Output of the results for the training data given :

Training Data set count	TestDataSet count	Threshold Value	Percentage Accuracy	Time Taken(secs)
40,000	25,000	1	73.488	196
40,000	25,000	0.1	72.524	150
40,000	25,000	0.05	70.464	145
40,000	50,000	1	73.488	220
40,000	50,000	0.1	72.524	170
40,000	50,000	0.05	70.464	172

In the second phase we double the test data by duplicating the given data the results are same.

Following are the steps that we used to solve the given problem :

1. The program parses the feature names file, training data set, training values sets and constructs a matrix of Data.
2. Then it parses the training data set files and creates a matrix of test data.
3. We are then calculating the entropy of the values using the following formula :

value = -probPos*math.log(probPos,2) - probNeg*math.log(probNeg,2)

Where probPos is the probability of number of 1 in the values column and probing is the probability of number of 0 in values column.

4. We then split the attribute values for each feature value by using the average of the all attribute values present. In that way we divide the attributes into two ways by comparing the values to the average vale and confirm them as left ones and the values above them as right ones.

5. We are calculating the Information gain for each and every feature using the following steps :

i. calculate the left value entropy :
leftvalue = -(float(leftCountPositive)/leftCount)*math.log(float(leftCountPositive) / leftCount,2) - (float(leftCount-leftCountPositive) / leftCount)*math.log(float(leftCount - leftCountPositive) / leftCount,2)

where leftCountPositive is the value for which left values have 1 in the values column
leftCount is the total left values in the feature

ii. calculate the right value entropy :
rightvalue = -(float(rightCountPositive)/rightCount)*math.log(float(rightCountPositive) / rightCount,2) - (float(rightCount - rightCountPositive) / rightCount)*math.log(float(rightCount - rightCountPositive) / rightCount,2)

where rightCountPositive is the value for which right values have 1 in the values column
rightCount is the total left values in the feature

iii. Calculate the feature entropy :
featureEntropy = (float(leftCount)/total)*leftvalue + (float(rightCount)/total)*rightvalue

iv . Information gain for the feature :

subtract the feature entropy from the values entropy.

informationGain = cal_entropy_valuesColumn() - featureEntropy

6. Creating a node with information gain, split value, leftInfvalue, rightInfValue

7. Once we get all the feature entropy's we are saving the node in to the max heap where the node contains maximum information gain will be the root node.

8. In this way we insert the nodes in to the max heap tree and stop until we reach a pure data set.

9. Now after the tree is constructed from the test data matrix that we created initially we select the depending upon the chi-square criteria and select the node's split value and compare against the clicks value that we get from the test set.

10. If the value is less than the split, we consider it as left node and compares whether the left Information gain is greater than the right information gain, if so then the result we are counting it as success.

11. Similarly, If the value is greater than the split, we consider it as right node and compares whether the right Information gain is greater than the left information gain, if so then the result we are counting it as success.

12. At the end we are finding the percentage of the success by using the number of success count divided by the total counts.