LEAF CLASSIFICATION USING R

**INDEX:**

|  |  |
| --- | --- |
| **PROBLEM DISCUSSION** | Pg.1 |
| **SIGNIFICANCE** | Pg.2 |
| **EXPLORATORY ANALYSIS** | Pg.2 |
| **DISCUSSION ON MODELS** | Pg.4 |
| **LITERATURE** | Pg.5 |
| **FORMULATION** | Pg.5 |
| **MODEL PERFORMANCE** | Pg.6 |
| **LEARNING AND LIMITATIONS** | Pg.6 |

**1.Problem Discussion**: Classification helps us to identify species, study group specific attributes even preserve them. There are at least a million of plant species in world. Leaves, due to their volume, prevalence, and unique characteristics, are an effective means of differentiating plant species, luckily not many plants come with two different set of leaves. Manual classification of leaves is prolonged and has higher chance of error. The objective of this problem is to automate the process of leaf classification.

***Objective*:**

1. Develop a classifier to study the provided features and classify the data.
2. Extract new features from the images provided.
3. Examine the errors and tune the model.

**2.Significance**:  Automating plant recognition might have many applications, including

1. Plant based medicinal research
2. Crop and food supply management

The interesting aspects of the problem involve extraction of data from the images and attempts to reduce the dimensions of the data. The applications stated above can be broadened to describe soil type, estimate climatic conditions etc.

**3.Exploratory Analysis / Data Cleaning**.

There are 991 rows and 194 columns in the training set. And there are 595 rows and 193 columns in the test set.There are **no missing values** in both the training and the test sets.Each leaf id is explained by 64 margin properties,64 texture properties and 64 shape properties.

One thing to notice in the training set is that margin and the texture properties are left skewed but the shape properties are close to normal distribution.Verified the normailty of the fields using hist().There are no outliers in the data both in the training and test set.

***Data:***

Features are extracted from binary leaf images

* Margin: Fine-scale margin histogram
* Texture: Interior texture histogram
* Shape: Shape contiguous descriptor

***Data fields***

* id - an anonymous id unique to an image
* margin\_1, margin\_2, margin\_3, ..., margin\_64 - each of the 64 attribute vectors for the margin feature
* shape\_1, shape\_2, shape\_3, ..., shape\_64 - each of the 64 attribute vectors for the shape feature
* texture\_1, texture\_2, texture\_3, ..., texture\_64 - each of the 64 attribute vectors for the texture feature

***Missing Values:***

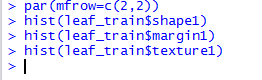
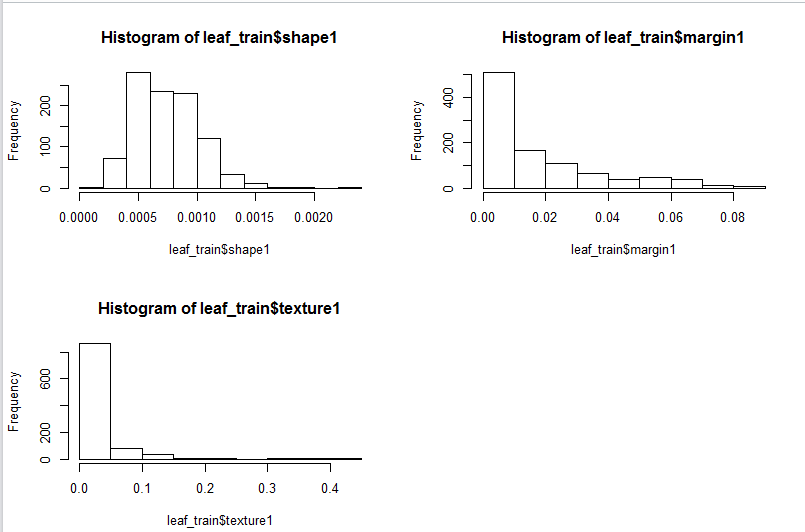


No Missing Values in the Training data



No Missing Values in the Test data

***Distribution of Data****:*

**4.Discussion on Models**: We pondered over using the following model and we have justified why.

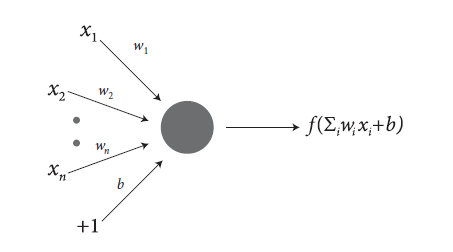
1. Decision tree
2. Random forest using principal component analysis.
3. Neural networks
4. Deep learning in h2o

***Decision tree***:

Decision tree is a tree type model which starts data division into pure subsets, like keeps subsetting data till it has pure response variable output.

In our case we have 192 independent variables even without data extracted from the images. This caused our the problem of overfitting in our model leading to poor decision results in the model.

***Neural networks and Deep learning:***

The human brain can process image or speech data with ease . The element behind the superior functionality of human brain is the neuron. A neuron communicates with other neurons through synaptic links. Billions of neurons communicate with each other and form a powerful and complex network. The concept of perceptron is derived from this :

The above diagram shows a perceptron receiving inputs multiplied by weights and are summed along and passed through an activation function along with bias. Complex networks

Involving multiple layers and many of these perceptron constitute a neural network.

Deep learning is a type of algorithm which emplyees multi layered architecture.

The Deep learning in h2o also uses feed forward neural networks with back propogation.

**Our Conclusion:**

1. Neural networks are best when it comes to recognition of a pattern.
2. Complex patterns can be trained into the networks with the presence of hidden layers.
3. Presence of excellent documentation being relatively newer package in field propelled us to implement these techniques.

***Principal Component Analysis(PCA):***

PCA is a very commonly used method for data reduction and since we have 192 variables it looked to be a good idea to do PCA and reduce the independent varaibles from 192 to somewhere around 15 to 20.But it proved to be of no use as it did not help in increasing the accuracy of the prediciton.And since we had 3 groups of independent variables shape,texture and margin , we needed to perform PCA separately on the group and reduce them and combined their respective components,. Which did not work out either in terms of increasing the accuracy.

**5.Literature**:

***H2o Deep Learning:***H2o deep learning has been used in the implementation of MNSIT data digit recognizer. The journal used for reference was h2o world’s tutorial.

[https://www.gitbook.com/book/h2o/h2o-world-2015-training/details](https://www.gitbook.com/book/h2o/h2o-world-2015-training/details%20) .

***Principal Component Analysis:***We thought of using principal component analysis and the following video was our reference

<https://www.youtube.com/results?search_query=principal+component+analysis+in+r>

***Factor Analysis:*** We thought of using principal component analysis and the following video was our reference

<https://www.youtube.com/results?search_query=principal+component+analysis+in+r>

***Image Reading:***We used Jpeg library to read images in R.

<https://cran.r-project.org/web/packages/jpeg/jpeg.pdf>

**6.Formulation / Libraries**:

***Model: Deep learning***

The model is formulated by understanding the basic structure of neural networks as mentioned above. A neural network has few main attributes to be considered.

1. Weights: They typically range between values of -1,0.5,0,1.
2. Activation function: The output is produced when combined input from the previous layer passes certain threshold.
3. Number of inputs: As the possibility of dimensional reduction from conventional methods is Not applicable to our data set we have to present all the inputs to the network.
4. Hidden layer: Selection of number of hidden layers to a given network doesn’t include any thumb rules. The number of neurons in the hidden layer should be between the number of inputs and out puts. Two hidden layers are well known to understand most of the complex relations. Taking too many neurons in the hidden layer causes the network to learn the training set and effect generalization.

H2o provides with many arguments which are mostly recommended to be in default.

1 .Standaridize: It is set to true to scale our data so that we can achieve convergence quicker.

2. Epochs: The number of iterations the training set is iterated to under go training and tune the weight parameters.

3. Hidden: It is used to specify the number of hidden layers and the neurons to be present in our neuron.

**6.Model Performance**:

The performance of the model with h2o deep learning was highest of all the models in comparison to time required for computation, lines of code and performance. The best rank we achieved so far was 400.

The performance of the model with PCA followed by RandomForest was convincing though not great.The computation time was high as the number of trees were increasing.Maximum rank we could get with this wa around 623.

The performance of the model with PCA followed by Decision tree was not convincing.The computation time was less compared to random forest.But the prediction accuracy was below expectations.

**7.Limitations**:

The model performance can be considerably increased by incorporating the image data. Main limitation of neural networks is once the model is trained it gives output to even an entirely new data with similar input parameters as it depends on weights.

The limitations of this model is that ,even though it has many dimensions,performing PCA or Factor Analysis did not give favourable results therefore having to deal with 192 dimensions is a limitation as it reduces the computation power

**8.Learning**:

*Facts are learned through mistakes* is the sentence that surfaces our mind after attempting the project.

The initial study began with the implementation of Decision tree in the model and learning its limitations and its performance. In order to improve the model performance and reduce the problem of over fitting factor analysis and principal component analysis has been attempted. This analysis couldn’t improve our Decision tree as these methods reduce the dimensions of the data without providing any information about the important variables instead reduce the scale of data by adding them which proved to be fatally in effective in this case due to nature of the data.

The concept of neural nets has been understood due to review of literature in order to properly tune the parameters.

The packages neural net and nnet also have been used to implement neural nets but they weren’t of much help develop a classifer.H2o proved challenging and useful due to its vast amount of arguments to use and understand about.

The first thought that came when we looked at 192 independent variables is to find a way to reduce their count using certain techniques(PCA/FA). During this study we learnt the concept behind PCA , Kaiser rule etc.PCA actually provides n componenet for n variables and kaiser rule states that it is enough to consider components whose variance is greater than .5.Following the kaiser will significantly reduce the number of components.

Random looked to be perfect model initially as it works as a combination of decision trees.The accuracy of our results increased as we used more trees in our random forest model using ntree argument of the RandomForest().