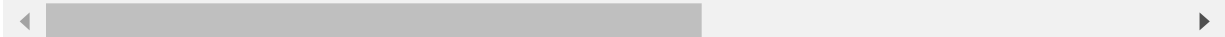


```
In [1]: from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: # Data Collection
data_loan = pd.read_csv("loan_borrower_data.csv")
data_loan.head()
```

```
Out[2]:
```

	credit.policy		purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.lin
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.95833	
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.00000	
2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	4710.00000	
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	2699.95833	
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	4066.00000	



```
In [3]: # Data Wrangling
X = data_loan.iloc[:,2:13]
Y = data_loan["not.fully.paid"]
```

```
In [4]: # data splitting
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=10)
```

```
In [5]: # Model Creation
random_cls = RandomForestClassifier()
random_cls.fit(x_train, y_train)
```

```
Out[5]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                               criterion='gini', max_depth=None, max_features='auto',
                               max_leaf_nodes=None, max_samples=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, n_estimators=100,
                               n_jobs=None, oob_score=False, random_state=None,
                               verbose=0, warm_start=False)
```

```
In [6]: # Data Prediction
predicted_value = random_cls.predict(x_test)
```

```
In [7]: # Check Accuracy Score
metrics.accuracy_score(predicted_value, y_test)
```

```
Out[7]: 0.8420320111343076
```

In []: