

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

To predict if a customer will repay loan amount or not using decision tree algorithm

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

```
df = pd.read_csv('/content/drive/MyDrive/Copy of Decision_Tree_Dataset.csv')
```

[+ Code](#)
[+ Text](#)

```
len(df)
```

```
1000
```

```
df.isnull().sum()
```

```
1          0
2          0
3          0
4          0
sum         0
Unnamed: 5   0
dtype: int64
```

```
df.duplicated().sum()
```

```
0
```

```
df.nunique()
```

```
1          369
2          917
3          614
4          773
sum         901
Unnamed: 5    2
dtype: int64
```

```
df.shape
```

```
(1000, 6)
```

```
df.head()
```

	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

```
df = df.rename(columns= {'1':'Initial Payment', '2':'Last payment', '3': 'Credit score', '4':'House number', 'Unnamed: 5':'Result' })
```

```
df.head()
```

	Initial Payment	Last payment	Credit score	House number	Result
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
```

```
X = df.drop('Result', axis=1)
y = df['Result']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.2, random_state=42)
```

```
#function to perform trainng with entrophy
```

```
clf_entropy = DecisionTreeClassifier(criterion = "entropy", random_state=100, max_depth=3, min_samples_leaf =5)
clf_entropy.fit(X_train, y_train)
```

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

```
y_pred = clf_entropy.predict(X_test)
y_pred
```

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

```
accuracy_score(y_test, y_pred)
```

```
0.93
```

```
# Function to get user input for each feature
def get_user_input(prompt):
    return float(input(prompt + ": "))

# Get user input for new data
initial_payment = get_user_input('Enter Initial Payment')
last_payment = get_user_input('Enter Last Payment')
credit_score = get_user_input('Enter Credit Score')
house_number = get_user_input('Enter House Number')

# Create new data as a DataFrame
new_data = pd.DataFrame({
    'Initial Payment': [initial_payment],
    'Last payment': [last_payment],
    'Credit score': [credit_score],
    'House number': [house_number]
})

# Make prediction for the new data
prediction = clf_entropy.predict(new_data)

# Display the prediction
print("Prediction:", prediction[0])

Enter Initial Payment: 567
Enter Last Payment: 567
Enter Credit Score: 567
Enter House Number: 4322
Prediction: yes
```

The bank can use this model to decide whether it should approve loan request from from a particular customer or not