MACHINE LEARNING PROJECT REPORT

-

PREDICTION OF SATISFACTION OF CUSTOMERS IN AIR TRAVEL OFFERED BY THE AIRLINE

INTRODUCTION: This machine learning project aims to predict customer's satisfaction in air travel based on the different services and amenities offered by the airline in which they are travelling. The dataset is a public dataset taken from Kaggle as a Survey by United Airlines. It has attributes like Ease of Boarding, Departure and Arrival Time,Departure and Arrival Delay, Ease Of Checking in, Seat Comfort, In Flight Entertainment system, the class of Travel,Purpose of travel,etc . The numeric attributes are values in between 0-5 where 5 means a higher rating for the attribute. The target attribute is a binary attribute as 'satisfied or dissatisifed'. Only the numeric attributes are considered for the training and testing purposes.

ALGORITHMS USED :

1) a) DECISION TREE: The main classification problem is run using a decision tree which is built using both Gini Index and Information Gain. The depth of the tree is varied to give better accuracy. The accuracy of the classification along with precision measures like recall and F-score are calculated and printed. The algorithm is implemented in python using the Scikit-Learn package. The tree for both cases of attribute selection measures are plotted using graphViz function into a pdf file. The dataset is split into training set and a test set using a 70% handout method.
The model is run using various depths of tree for information gain and corresponding accuracy values are noted. A graph of Accuracy vs Depth is plotted in Gnuplot using this data.

b) DECISION TREE WITH BAGGING: The same binary classification problem is run again,this time with bagging the dataset and 10 fold cross-validation. For each of sub – datasets, a decision tree is constructed . The accuracy values of the 10 folds are printed and the overall accuracy is the mean of all the folds. Bagging Classifier function is used from Scikit-Learn package. It is a meta-algorithm, which takes M subsamples (with replacement) from the initial dataset and trains the predictive model on those subsamples.

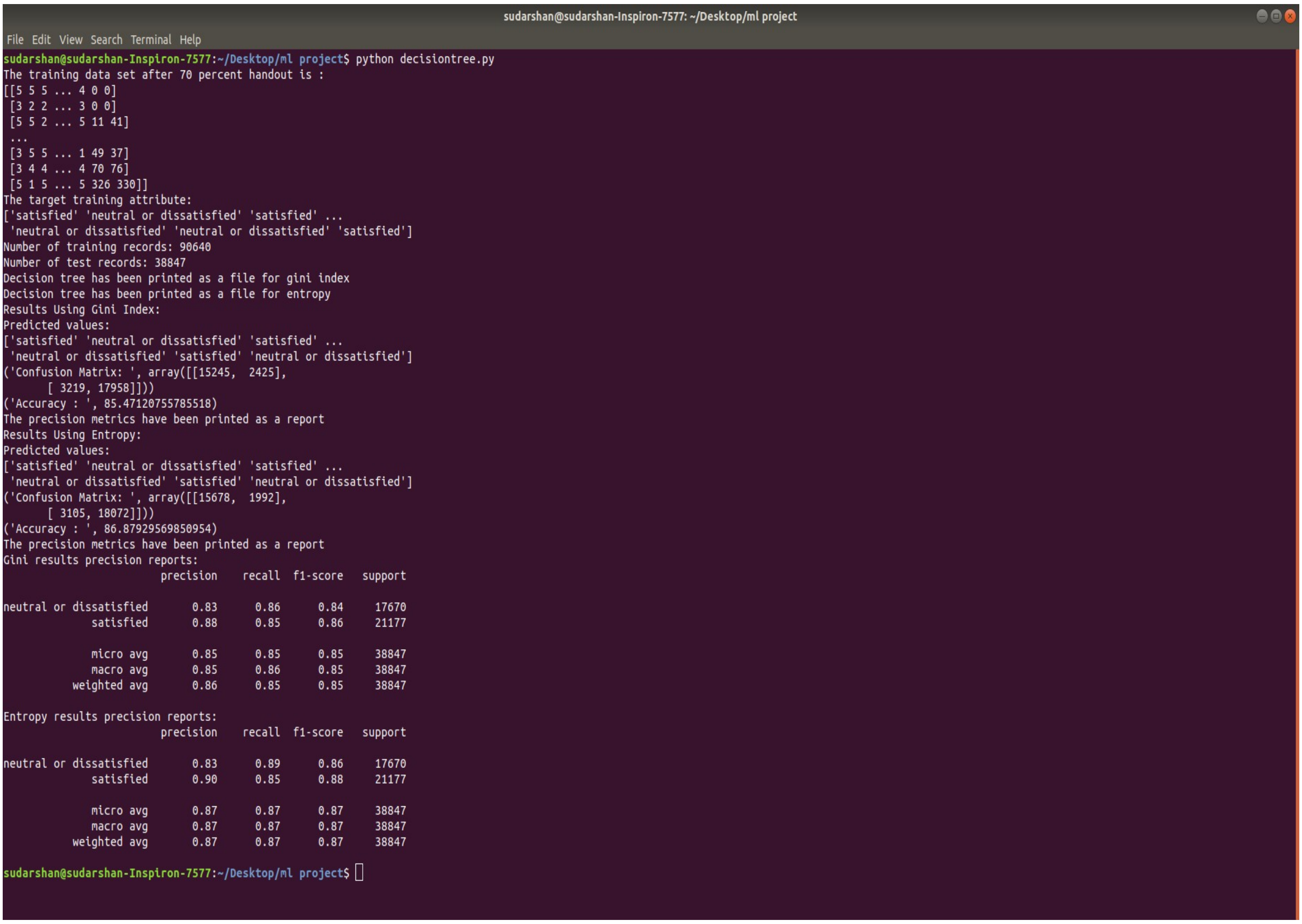
The final model is obtained by averaging the "bootstrapped" models and usually yields better results.

2) GAUSSIAN NAIVE BAYES CLASSIFIER: To compare the accuracy of various models ,the same problem is this time run using bagging and 10 fold cross-validation and Gaussian naive bayes Classifier is used for each of the sub datasets. The accuracy values for each of the folds is printed and the overall accuracy is the mean of all accuracy values.

3) SUPPORT VECTOR MACHINES: The same problem is run using support vector machines to predict the accuracy and compare it with other models. The model’s training time is significantly longer than other models and hence around 1,00,000 records is used for the training.

IMPLEMENTATION OUTPUTS

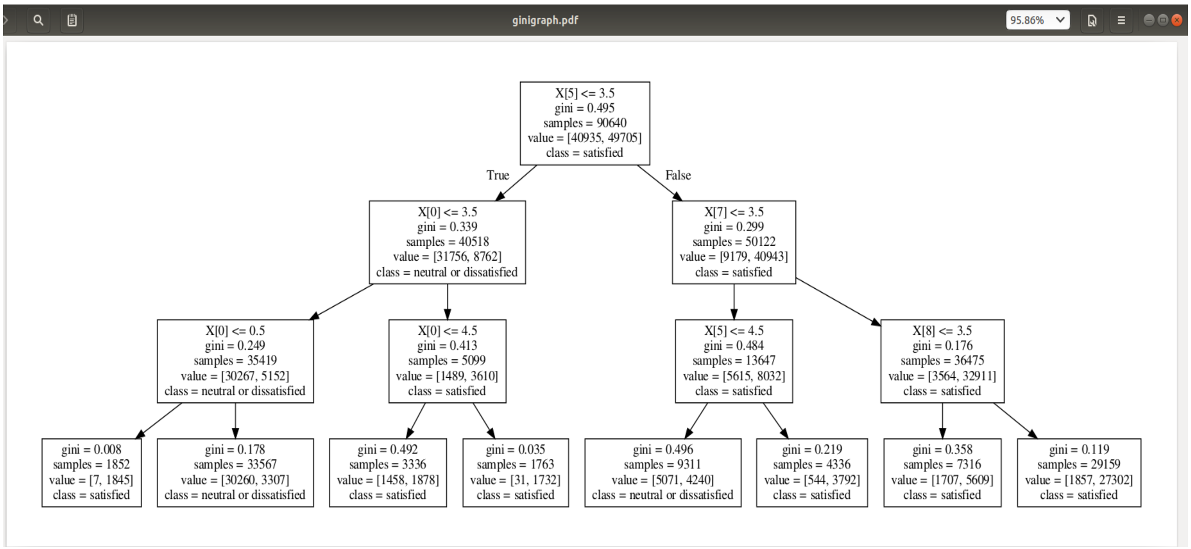
1) Running decisiontree.py, we get the results of the decision tree run on the dataset as:

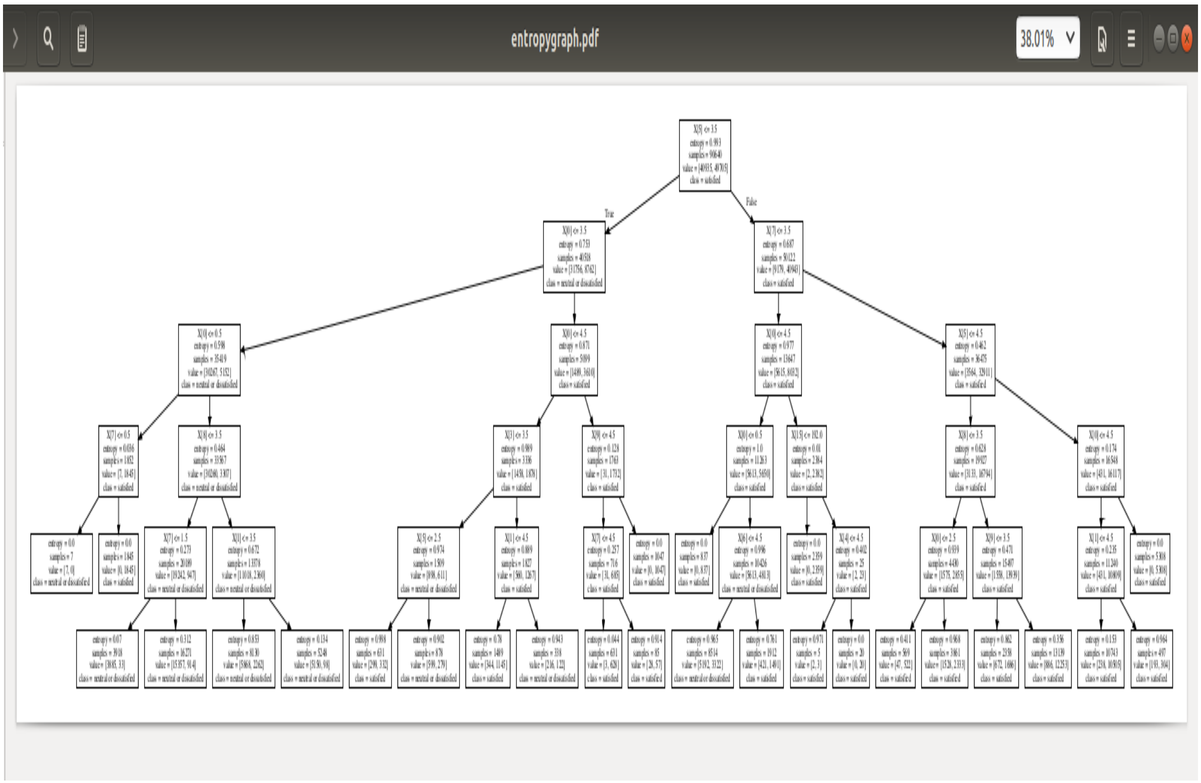

sudarshan@sudarshan-Inspiron-7577: ~/Desktop/ml project
File Edit View Search Terminal Help

```
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ python decisiontree.py
The training data set after 70 percent handout is :
[[5 5 5 ... 4 0 0]
 [3 2 2 ... 3 0 0]
 [5 5 2 ... 5 11 41]
 ...
 [3 5 5 ... 1 49 37]
 [3 4 4 ... 4 70 76]
 [5 1 5 ... 5 326 330]]
The target training attribute:
['satisfied' 'neutral or dissatisfied' 'satisfied' ...
 'neutral or dissatisfied' 'neutral or dissatisfied' 'satisfied']
Number of training records: 90640
Number of test records: 38847
Decision tree has been printed as a file for gini index
Decision tree has been printed as a file for entropy
Results Using Gini Index:
Predicted values:
['satisfied' 'neutral or dissatisfied' 'satisfied' ...
 'neutral or dissatisfied' 'satisfied' 'neutral or dissatisfied']
('Confusion Matrix: ', array([[15245,  2425],
       [ 3219, 17958]]))
('Accuracy : ', 85.47120755785518)
The precision metrics have been printed as a report
Results Using Entropy:
Predicted values:
['satisfied' 'neutral or dissatisfied' 'satisfied' ...
 'neutral or dissatisfied' 'satisfied' 'neutral or dissatisfied']
('Confusion Matrix: ', array([[15678,  1992],
       [ 3105, 18072]]))
('Accuracy : ', 86.87929569850954)
The precision metrics have been printed as a report
Gini results precision reports:
                         precision    recall  f1-score   support

neutral or dissatisfied       0.83      0.86      0.84     17670
              satisfied       0.88      0.85      0.86     21177

              micro avg       0.85      0.85      0.85     38847
              macro avg       0.85      0.86      0.85     38847
           weighted avg       0.86      0.85      0.85     38847

Entropy results precision reports:
                         precision    recall  f1-score   support

neutral or dissatisfied       0.83      0.89      0.86     17670
              satisfied       0.90      0.85      0.88     21177

              micro avg       0.87      0.87      0.87     38847
              macro avg       0.87      0.87      0.87     38847
           weighted avg       0.87      0.87      0.87     38847

sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ 
```

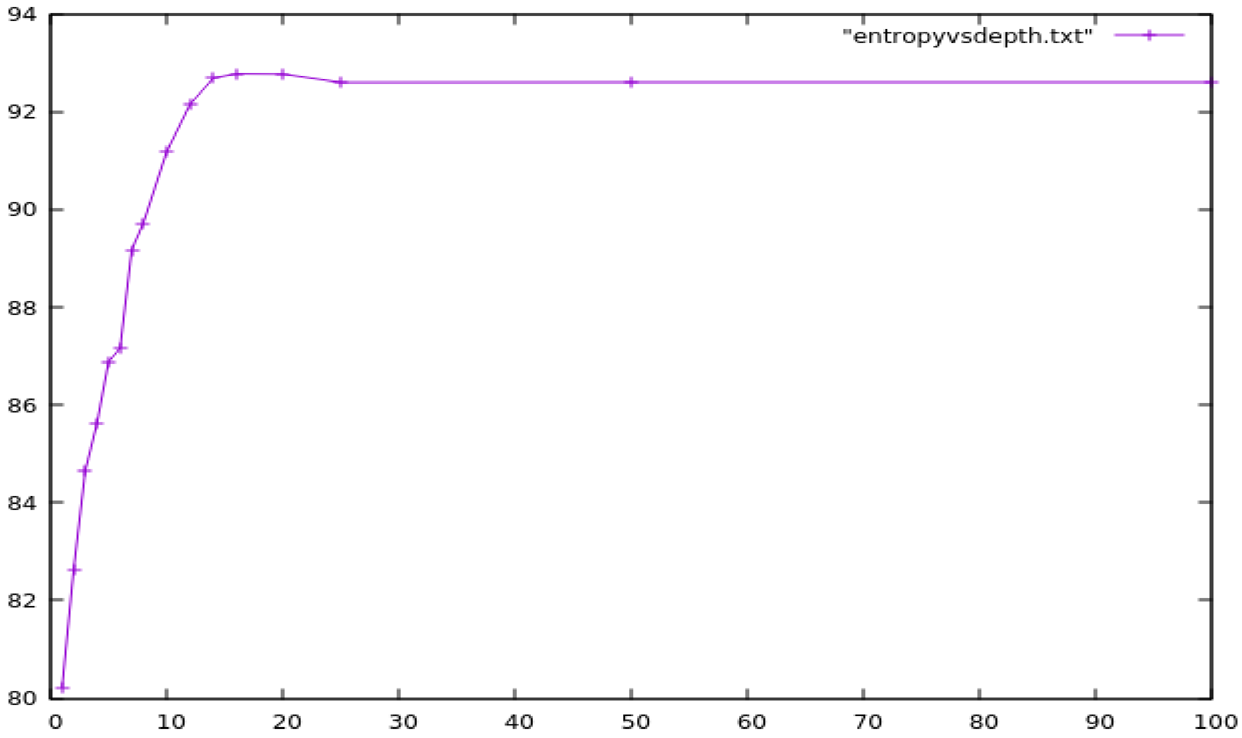
The model prints the accuracy for both cases and an accuracy of about 85% is obtained for gini and 86% for information gain with depth of 5.

This creates a pdf file for each of the 2 decision trees built using information gain and gini index as:

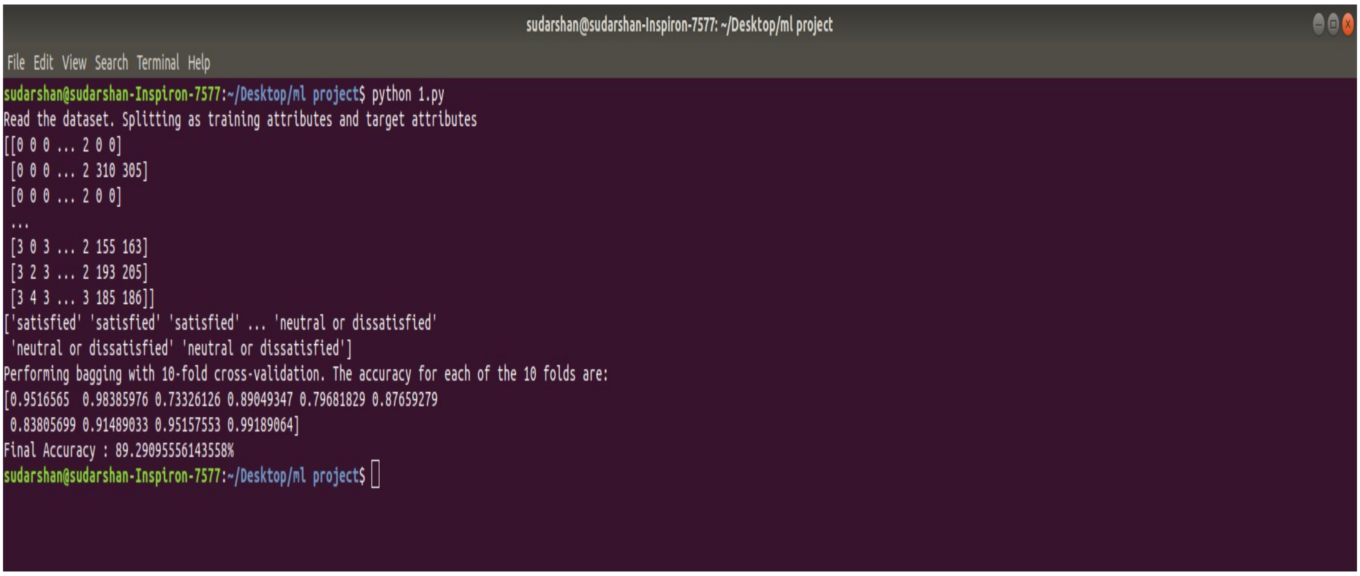




Running the same model different times and plotting a graph of Accuracy vs depth of the tree using Gnuplot we get :



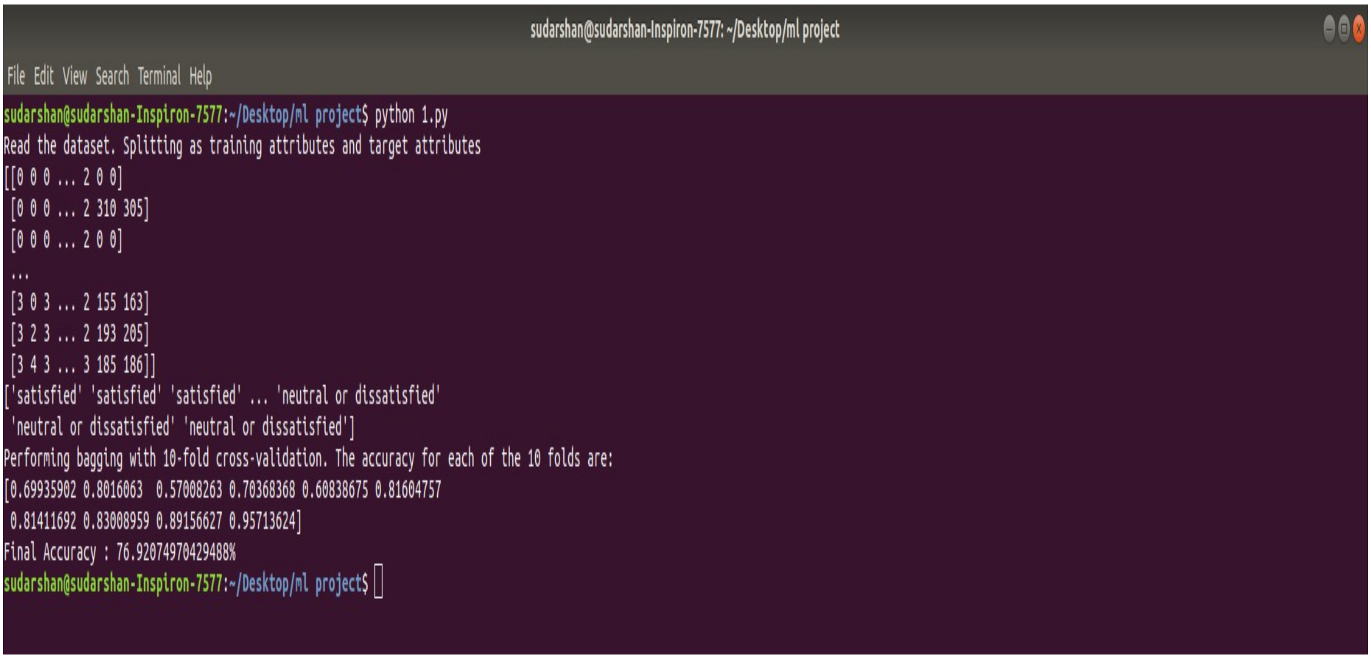
2) Running Decision Tree with Bagging as
python 1.py we get:


sudarshan@sudarshan-Inspiron-7577: ~/Desktop/ml project
File Edit View Search Terminal Help

```
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ python 1.py
Read the dataset. Splitting as training attributes and target attributes
[[0 0 0 ... 2 0 0]
 [0 0 0 ... 2 310 305]
 [0 0 0 ... 2 0 0]
 ...
 [3 0 3 ... 2 155 163]
 [3 2 3 ... 2 193 205]
 [3 4 3 ... 3 185 186]]
['satisfied' 'satisfied' 'satisfied' ... 'neutral or dissatisfied'
 'neutral or dissatisfied' 'neutral or dissatisfied']
Performing bagging with 10-fold cross-validation. The accuracy for each of the 10 folds are:
[0.9516565  0.98385976 0.73326126 0.89049347 0.79681829 0.87659279
 0.83805699 0.91489033 0.95157553 0.99189064]
Final Accuracy : 89.29095556143558%
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ 
```

The final accuracy of the model is printed on the terminal and gives a rough accuracy of around 89% with 10 fold cross validation.

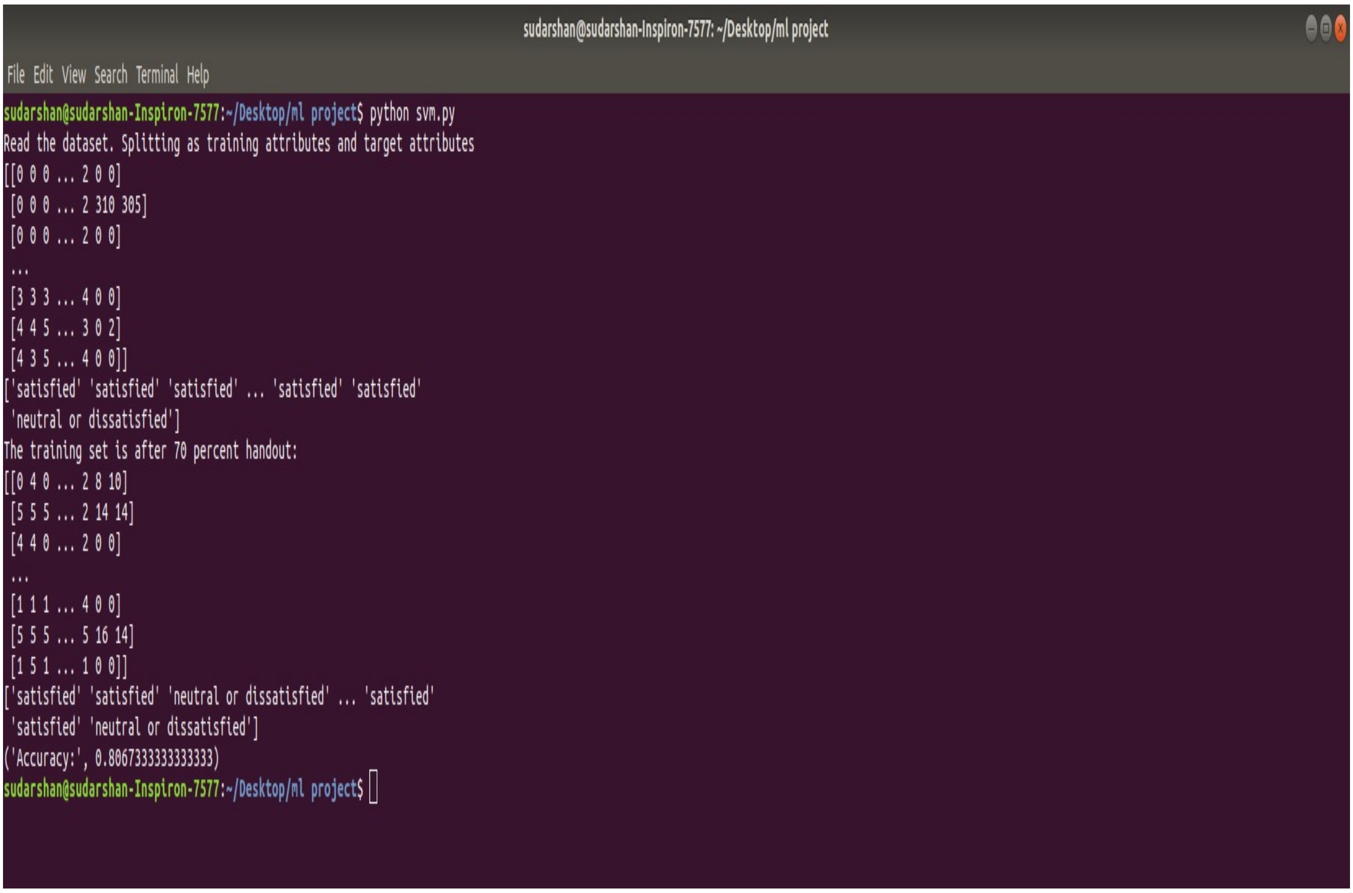
3) Running Gaussian Naive Bayes classifier with bagging by changing the file 1.py according to the compile instructions and running the model again:


sudarshan@sudarshan-Inspiron-7577: ~/Desktop/ml project
File Edit View Search Terminal Help

```
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ python 1.py
Read the dataset. Splitting as training attributes and target attributes
[[0 0 0 ... 2 0 0]
 [0 0 0 ... 2 310 305]
 [0 0 0 ... 2 0 0]
 ...
 [3 0 3 ... 2 155 163]
 [3 2 3 ... 2 193 205]
 [3 4 3 ... 3 185 186]]
['satisfied' 'satisfied' 'satisfied' ... 'neutral or dissatisfied'
 'neutral or dissatisfied' 'neutral or dissatisfied']
Performing bagging with 10-fold cross-validation. The accuracy for each of the 10 folds are:
[0.69935902 0.8016063  0.57008263 0.70368368 0.60838675 0.81604757
 0.81411692 0.83008959 0.89156627 0.95713624]
Final Accuracy : 76.92074970429488%
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ 
```

the overall accuracy of the 10 folds is printed in the terminal. This gives an overall accuracy of about 76%

4) Running the model support vector machines using:
python svm.py


sudarshan@sudarshan-Inspiron-7577: ~/Desktop/ml project
File Edit View Search Terminal Help

```
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ python svm.py
Read the dataset. Splitting as training attributes and target attributes
[[0 0 0 ... 2 0 0]
 [0 0 0 ... 2 310 305]
 [0 0 0 ... 2 0 0]
 ...
 [3 3 3 ... 4 0 0]
 [4 4 5 ... 3 0 2]
 [4 3 5 ... 4 0 0]]
['satisfied' 'satisfied' 'satisfied' ... 'satisfied' 'satisfied'
 'neutral or dissatisfied']
The training set is after 70 percent handout:
[[0 4 0 ... 2 8 10]
 [5 5 5 ... 2 14 14]
 [4 4 0 ... 2 0 0]
 ...
 [1 1 1 ... 4 0 0]
 [5 5 5 ... 5 16 14]
 [1 5 1 ... 1 0 0]]
['satisfied' 'satisfied' 'neutral or dissatisfied' ... 'satisfied'
 'satisfied' 'neutral or dissatisfied']
('Accuracy:', 0.8067333333333333)
sudarshan@sudarshan-Inspiron-7577:~/Desktop/ml project$ 
```

RESULT:

The different algorithms are run on the dataset and their accuracy measures are obtained. From this we can see that Decision tree and Support Vector Machines give good accuracy of about 85-90% while Gaussian Naive Bayes Classifier gives an accuracy of about 76%. The decision tree is also plotted and the most important attributes can be analysed from the graph. Airlines would need to focus on these attributes to better their overall product so that more people are satisfied with their product.