**Name: Aditi Helekar** **UNCC ID: 800966667**

Course: Machine Learning(ITCS_6156)

# Decision Tree Classification Algorithm Report

**Name: Aditi Helekar** **UNCC ID: 800966667**

## Problem Statement:

Design a Decision Tress classification algorithm for the given datasets.

## Dataset exploration questions:

1. What is the number of attributes in each dataset?

Dataset 1: Optical digit recognition

This contains 64 input attributes and 1 class attribute.

Dataset 2: Amazon reviews

This contains 3 attributes: name, review, rating.

2. What is the number of observations?

Dataset 1: Optical digit recognition

This contains 3823 observations for training and 1796 for testing.

Dataset 2: Amazon reviews

This contains 146824 observations for training and 36707 for testing.

3. What is the mean and standard deviation of each attribute?

Dataset 1: Optical digit recognition

These were included in Mean_Standard_deviation_calculation.xlsx.

Dataset 2: Cannot be calculated since it is text data.

4. What is the distribution of different classes in each of the datasets?

Dataset 1: Optical digit recognition

The class distribution for training set of Optical digit recognition.

Class: No of examples in training set

0: 376

1: 389

2: 380

3: 389

4: 387

5: 376

6: 377

7: 387

8: 380

9: 382

The class distribution for testing set of Optical digit recognition.

Class: No of examples in testing set

0: 178

1: 182

2: 177

3: 183

4: 181

5: 182

6: 181

7: 179

8: 174

9: 180

Dataset 2: Amazon reviews

The class distribution for training set of Amazon reviews

Class: No of examples in training set

1: 12146

2: 9040

3: 13364

4: 26509

5: 85765

The class distribution for testing set of Amazon reviews

Class: No of examples in testing set

1: 3037

2: 2270

3: 3415

4: 6696

5: 21289

**Approach:**

I have implemented decision tree algorithm in Python using scikit-learn library. Scikit-learn is library designed for implementing Machine learning techniques in Python.

The basic approach is described below:

1. Read the csv file using pandas.io library and store it as a Dataframe.
2. Classify the attributes as samples and features to apply it to Decision Tree Model. Features are also referred as predictors and samples as responses.
3. Create an object of DecisionTreeClassifier imported from scikit learn library.
4. Using fit function, fit the dataframe to tree model. Fit function finds patterns in the given dataset.
5. For cross-validation, divide the training dataset in train and test data using train_test_split function.
6. Predict the behavior of model using predict function with test data.
7. After cross validation use the test dataset and predict again.
8. Calculate accuracy using accuracy_score function.
9. Generate image of tree using pydotplus and grapgviz library.
   The figure below has the snippet of code:

```
new_classifier = tree.DecisionTreeClassifier()
new_classifier = new_classifier.fit(X_train, Y_train)
print("************************************************************")
new_predictions = new_classifier.predict(X_test)
#print(new_predictions)
#print(Y_test)

sklearn.metrics.confusion_matrix(Y_test,new_predictions)
print("Testing Accuracy:")
print(sklearn.metrics.accuracy_score(Y_test, new_predictions))
```

**Name: Aditi Helekar** **UNCC ID: 800966667**

**Observations:**

1. For Digital Recognition dataset:

```
Reading dataset.
(3822, 65)
Training Accuracy:
0.899934597776
**************************
Testing Accuracy:
0.857461024499
Creating graph image:
Execution completed.
```

Accuracy of Training dataset = 0.899 = 89.99%

Accuracy of Testing dataset = 0.857 = 85.7%

I have attached the image file of Decision tree named “OPTTree.png”

2. For Amazon dataset

The figure below shows the output of program displaying the accuracy.

```
Reading dataset:
(19896,)
(19896,)
After fit transform
Training Accuracy:
0.835783389873
------------------------
Testing Accuracy:
0.835877862595
Execution completed.
```

   a. Accuracy of Training dataset = 0.8357 = 83.57%
   b. Accuracy of Testing dataset = 0.8358 = 83.58%

**Function Description:**

1. fit(X,y): Builds a decision tree classifier from the training set(X,y)
2. predit(X,check_input=True): Predit class or regression value for X
3. train_test_split(x,y,): Split arrays or matrices into random train and test subsets
4. confusion_matrix(x,y): Used to describe performance of classification model on a set of test data for which the values are known
5. accuracy_score(x,y): Computes subset accuracy
6. CountVectorizer(): Converts a collection of text document to a matrix of token counts
7. transform(): Transforms document to document-term matrix
8. fit_transform(): Learns the vocabulary dictionary and returns term document matrix

**Name: Aditi Helekar** **UNCC ID: 800966667**

**Conclusion:**

The accuracy of the model increases if we increase the training dataset as the model will have large data to learn from. So for Amazon dataset at first we divided the training dataset the accuracy was 83% and after considering the total training and testing dataset the accuracy remains constant. But whereas for Digital recognition dataset the accuracy decreases.

**References:**

1. http://scikit-learn.org/stable/modules/classes.html
2. https://github.com/justmarkham/pycon-2016-tutorial/blob/master/tutorial.py
3. https://www.youtube.com/watch?v=ZiKMIuYidY0&t=5526s
4. https://www.coursera.org/learn/machine-learning-data-analysis/lecture/yHOYj/building-a-decision-tree-with-python
5. http://pandas.pydata.org/pandas-docs/stable/
6. http://www.graphviz.org/Documentation.php