

**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
```