Packages to import

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import defaultdict
%matplotlib inline
```

In [ ]:
```
Train=pd.read_csv("Loan Prediction.csv")
```

In [ ]:
```
Train.head()
```

In [ ]:
```

```

In [ ]:
```
Train.head()
```

In [ ]:
```
Train.info()
```

In [ ]:
```
Train.columns
```

In [ ]:
```
Train.shape
```

In [ ]:
```
Train["Loan_Status"].value_counts()
```

In [ ]:
```
Train["Loan_Status"].value_counts(normalize=True)
```

In [ ]:
```
Train["Loan_Status"].value_counts().plot.bar()
```

In [ ]:
```
Train.isnull().sum()
```

In [ ]:
```
Train['Gender'].fillna(Train['Gender'].mode()[0], inplace=True)
Train['Married'].fillna(Train['Married'].mode()[0], inplace=True)
Train['Dependents'].fillna(Train['Dependents'].mode()[0], inplace=True)
Train['Self_Employed'].fillna(Train['Self_Employed'].mode()[0], inplace=True)
Train['Credit_History'].fillna(Train['Credit_History'].mode()[0], inplace=True)
```

In [ ]:
```
Train['LoanAmount'].fillna(Train['LoanAmount'].median(), inplace=True)
```

In [ ]:
```
Train.isnull().sum()
```

In [ ]:
```
Train['Loan_Amount_Term'].fillna(Train['Loan_Amount_Term'].mode()[0], inplace=True)
```

In [ ]:
```
Train.isnull().sum()
```

In [ ]:
```
Train['LoanAmount_log'] = np.log(Train['LoanAmount'])
Train['LoanAmount_log'].hist(bins=20)
Train['Loan_Status'] = Train['Loan_Status']
```

In [ ]:
```
Train.head()
```

In [ ]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
Train['Gender'] = label_encoder.fit_transform(Train['Gender'])
Train['Married'] = label_encoder.fit_transform(Train['Married'])
Train['Education'] = label_encoder.fit_transform(Train['Education'])
Train['Self_Employed'] = label_encoder.fit_transform(Train['Self_Employed'])
Train['Property_Area'] = label_encoder.fit_transform(Train['Property_Area'])
Train['Loan_Status'] = label_encoder.fit_transform(Train['Loan_Status'])

print(Train.head())
```

In [ ]:
```
index = ['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
       'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area','LoanAmount_log',
'Loan_Status',]
```

In [ ]:
```
Train = Train[index]
```

In [ ]:
```
X = Train.iloc[:, :-1].values
y = Train.iloc[:, -1].values
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, ran
dom_state = 0)
```

In [ ]:
```
y_train
```

In [ ]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier(criterion="entropy")
tree_1 = DT.fit(X_train, y_train)
```

In [ ]:
```
y_pred = DT.predict(X_test)
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```