

```
import pandas as pd
```

import or get the consumer data

Creating a small dataset with perfect separation

```
data = {
    'Credit_Score': [700, 650, 600, 750, 680, 620, 720, 690],
    'Income': [50000, 60000, 45000, 70000, 55000, 48000, 65000,
58000],
    'Debt_to_Income_Ratio': [0.3, 0.4, 0.5, 0.2, 0.35, 0.45, 0.25,
0.38],
    'Default_on_Loan': ['No', 'No', 'Yes', 'No', 'No', 'Yes', 'No',
'Yes']
}
```

print the data

```
data
{'Credit_Score': [700, 650, 600, 750, 680, 620, 720, 690],
 'Income': [50000, 60000, 45000, 70000, 55000, 48000, 65000, 58000],
 'Debt_to_Income_Ratio': [0.3, 0.4, 0.5, 0.2, 0.35, 0.45, 0.25, 0.38],
 'Default_on_Loan': ['No', 'No', 'Yes', 'No', 'No', 'Yes', 'No',
'Yes']}
```

convert the data into two dimensional data structure

```
df_credit = pd.DataFrame(data)
```

```
df_credit
```

	Credit_Score	Income	Debt_to_Income_Ratio	Default_on_Loan
0	700	50000	0.30	No
1	650	60000	0.40	No
2	600	45000	0.50	Yes
3	750	70000	0.20	No
4	680	55000	0.35	No
5	620	48000	0.45	Yes
6	720	65000	0.25	No
7	690	58000	0.38	Yes

import libraries

decision tree - to solve classification problems and categorize objects depending on their learning features.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

separate features and target

```
x_credit = df_credit[['Credit_Score', 'Income',  
'Debt_to_Income_Ratio']]  
y_credit = df_credit['Default_on_Loan']
```

train a decision tree classifier

```
clf_credit = DecisionTreeClassifier()  
clf_credit.fit(x_credit, y_credit)  
  
DecisionTreeClassifier()
```

evaluate accuracy on training data

```
y_pred_credit = clf_credit.predict(x_credit)  
accuracy_credit = accuracy_score(y_credit, y_pred_credit)  
print(f'Accuracy on training data: {accuracy_credit * 100:.2f}%')  
  
Accuracy on training data: 100.00%
```

now let's use the trained model to predict outcomes for new samples

```
new_samples_credit = {  
    'Credit_Score': [680, 720, 600],  
    'Income': [55000, 65000, 50000],  
    'Debt_to_Income_Ratio': [0.33, 0.28, 0.42]  
}  
  
new_df_credit = pd.DataFrame(new_samples_credit)  
  
# Predict outcomes for new samples  
new_predictions_credit = clf_credit.predict(new_df_credit)  
  
# Display predictions for new samples  
for i, pred in enumerate(new_predictions_credit):  
    print(f"Sample {i + 1}: Predicted Default on Loan - {pred}")  
  
Sample 1: Predicted Default on Loan - No  
Sample 2: Predicted Default on Loan - No  
Sample 3: Predicted Default on Loan - Yes
```

Result

A python program was developed using decision tree algorithm based on customer credit information to predict loan defaulter.