on this dataset to judge which one gives a better result without overfitting. The following models have been used:

* **Random forest Classifier**- A random forest is an estimator that fits several decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

**Code Snippet:**

|  |
| --- |
|  |

* **Support vector machines (SVMs)-** The support vector machines are effective in high dimensional spaces. They use a subset of training points in the decision function called support vectors and hence it is also memory efficient.

**Code Snippet:**

|  |
| --- |
|  |

* **Multinomial Logistic Regression-** Multinomial Logistic Regression is the linear regression analysis to conduct when the dependent variable is nominal with more than two levels. It is used to describe data and to explain the relationship between one dependent nominal variable and one or more continuous-level (interval or ratio scale) independent variables.

**Code Snippets:**

|  |
| --- |
|  |

* **Neural Networks-** The basic neural network consists of 3 layers.

1. Input layer -The input layer consists of source nodes. This layer captures the features pattern for classification. The number of nodes in this layer depends upon the dimension of feature vector used at the input.
2. Hidden layer -The hidden layer lies between the input and output layer. The number of hidden layers can be one or more. Each hidden layer has a specific number of nodes (neurons) called as hidden nodes or hidden neurons. The hidden nodes can be varying to get the desired performance. These hidden neurons play a significant role in performing higher order computations. The output of this layer is supplied to the next layer.
3. Output layer -The output layer is the end layer of neural network. It results the output after features are passed through neural network. The set of outputs in output layer decides the overall response of the neural network for a supplied input features.

**Code Snippets:**

|  |
| --- |
|  |

**Literature:**

We have used several online resources to learn more on the several models that we have implemented on our project.

Below are the examples of the websites that we have used to get the model’s details with example.

1.Multinomial Logistic Regression:

<http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

2. Support vector machines (SVM):

<http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

3.Random forest Classifier:

<http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

4.Principal Component Analysis:

<http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

5.LabelEncoder:

<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

6.StandardScaler:

<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

7.GridSearchCv:

<http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>

8.Neural Network:

<http://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>

**Formulation / Libraries:**

We have followed the below steps for our project. These include all the steps of the formulation and step wise feature selections and finally the tuning of the model.

* We started with random forest classifier and did not use any feature selection method. We used all the available columns of data in the training set. This did not give us any solid result and we had log loss of 0.69615
* After this we used the feature scaling and normalized the independent variables. This was a great improvement on our existing score.
* After some research, we found that when number of independent variables are large logistic regression gives a better result than other classifiers. So, we used Multinomial Logistic Regression. This brought a great improvement on our score.
* Next thing we did is to fine tune the Logistic model. We used the GridSearchCV functionality present in sklearn. This also improved the score.
* To further improve the score, we tried to implement feature selection for which we tried Backward selection and PCA. However, these methods did not give a better result, so we decide not to use it in our project.
* After Tuesday’s Multivariate class we got idea to implement 3 things: - Using the image features, Neural Network and ensemble. We did not have time to implement Ensemble. However, we were successful in implementing images and Neural Network. This was a great improvement on our score. Our log-loss was 0.01705.

To formulate the data before using the classifiers we have used many techniques which have improved our models and in turn have helped us predict a better model. The following techniques were used:

* **StandardScaler-** The StandardScaler helps to standardize features by changing the mean to 0 and scaling to unit variance.
* **LabelEncoder-** LabelEncoder is used to normalize labels between 0 and 1 and transform non-numerical labels to numerical labels.
* **Principle Component Analysis-** principal components analysis reduces the number of variables to avoid multicollinearity. It reduces the dimensionality of the dataset.
* **Feature selection-** Feature selection is a process where we automatically select those features in our data that contribute most to the prediction variable or output. It is important as Having too many irrelevant features in your data can decrease the accuracy of the models.

We have done feature selection using the two following methods:

* + **Recursive Feature Elimination (RFE)-** It works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.
  + **Using Pipeline-** The purpose of the pipeline is to assemble several steps that can be cross-validated together while setting different parameters.
* **GridSearchCV-** The hyperparameters were tuned using GridSearchCV. It allows to construct a grid of all the combinations of parameters, tries each combination, and then reports back the best combination/model.

**Model Performance:**

Performance of each model is as given in the table below:

|  |  |  |
| --- | --- | --- |
| **Model Used** | **Log Loss** | **Kaggle Rank** |
| Support vector machines (SVMs) | 2.64136 | 648 |
| Random forest Classifier | 0.69619 | 540 |
| Multinomial Logistic Regression(GridSearchCv) | 0.03013 | 127 |
| Multinomial Logistic Regression (image processing) | 0.02402 | 82 |
| Neural Network (image processing) | 0.01705 | 40 |

**Our Rank:**

|  |
| --- |
|  |

**Limitations:**

Using Image Processing with Neural Network gave us the least log loss. The limitations of using Neural Network was that the code took nearly 9 hours to run.

We could have used tuning on the images such as Probabilistic Neural Network (PNN), Boundary Enhancement to reduce the runtime of the code.

Another limitation of our model was that using the feature selection did not give us any better score.

We could have used a better feature selection technique to reduce the log loss.

**Learning:**

Firstly, we would like to thank Professor Dr. Lawrence Fulton, for giving us an opportunity to work on this Leaf Classification Competition on Kaggle.

Working on the Leaf classification not only enhanced our knowledge exponentially on machine and deep learning but also gave us a chance to compete on a global platform.

We learned and used many machine learning techniques such as feature scaling, feature selection, PCA.

Secondly, we learnt how to tune the dataset given to us and extract the best information from it and use it to predict the best model using various classifiers from the Sklearn Package in Python.

While working on this project we also got some basic learning on Neural Networks, but due to shortage of time we could not get in depth knowledge about it. However Neural Network has enticed us so much that we aspire to continue learning more about it and try to implement many more techniques and see if we can predict an even better model.