Project Report:

By:

ANUPKUMAR N J
(101880602)

AIMEE CIANE NYAMBO
(101880199)

DATE: 02/27/2020

Contents:

1.Description about the code:

Code requires following libraries to be imported:

a. Numpy : Numerical python is used for high-level mathematical calculation
b. Pandas : Dataframes are used for formatting datasets according to the need
c. Copy : For shallow and deep copying operations. (for trees)
d. Pprint : To print data in pretty way (for trees)
e. Collections : This is used to store collections of dictionaries

The whole project can be divided into modules :

1. DataFormatting : The functions formatData() and foramtDataFT() helps to convert from csv file to the pandas dataframe. The frameworks helps easy accessing and manipulating Data for the functions. formatData() returns dictionary with keyword ID,attributes(1 to 60) and class which is used for training. The foramtDataFT() returns without class keyword which is used for testing purposes. Due to conflict of structure, different functions are used.

2. Entropy method : The functions findEntropyWinner(), findEntropyAttribute(), findEntropy() are used to calculate IG with entropy.

   findEntropyWinner() : This function takes dataframe as input and returns node with highest information gain. It generates attributes from 1 to 60 and sends to the rest of functions and returns information gain of all attributes. Finally this function, returns node with maximum IG.

   findEntropy(): This function takes df (dataframe) as input and returns entropy of the whole. It extracts different classes and their respected count is found . Finally the total entropy is returned using the entropy formula (fraction * log2(fraction))

   findEntropyAttribute(): This function takes df and attribute(1-60) and returns entropy of individual attribute. This function calculates number of occurrence of that attribute in different class and also total occurrence for all the values (A,G,T,C). Finally, it returns entropy of individual by using the entropy formula (fraction * log2(fraction)).

3. Gini index method: The functions find_gini_winner(), findGiniIndexOfAttribute(), findGini() helps to calculate gini index of attributes and return node of minimum value.

Find_gini_winner(): This function recurses through all keys and passes into functions and holds array of gini index value of attributes. This finally returns minimum value out of array.The node with lower value is used first to build tree.

findGiniIndexOfAtribute(): This function accepts df and attribute and returns gini index of that attribute. It recurses through attribute values (A G T C) and find number of occurrence of each values which is divided by total. Then passes control to find gini to find gini index of each values.

Find gini() : This calculates number of occurrences in each target classes(num). This helps to calculate gini index for individual attribute value (A G T C) by using formula 1 – summation (num/den) ** 2

4. Chi square modules: The functions getAppropriateNode(), chiSquareTestPruning(), chiSquare() and findExpectedNode() help selecting the node which satisfies criteria.

   getAppropriateNode(): This accepts dataframe and returns the node which fulfills the criteria. This function finds node with higher IG. And returns the node if it satisfies criteria else checks for another node. This depends on value from another function

   chiSquareTestPruning() : This accepts node and confidentiality value and returns 0 if node is not selected , else 1. This checks chi square value of node with hard coded values which in turn depend on confidentiality level.This function gets chi square value of a node from function chiSquare

   chiSquare(): This accepts node and return its chisquare value. This function recurses through attributes (A G T C) classes (N, IE, EI) and find chisquare value by using formula (actual – expected) **2/ expected

   findExpectedNode() : This accepts node and returns its expected number. This function recurses through A G T C and N IE EI forming a table which gives occurences of the combinations. This function has the flag saveFlag, when set the expected number is saved in global variable for future calculations.When unset, just returns expected number.

5. Tree related: The functions buildTree() and subtable() help to build the tree.

   buildTree(): This function helps building tree using nested dictionaries.Here, we can set different flags which help to choose the different methods.

   Subtable(): This function maintains table which helps to recurse through tree.

6. Predict and accuracy : predict() and findAccuracy() helps to predict by building tree and find accuracy respectively.

7. Miscalculation error : The functions Find_misEr_winner(), findmisErIndexOfAttribute(),

   FindMisEr() help to calculate the node with minimum value.

   Find_misEr_winner(): This function recurses through all keys and passes into functions and holds array of max probabilities value of attributes. This finally returns minimum value out of array.The node with lower value is used first to build tree.

   findmisErIndexOfAttribute (): This function accepts df and attribute and returns probabilities of that attribute. It recurses through attribute values (A G T C) and find number of the occurrence of each values which is divided by total. Then passes control to find misEr to find max probabilities of each values.

   findMisEr(): This calculates number of occurrences in each target classes(num). This helps to calculate max probability for individual attribute value (A G T C) by using formula 1 – max of probabilities

2.Accuracies of different methods:

The methods used are:

1. Entropy and IG: The accuracy observed is around 80%.
2. Gini index and IG: The accuracy is observed around 62 %
3. Miscalculation error: The accuracy observed around 60%
4. Chi square pruning: Not achieved

3.Method with highest accuracy:

We would go with entropy because accuracy is higher than other. The tree building using entropy is faster compared to other. And also, it gives clear picture of the data splitting. This option works well because it considers entropy which helps to measure purity of the data more efficiently. Entropy's maximum impurity is 1 and maximum purity is 0. So we have intuition that higher the margin, better data split which in turn gives better accuracy.

Method with relatively lower accuracy:

Gini index, miscalculation error and chisquare testshowed lower accuracy because there was problem while building tree. This error showed because of more number of nodes compared to entropy. The program control flow was lost in the recursion. We have also implemented tree using nested dictionary which is not the best way to do it. But we tried writing better recursive function for the same. But ended up stack error.