**Code:**

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

data = pd.read\_csv("C:\\Users\CSE-14\Desktop\credit.csv")

data.isnull().sum()

data.dropna()

data.columns

data.info()

data = data.drop(["phone"], axis = 1)

desc = data.describe()

# Converting into Numeric

lb = LabelEncoder()

data["checking\_balance"] = lb.fit\_transform(data["checking\_balance"])

data["credit\_history"] = lb.fit\_transform(data["credit\_history"])

data["purpose"] = lb.fit\_transform(data["purpose"])

data["savings\_balance"] = lb.fit\_transform(data["savings\_balance"])

data["employment\_duration"] = lb.fit\_transform(data["employment\_duration"])

data["other\_credit"] = lb.fit\_transform(data["other\_credit"])

data["housing"] = lb.fit\_transform(data["housing"])

data["job"] = lb.fit\_transform(data["job"])

#data["default"]=lb.fit\_transform(data["default"])

data['default'].unique()

data['default'].value\_counts()

colnames = list(data.columns)

predictors = colnames[:15]

target = colnames[15]

# Splitting data into training and testing data set

from sklearn.model\_selection import train\_test\_split

train, test = train\_test\_split(data, test\_size = 0.3)

from sklearn.tree import DecisionTreeClassifier as DT

help(DT)

model = DT(criterion = 'entropy')

model.fit(train[predictors], train[target])

preds = model.predict(train[predictors])

pd.crosstab(train[target], preds, rownames = ['Actual'], colnames = ['Predictions'])

np.mean(preds == train[target])

# Prediction on Test Data

preds = model.predict(test[predictors])

pd.crosstab(test[target], preds, rownames=['Actual'], colnames=['Predictions'])

np.mean(preds == test[target]) # Test Data Accuracy

# Prediction on Train Data

# Train Data Accuracy

# Automatic Tuning - Hyperparameters

#####

# GridSearchCV