```
import pandas as pd
df = pd.read_csv("/content/German_Credit_Card_Dataset.csv")
df.info()
df.shape
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   checkin_acc       1000 non-null   object
 1   duration          1000 non-null   int64 
 2   credit_history    1000 non-null   object
 3   amount            1000 non-null   int64 
 4   savings_acc       1000 non-null   object
 5   present_emp_since 1000 non-null   object
 6   inst_rate         1000 non-null   int64 
 7   personal_status   1000 non-null   object
 8   residing_since    1000 non-null   int64 
 9   age               1000 non-null   int64 
 10  inst_plans        1000 non-null   object
 11  num_credits       1000 non-null   int64 
 12  job               1000 non-null   object
 13  status            1000 non-null   int64 
dtypes: int64(7), object(7)
memory usage: 109.5+ KB
```

	checkin_acc	duration	credit_history	amount	savings_acc	present_emp_since	inst_r
0	A11	6	A34	1169	A65	A75	
1	A12	48	A32	5951	A61	A73	
2	A14	12	A34	2096	A61	A74	
3	A11	42	A32	7882	A61	A74	
4	A11	24	A33	4870	A61	A73	

```
df.iloc[0:5,0:7]
```

	checkin_acc	duration	credit_history	amount	savings_acc	present_emp_since	inst_r
0	A11	6	A34	1169	A65	A75	
1	A12	48	A32	5951	A61	A73	
2	A14	12	A34	2096	A61	A74	
3	A11	42	A32	7882	A61	A74	
4	A11	24	A33	4870	A61	A73	

```
#print the first five records and rem col
df.iloc[0:5,:]
```

	checkin_acc	duration	credit_history	amount	savings_acc	present_emp_since	inst_r
0	A11	6	A34	1169	A65	A75	
1	A12	48	A32	5951	A61	A73	
2	A14	12	A34	2096	A61	A74	
3	A11	42	A32	7882	A61	A74	
4	A11	24	A33	4870	A61	A73	

```
df['checkin_acc'].unique()
```

```
array(['A11', 'A12', 'A14', 'A13'], dtype=object)
```

```
x_features=list(df.columns)
x_features.remove('status')
encoded_df=pd.get_dummies(df[x_features],drop_first=True)
print(list(encoded_df.columns))
```

```
['duration', 'amount', 'inst_rate', 'residing_since', 'age', 'num_credits', 'checkin_acc
```

```
x=encoded_df
y=df['status']
```

```
#Divide data into 78% training and 30% as testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier(criterion='gini',max_depth=3)
```

```
clf.fit(x_train,y_train)
```

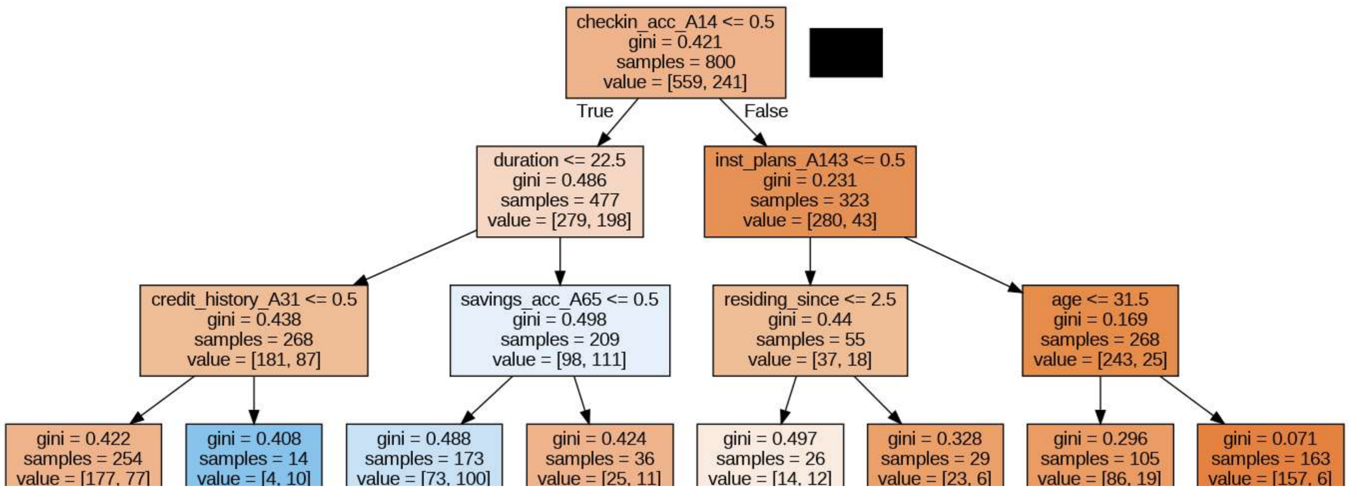
```
▾      DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3)
```

```
pred_y=clf.predict(x_test)
```

```
from sklearn import metrics
print("Confusion Matrix is\n",metrics.accuracy_score(pred_y ,y_test))
print("Accuracy is",metrics.precision_score(pred_y,y_test))
print("AUC Score is",metrics.recall_score(pred_y,y_test))
```

```
Confusion Matrix is
 0.76
Accuracy is 0.5084745762711864
AUC Score is 0.6122448979591837
```

```
from sklearn.tree import export_graphviz
import pydotplus as pdot
from IPython.display import Image
export_graphviz(clf,out_file='tree.odt',feature_names=x_train.columns,filled=True)
graph=pdot.graphviz.graph_from_dot_file("tree.odt")
graph.write_png("tree.png")
Image(filename="tree.png")
```


checkin_acc_A14 <= 0.5
gini = 0.421
samples = 800
value = [559, 241]
[redacted]
True
False
duration <= 22.5
gini = 0.486
samples = 477
value = [279, 198]
inst_plans_A143 <= 0.5
gini = 0.231
samples = 323
value = [280, 43]
credit_history_A31 <= 0.5
gini = 0.438
samples = 268
value = [181, 87]
savings_acc_A65 <= 0.5
gini = 0.498
samples = 209
value = [98, 111]
residing_since <= 2.5
gini = 0.44
samples = 55
value = [37, 18]
age <= 31.5
gini = 0.169
samples = 268
value = [243, 25]
gini = 0.422
samples = 254
value = [177, 77]
gini = 0.408
samples = 14
value = [4, 10]
gini = 0.488
samples = 173
value = [73, 100]
gini = 0.424
samples = 36
value = [25, 11]
gini = 0.497
samples = 26
value = [14, 12]
gini = 0.328
samples = 29
value = [23, 6]
gini = 0.296
samples = 105
value = [86, 19]
gini = 0.071
samples = 163
value = [157, 6]