

```
In [132]: import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

```
In [133]: data=pd.read_csv("C:/Users/NARESH SANA/Downloads/Loan.csv")
```

```
In [134]: data
```

```
Out[134]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849
1	LP001003	Male	Yes	1	Graduate	No	4583
2	LP001005	Male	Yes	0	Graduate	Yes	3000
3	LP001006	Male	Yes	0	Not Graduate	No	2583
4	LP001008	Male	No	0	Graduate	No	6000
...
609	LP002978	Female	No	0	Graduate	No	2900
610	LP002979	Male	Yes	3+	Graduate	No	4106
611	LP002983	Male	Yes	1	Graduate	No	8072
612	LP002984	Male	Yes	2	Graduate	No	7583
613	LP002990	Female	No	0	Graduate	Yes	4583

614 rows × 13 columns



```
In [135]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Loan_ID                               614 non-null    object
1   Gender                                601 non-null    object
2   Married                               611 non-null    object
3   Dependents                            599 non-null    object
4   Education                             614 non-null    object
5   Self_Employed                         582 non-null    object
6   ApplicantIncome                       614 non-null    int64
7   CoapplicantIncome                     614 non-null    float64
8   LoanAmount                            592 non-null    float64
9   Loan_Amount_Term                       600 non-null    float64
10  Credit_History                         564 non-null    float64
11  Property_Area                          614 non-null    object
12  Loan_Status                            614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
In [136]: data1=data.drop(['Loan_ID','Self_Employed','CoapplicantIncome','Loan_Amount'])
```

```
In [137]: data1
```

```
Out[137]:
```

	Education	ApplicantIncome	LoanAmount	Loan_Status
0	Graduate	5849	NaN	Y
1	Graduate	4583	128.0	N
2	Graduate	3000	66.0	Y
3	Not Graduate	2583	120.0	Y
4	Graduate	6000	141.0	Y
...
609	Graduate	2900	71.0	Y
610	Graduate	4106	40.0	Y
611	Graduate	8072	253.0	Y
612	Graduate	7583	187.0	Y
613	Graduate	4583	133.0	N

614 rows × 4 columns

```
In [138]: data1.isnull().sum()
```

```
Out[138]: Education      0
ApplicantIncome    0
LoanAmount         22
Loan_Status        0
dtype: int64
```

```
In [139]: data1['LoanAmount'].mean()
```

```
Out[139]: 146.41216216216216
```

```
In [140]: data1['LoanAmount'].median()
```

```
Out[140]: 128.0
```

```
In [141]: data1['LoanAmount']=data1['LoanAmount'].fillna(value=146)  
data1
```

```
Out[141]:
```

	Education	ApplicantIncome	LoanAmount	Loan_Status
0	Graduate	5849	146.0	Y
1	Graduate	4583	128.0	N
2	Graduate	3000	66.0	Y
3	Not Graduate	2583	120.0	Y
4	Graduate	6000	141.0	Y
...
609	Graduate	2900	71.0	Y
610	Graduate	4106	40.0	Y
611	Graduate	8072	253.0	Y
612	Graduate	7583	187.0	Y
613	Graduate	4583	133.0	N

614 rows × 4 columns

```
In [142]: data1.isnull().sum()
```

```
Out[142]: Education      0  
ApplicantIncome    0  
LoanAmount         0  
Loan_Status        0  
dtype: int64
```

```
In [143]: data1['Loan_Status']=data1['Loan_Status'].map({'Y':1,'N':0})  
data1
```

```
Out[143]:
```

	Education	ApplicantIncome	LoanAmount	Loan_Status
0	Graduate	5849	146.0	1
1	Graduate	4583	128.0	0
2	Graduate	3000	66.0	1
3	Not Graduate	2583	120.0	1
4	Graduate	6000	141.0	1
...
609	Graduate	2900	71.0	1
610	Graduate	4106	40.0	1
611	Graduate	8072	253.0	1
612	Graduate	7583	187.0	1
613	Graduate	4583	133.0	0

614 rows × 4 columns

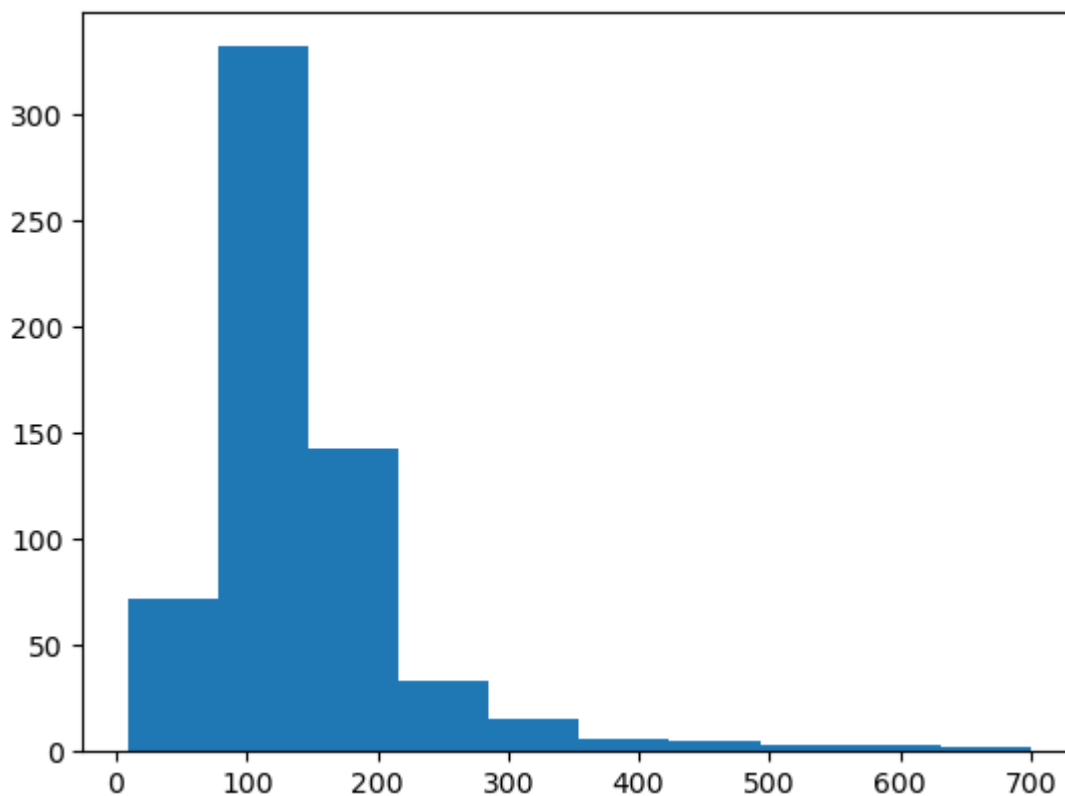
```
In [145]: import matplotlib.pyplot as plt  
plt.hist(data1['ApplicantIncome'])
```

```
Out[145]: (array([533.,  59.,  15.,   0.,   4.,   0.,   1.,   1.,   0.,   1.]),  
          array([ 150.,  8235., 16320., 24405., 32490., 40575., 48660., 56745.,  
                64830., 72915., 81000.]),  
          <BarContainer object of 10 artists>)
```



```
In [146]: plt.hist(data1['LoanAmount'])
```

```
Out[146]: (array([ 72., 332., 143.,  33.,  15.,   6.,   5.,   3.,   3.,   2.]),  
          array([  9.,  78.1, 147.2, 216.3, 285.4, 354.5, 423.6, 492.7, 561.8,  
                630.9, 700. ]),  
          <BarContainer object of 10 artists>)
```



In []:

In []: `X = data1.iloc[:, :-1]`In []: `X`In []: `y=data1.iloc[:, -1]`In []: `y`In [147]: `X =pd.get_dummies(X, dtype=int)`
`X`

Out[147]:

	ApplicantIncome	LoanAmount	Education_Graduate	Education_Not Graduate
0	5849	146.0	1	0
1	4583	128.0	1	0
2	3000	66.0	1	0
3	2583	120.0	0	1
4	6000	141.0	1	0
...
609	2900	71.0	1	0
610	4106	40.0	1	0
611	8072	253.0	1	0
612	7583	187.0	1	0
613	4583	133.0	1	0

614 rows × 4 columns

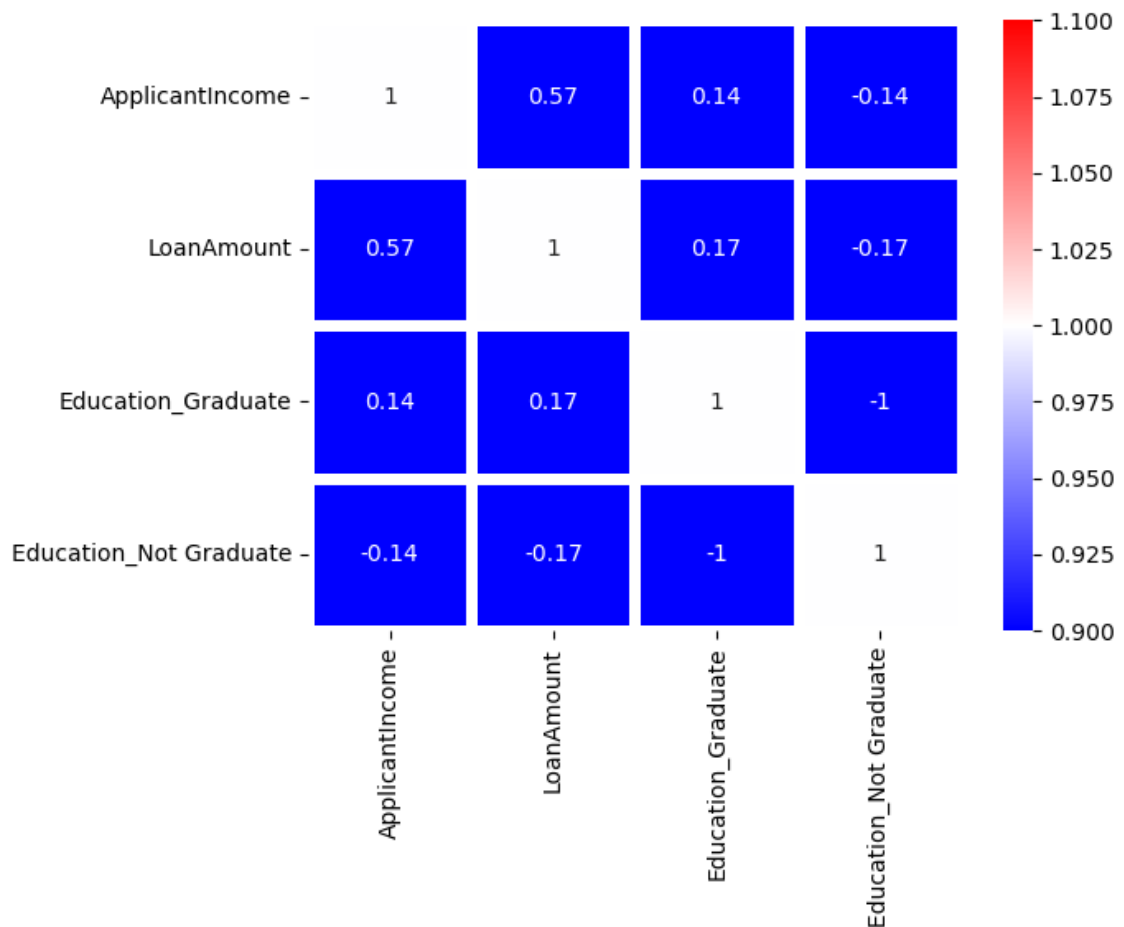
In [159]: `cor_mat = X.corr()`
`cor_mat`

Out[159]:

	ApplicantIncome	LoanAmount	Education_Graduate	Education_Not Graduate
ApplicantIncome	1.000000	0.565621	0.140760	-0.140760
LoanAmount	0.565621	1.000000	0.167041	-0.167041
Education_Graduate	0.140760	0.167041	1.000000	-1.000000
Education_Not Graduate	-0.140760	-0.167041	-1.000000	1.000000

```
In [160]: import seaborn as sns
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[160]: <Axes: >



```
In [149]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=20,random_stat
```

```
In [150]: X_train.count()
```

```
Out[150]: ApplicantIncome      594
LoanAmount                    594
Education_Graduate            594
Education_Not Graduate        594
dtype: int64
```

```
In [151]: X_test.count()
```

```
Out[151]: ApplicantIncome      20
LoanAmount                    20
Education_Graduate            20
Education_Not Graduate        20
dtype: int64
```

```
In [152]: y_train.count()
```

```
Out[152]: 594
```

```
In [153]: y_test.count()
```

```
Out[153]: 20
```

```
In [154]: from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(X_train,y_train)
```

```
Out[154]: LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [155]: y_pred=reg.predict(X_test)
y_pred
```

```
Out[155]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
              dtype=int64)
```

```
In [156]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
Out[156]: array([[ 0,  4],
                 [ 0, 16]], dtype=int64)
```

```
In [157]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

```
Out[157]: 0.8
```

```
In [ ]:
```