

```
In [26]: #importing all the libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree

data=pd.read_csv("/kaggle/input/load-re-payment-data/Decision_Tree_ Dataset.csv",so
```

```
In [27]: #Top 5 records observation
data.head()
```

```
Out[27]:
```

	1	2	3	4	sum	Unnamed: 5
0	201	10018	250	3046	13515	yes
1	205	10016	395	3044	13660	yes
2	257	10129	109	3251	13746	yes
3	246	10064	324	3137	13771	yes
4	117	10115	496	3094	13822	yes

```
In [28]: #Columns information, missing value, data type, columns name
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0    1           1000 non-null   int64
1    2           1000 non-null   int64
2    3           1000 non-null   int64
3    4           1000 non-null   int64
4    sum         1000 non-null   int64
5    Unnamed: 5  1000 non-null   object
dtypes: int64(5), object(1)
memory usage: 47.0+ KB
```

```
In [29]: #Shape of dataframe
data.shape
```

```
Out[29]: (1000, 6)
```

```
In [30]: #Dropping the 'sum' column from data
data.drop(columns="sum",inplace=True)
```

```
In [31]: #Top 5 records observation
data.head()
```

```
Out[31]:
```

	1	2	3	4	Unnamed: 5
0	201	10018	250	3046	yes
1	205	10016	395	3044	yes
2	257	10129	109	3251	yes
3	246	10064	324	3137	yes
4	117	10115	496	3094	yes

Splitting the indipendent and dipendent variables

```
In [62]: X=data.values[:,0:4]
Y=data.values[:, -1]
```

Splitting the 'Train' and Test data from ML

```
In [72]: X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=100)

Classification=DecisionTreeClassifier(criterion="entropy",random_state=100,max_dep
```

Fitting the ML Model in DecisionTreeClassifier for Training

```
In [73]: Classification.fit(X_train,Y_train)
```

```
Out[73]:
```

DecisionTreeClassifier

DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=5, random_state=100)

Predicting the Indipendent varibale

```
In [75]: Y_Predict=Classification.predict(X_test)
```

```
In [76]: Y_Predict
```

```
Out[76]: array(['yes', 'yes', 'No', 'yes', 'No', 'yes', 'yes', 'No', 'No', 'No',  
            'No', 'No', 'yes', 'No', 'No', 'No', 'yes', 'No', 'yes', 'No',  
            'No', 'yes', 'No', 'yes', 'yes', 'No', 'No', 'yes', 'No', 'No',  
            'No', 'yes', 'yes', 'yes', 'yes', 'No', 'No', 'No', 'yes', 'No',  
            'yes', 'yes', 'yes', 'yes', 'No', 'yes', 'No', 'yes', 'No', 'No',  
            'yes', 'No', 'yes', 'yes', 'yes', 'yes', 'No', 'yes', 'No', 'yes',  
            'yes', 'No', 'No', 'yes', 'No', 'yes', 'yes', 'yes', 'No', 'yes',  
            'No', 'No', 'No', 'yes', 'No', 'yes', 'yes', 'No', 'yes', 'No',  
            'No', 'No', 'No', 'yes', 'No', 'yes', 'No', 'yes', 'yes', 'No',  
            'yes', 'yes', 'yes', 'yes', 'yes', 'No', 'yes', 'yes', 'yes',  
            'yes', 'No', 'No', 'yes', 'yes', 'No', 'yes', 'yes', 'yes', 'No',  
            'No', 'yes', 'yes', 'yes', 'No', 'No', 'yes', 'yes', 'yes', 'No',  
            'No', 'No', 'No', 'yes', 'yes', 'No', 'yes', 'yes', 'yes', 'No',  
            'No', 'yes', 'yes', 'No', 'yes', 'yes', 'yes', 'No', 'yes', 'yes',  
            'No', 'No', 'No', 'yes', 'yes', 'No', 'yes', 'yes', 'yes', 'No',  
            'No', 'No', 'No', 'yes', 'No', 'No', 'yes', 'yes', 'yes', 'yes',  
            'yes', 'No', 'yes', 'No', 'yes', 'No', 'No', 'No', 'yes', 'yes',  
            'No'], dtype=object)
```

Testing the Accuracy of model

```
In [78]: print("Accuracy Score",accuracy_score(Y_test,Y_Predict)*100)
```

Accuracy Score 93.0

```
In [ ]:
```