PREDICTING SATISFIED/ DISSATISFIED CUSTOMER

BY:SEERAT CHHABRA

SANTANDER BANK

Overview

- Customer satisfaction is a measure of success for the bank
- Dissatisfied customers won’t continue with the bank
- Predicting dissatisfied customers early in the relationship can help take corrective steps to improve satisfaction level of customers and retain them

Costlier to get new customers than to retain them!

Data Summary

Training Data

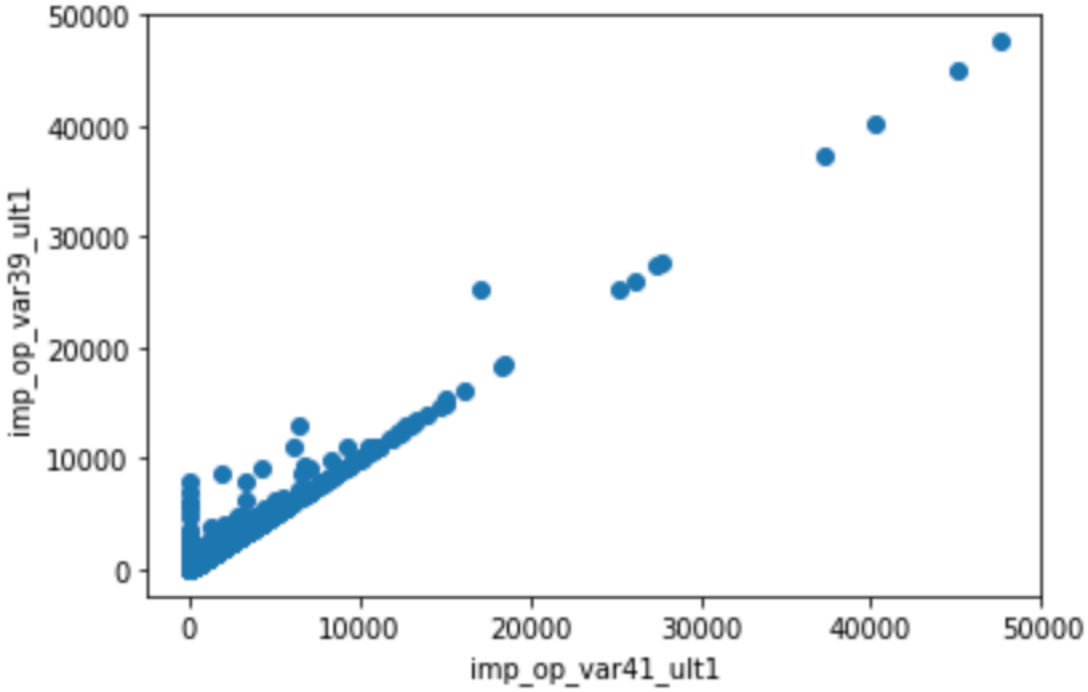
- 76020 data points
- 371 attributes
- Includes Target column
  - 1 - Unsatisfied Customer
  - 0 - Satisfied customer
- Unbalanced data:

0	73012	96.04%
1	3008	3.95%

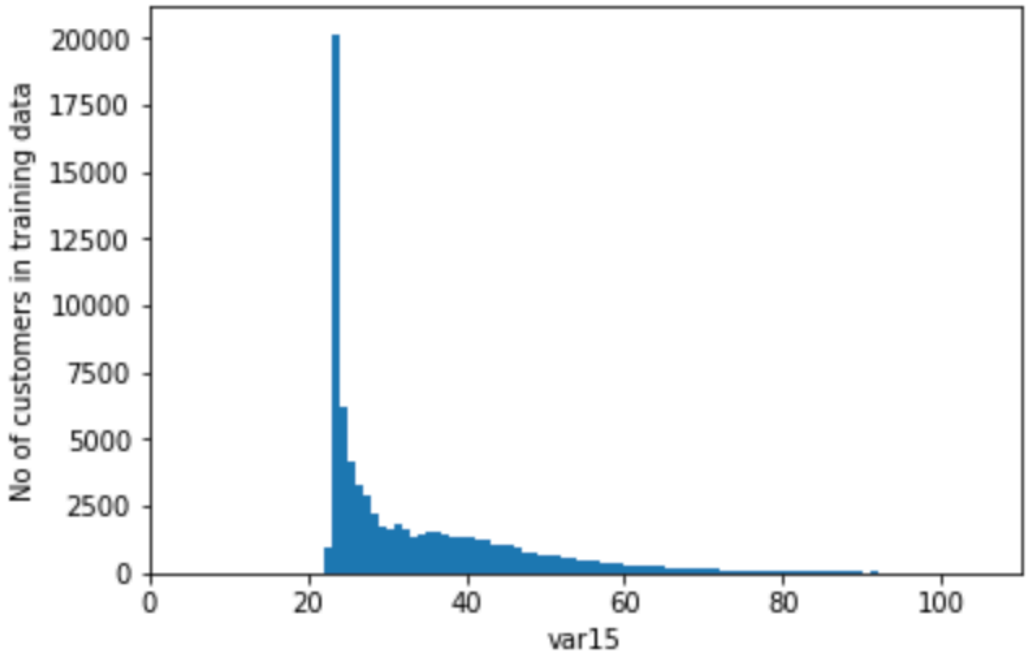
Test Data

- 75818 data points
- 370 attributes
- No Target column
  - Predict whether customer is satisfied (0) or unsatisfied (1)

Exploratory Data Analysis



High correlation is found among some attributes



Var15 has positively skewed distribution

Python based Predictive Model

DECISION TREE ALGORITHM

```
# BASE CASE - DEFAULT PARAMETERS
#Select just Target Column from training dataset
Y_Train = cpy_traindata.iloc[:, -1]

#Select features from training and test dataset
X_Train = cpy_traindata.iloc[:, :-1]
X_Test = cpy_testdata

#Create Decision Tree Classifier
clf=DecisionTreeClassifier()

#Apply Classifier on Train and Target
clf.fit(X_Train,Y_Train)

#Get Class Prediction as a data frame with header as Prediction
pred=pd.DataFrame(clf.predict(X_Train),columns=["Prediction"])

pred.head()
```

Uses default parameters:

Criterion: “Gini”
Splitter: “Best”
Max_depth: None
Min_samples_split: 2
Min_samples_leaf: 1
Max_leaf_nodes: None

Accuracy & Confusion Matrix

	Predicted Satisfied	Predicted Unsatisfied
Actual Satisfied	73102 (True Positive)	0 (False Positive)
Actual Unsatisfied	0 (False Negative)	3008 (True Negative)

Accuracy – 100%

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Submission.csv	just now	0 seconds	0 seconds	0.55526

Complete

Jump to your position on the leaderboard

Kaggle score on Test data | 0.55526


NOT SURE IF GOOD MODEL...
...OR JUST OVERFITTING
memegenerator.net

Model post Data Pre-processing

Data Pre-Processing

- Removed 34 attributes with 0 standard deviation

Data -[76020,337]

Data -[76020,308]

- Removed 29 attributes which are duplicates

- Removed ID column

Data -[76020,307]

Data -[76020,307]

- Replaced "-999999" with "2" (Most common value)

- Removed 76 columns which are highly correlated (>0.98)

Data -[76020,231]

DECISION TREE ALGORITHM

```
# BASE CASE - DEFAULT PARAMETERS
#Select just Target Column from training dataset
Y_Train = cpy_traindata.iloc[:, -1]

#Select features from training and test dataset
X_Train = cpy_traindata.iloc[:, :-1]
X_Test = cpy_testdata

#Create Decision Tree Classifier
clf=DecisionTreeClassifier()

#Apply Classifier on Train and Target
clf.fit(X_Train,Y_Train)

#Get Class Prediction as a data frame with header as Prediction
pred=pd.DataFrame(clf.predict(X_Train),columns=["Prediction"])

pred.head()
```

Uses default parameters:

Criterion: “Gini”
Splitter: “Best”
Max_depth: None
Min_samples_split: 2
Min_samples_leaf: 1
Max_leaf_nodes: None

Accuracy & Confusion Matrix

	Predicted Satisfied	Predicted Unsatisfied
	True Positive	False Positive
Actual Satisfied	73008	4
Actual Unsatisfied	312	2696
	False Negative	True Negative

Accuracy – 99.58%

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Submission_base.csv	just now	0 seconds	0 seconds	0.54360

Complete

Jump to your position on the leaderboard

Kaggle score on Test data | 0.5436

1 Parameters Variation to Decision Tree Algorithm

1

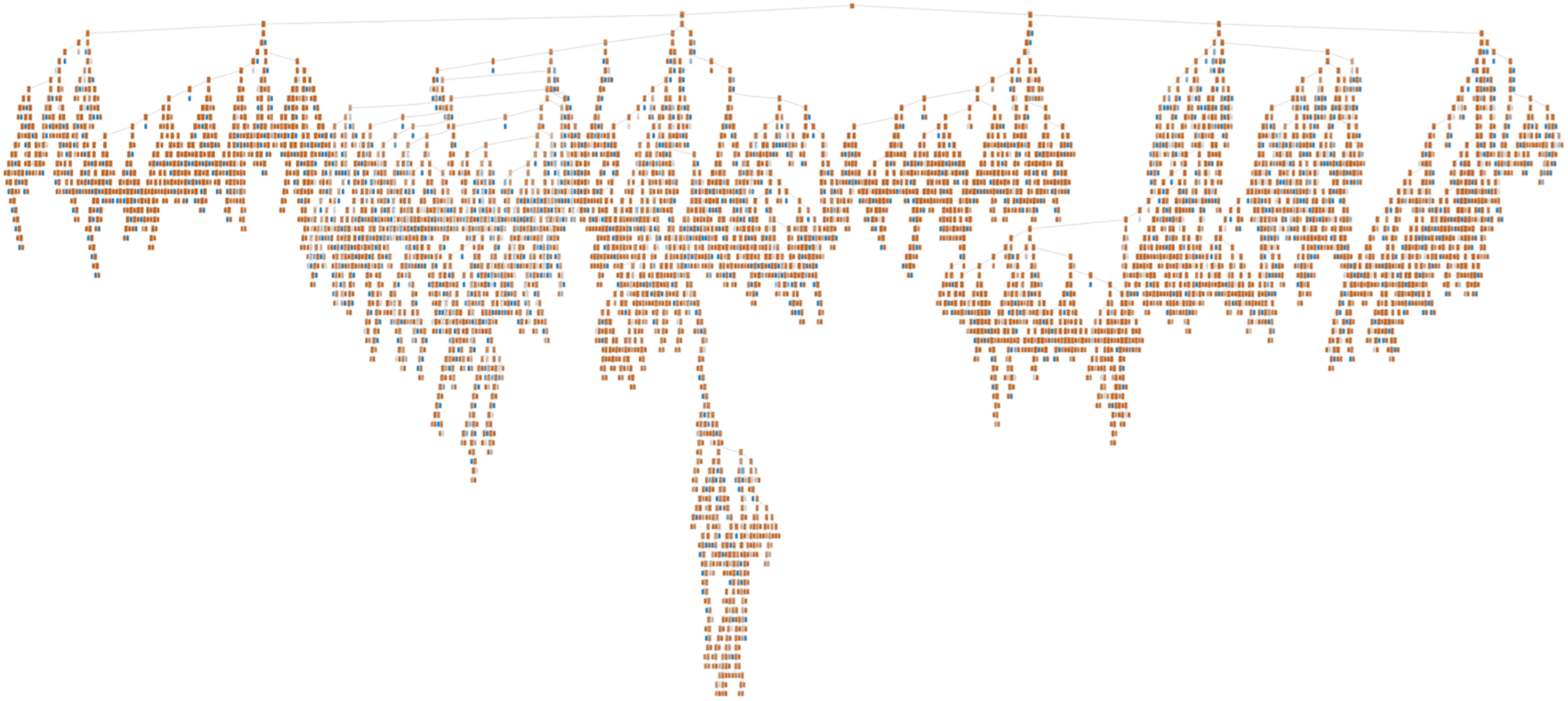
DECISION TREE ALGORITHM

```
# Changing parameters with min_samples_split
#Select just Target Column from training dat
Y_Train = cpy_traindata.iloc[:, -1]

#Select features from training and test data
X_Train = cpy_traindata.iloc[:, :-1]
X_Test =  cpy_testdata

#Create Decision Tree Classifier
clf=DecisionTreeClassifier(min_samples_split = 10)

#Apply Classifier on Train and Target
clf.fit(X_Train,Y_Train)

#Get Class Prediction as a data frame with header as Prediction
pred=pd.DataFrame(clf.predict(X_Train),columns=["Prediction"])

pred.head()
```

Uses all default parameters except:

Criterion: “Gini”
Splitter: “Best”
Max_depth: None
Min_samples_split: 10
Min_samples_leaf: 1
Max_leaf_nodes: None

Too huge to understand!



1

Accuracy & Confusion Matrix

	Predicted Satisfied	Predicted Unsatisfied
	True Positive	False Positive
Actual Satisfied	↓ 72754	↑ 258
Actual Unsatisfied	↑ 1406	↓ 1602
	False Negative	True Negative

↑ Increased compared to default parameters model

↓ Decreased compared to default parameters model

Accuracy – 97.81%

Predicts more False positives and False negatives

1

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Submission_min_samples_split.csv	just now	0 seconds	0 seconds	0.53422

Complete

Jump to your position on the leaderboard

Kaggle score on Test data | 0.53422

2 | Parameters Variation to Decision Tree Algorithm

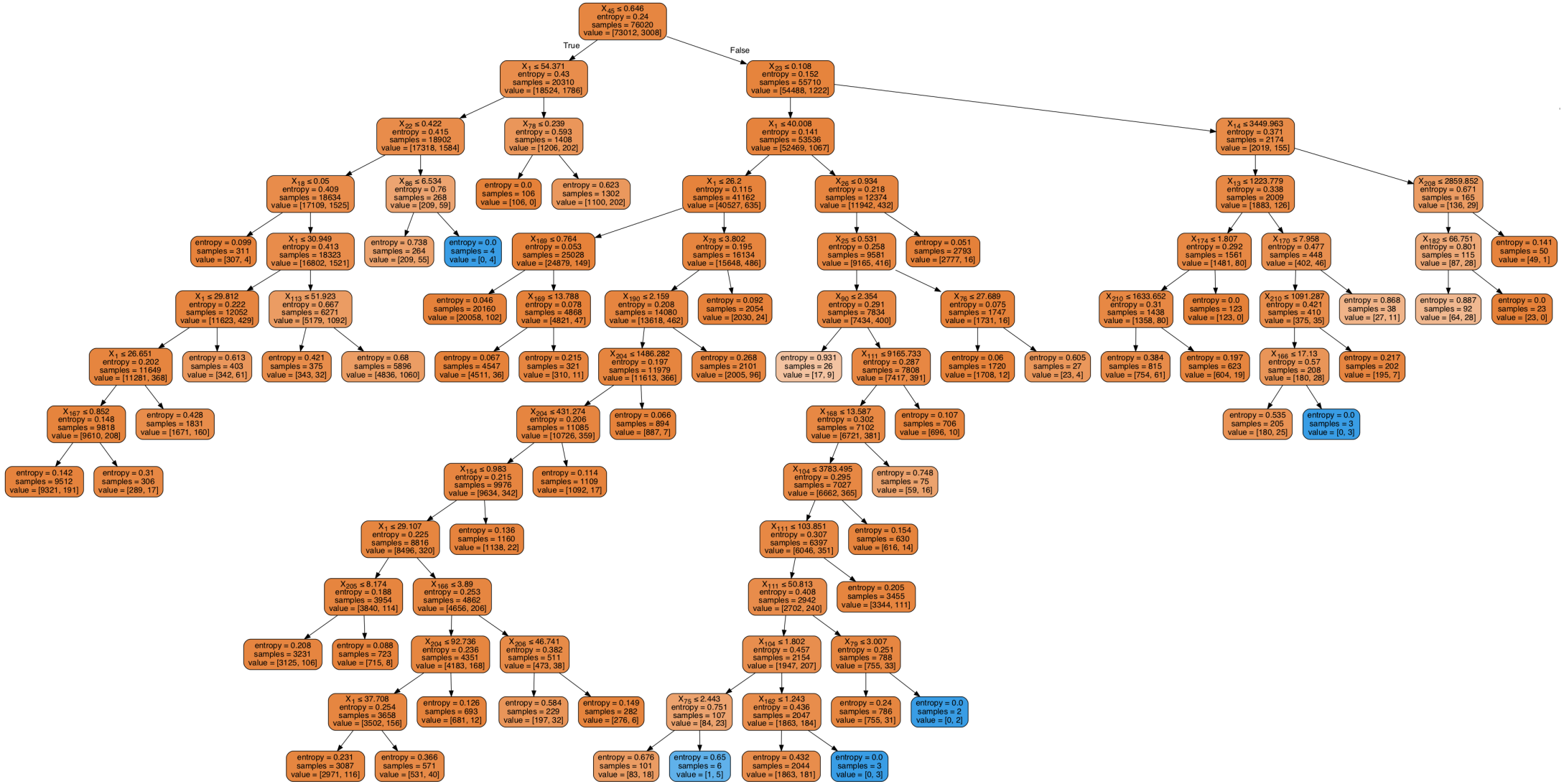
DECISION TREE ALGORITHM

```
# Changing parameters with entropy
#Select just Target Column from training dataset
Y_Train = cpy_traindata.iloc[:, -1]

#Select features from training and test dataset
X_Train = cpy_traindata.iloc[:, :-1]
X_Test =  cpy_testdata

#Create Decision Tree Classifier
clf=DecisionTreeClassifier(criterion = "entropy", splitter = "random", max_leaf_nodes = 50)

#Apply Classifier on Train and Target
clf.fit(X_Train,Y_Train)

#Get Class Prediction as a data frame with header as Prediction
pred=pd.DataFrame(clf.predict(X_Train),columns=["Prediction"])

pred.head()
```

Uses all default parameters except:

Criterion: “Entropy”
Splitter: “Random”
Max_depth: None
Min_samples_split: 2
Min_samples_leaf: 1
Max_leaf_nodes: 50

Pruned Decision tree



Accuracy & Confusion Matrix

	Predicted Satisfied	Predicted Unsatisfied
	True Positive	False Positive
Actual Satisfied	⇧ 73012	⬇ 0
Actual Unsatisfied	⇧ 2995	⬇ 13
	False Negative	True Negative

⇧ Increased compared to default parameters model

⬇ Decreased compared to default parameters model

Accuracy – 96.06%

Unable to predict true unsatisfied customers

2

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Submission_entropy.csv	just now	0 seconds	1 seconds	0.49995

Complete

Jump to your position on the leaderboard

Kaggle score on Test data | 0.49995

3 | Parameters Variation to Decision Tree Algorithm

DECISION TREE ALGORITHM

```
# Changing parameters with min_samples_split
#Select just Target Column from training data
Y_Train = cpy_traindata.iloc[:, -1]

#Select features from training and test data
X_Train = cpy_traindata.iloc[:, :-1]
X_Test =  cpy_testdata

#Create Decision Tree Classifier
clf=DecisionTreeClassifier(min_samples_leaf = 2)

#Apply Classifier on Train and Target
clf.fit(X_Train,Y_Train)

#Get Class Prediction as a data frame with header as Prediction
pred=pd.DataFrame(clf.predict(X_Train),columns=["Prediction"])

pred.head()
```

Uses all default parameters except:

Criterion: “Gini”
Splitter: “Best”
Max_depth: None
Min_samples_split: 2
Min_samples_leaf: 2
Max_leaf_nodes: None

Accuracy & Confusion Matrix

	Predicted Satisfied	Predicted Unsatisfied
	True Positive	False Positive
Actual Satisfied	↓ 72841	↑ 171
Actual Unsatisfied	↑ 1509	↓ 1499
	False Negative	True Negative

↑ Increased compared to default parameters model

↓ Decreased compared to default parameters model

Accuracy – 97.79%

Predicts more False positives and False negatives

3

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Submission_min_samples_leaf.csv	just now	0 seconds	1 seconds	0.53569

Complete

Jump to your position on the leaderboard

Kaggle score on Test data | 0.53569

COMPARISON – Kaggle Score

Without data pre-processing

With data pre-processing

Default Parameters	Default Parameters	Min sample leaf = 2	Min sample split = 10	Entropy Criterion
0.55526	0.5436	0.53569	0.53422	0.49995

Prediction accuracy

Limitation of Decision Trees

- Unstable - sensitive to small change in data
- High tendency for overfitting of data
- Out of sample predictions are not very accurate
- Tres get very complex with increase in number of attributes