```
# For using local files from system, import os
import os
import pandas as pd
#Changing directory to our specified location
os.chdir(data_dir)
dat=pd.read_csv("credit_history.csv")
dat.head()
 ₹
                     default amount
                                                             grade
                                                                              years
                                                                                                    ownership
                                                                                                                               income
                                                                                                                                                 age
              0
                                    0
                                                1000
                                                                        В
                                                                                     2.0
                                                                                                              RENT 19200.0
                                                                                                                                                     24
                                     1
                                                6500
                                                                        Α
                                                                                     2.0
                                                                                                MORTGAGE
                                                                                                                              66000.0
                                                                                                                                                      28
              2
                                    0
                                                                                     2.0
                                                2400
                                                                        Α
                                                                                                              RENT
                                                                                                                              60000.0
                                                                                                                                                      36
                                                                        С
              3
                                    0
                                              10000
                                                                                     3.0
                                                                                                                              62000.0
                                                                                                              RENT
                                                                                                                                                     24
                                                                        С
                                                                                     2.0
                                                                                                              RENT 20000.0
                                     1
                                                4000
                                                                                                                                                     28
#Checking for null values
dat.isnull().sum()
 → default
                                                 0
                                                 0
            amount
            grade
                                                 0
            years
                                             279
            ownership
                                                 0
            income
                                                 0
            age
                                                 0
            dtype: int64
# Years column have null values which can be imputed by median here.
dat['years'].describe()
                                  7448.000000
 ₹
           count
                                          6.086332
            mean
            std
                                          6.700758
                                          0.000000
            min
            25%
                                          2.000000
                                         4.000000
            50%
            75%
                                          8.000000
            max
                                       62.000000
            Name: years, dtype: float64
dat['years'].fillna(4,inplace=True)
 🔂 C:\Users\abhis\AppData\Local\Temp\ipykernel_6336\3242087125.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or
            The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting value.
            For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value, inplace=True)' or df[col] = df[col] =
                 dat['years'].fillna(4,inplace=True)
           4
dat.isnull().sum()
 → default
            amount
                                            0
                                            0
            grade
            years
                                            0
            ownership
                                            0
                                            0
            income
            age
                                            0
            dtype: int64
```

dat.head()

<b>→</b>		default	amount	grade	years	ownership	income	age
-	0	0	1000	В	2.0	RENT	19200.0	24
	1	1	6500	Α	2.0	MORTGAGE	66000.0	28
:	2	0	2400	Α	2.0	RENT	60000.0	36
;	3	0	10000	С	3.0	RENT	62000.0	24
4	4	1	4000	С	2.0	RENT	20000.0	28
4								

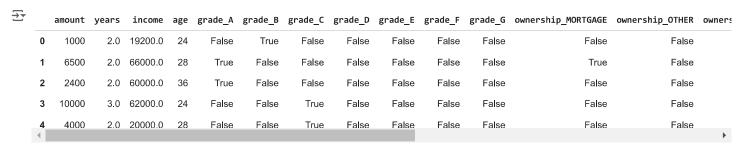
#no null values, now we remove default column for making it predictor matrix

X=dat.drop("default",axis=1)

# one hot encoding categorical variables

X=pd.get\_dummies(X)

X.head()



#Assigning default column as target variable

y=dat['default']

#Splitting predictor matrix into 80-20 ratio into training and testing set.

import sklearn.model\_selection as model\_selection
X\_train,X\_test,y\_train,y\_test=model\_selection.train\_test\_split(X,y,test\_size=0.2,random\_state=200)

import sklearn.tree as tree
clf=tree.DecisionTreeClassifier(max\_depth=3,random\_state=200)
clf.fit(X\_train,y\_train)
clf.score(X\_test,y\_test)

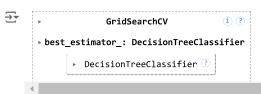
→ 0.6274256144890039

# We get accuracy score as 0.63(approx) with above parameters, now we try to get best parameters

## Grid Search-Cross Validation

clf=tree.DecisionTreeClassifier(max\_depth=3,random\_state=200)

 $\label{local_model_selection.GridSearchCV(clf,param_grid=\{'max_depth':[2,3,4,5,6]\})} \\ mod.fit(X_train,y_train)$ 





We can see, for depth=2, we get the best accuracy (0.63). This is how a simple classification model can be built from decision trees.

Start coding or generate with AI.