

Unsupervised Classification of Loan Approval Data

At some point in each of our lives, we will most likely consider taking out a loan. Whether that be for school, a car, a house, a business, or a personal project. With the increasing amount of algorithmic use in the finance industry, this project will seek to see if unsupervised classification method, Nonnegative Matrix Factorization, is a good option for loan approval decision making. Then, we will see if a recommender system can be built for the loan amount based on similar factors in loans that have been approved. The data being used is from a kaggle dataset here:

<https://www.kaggle.com/datasets/bhavikjikadara/loan-status-prediction>

```
In [81]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy
```

EDA and Cleaning

For the NFM, all factors must be numerical and non-negative. We will take a look at the values of each column to know the transformations needed.

```
In [78]: loan_data = pd.read_csv('data/loan_data.csv')

for col in loan_data.columns[1:]:
    print(f'{col} : {loan_data[col].unique()}')
```

Gender : ['Male' 'Female' nan]
 Married : ['Yes' 'No']
 Dependents : ['1' '0' '2' '3+' nan]
 Education : ['Graduate' 'Not Graduate']
 Self_Employed : ['No' 'Yes' nan]
 ApplicantIncome : [4583 3000 2583 6000 2333 3200 2500 1853 1299 4950 3596 3510 4887 2600
 7660 3365 2799 4226 1442 3750 3167 4692 3500 1828 3667 3748 3600 1800
 2400 3941 4695 3410 5649 5821 2645 4000 1928 3086 4230 4616 2708 2132
 3366 3357 3029 4166 7100 4300 1875 4755 1000 3333 3846 3988 2366 2958
 3273 4133 3620 2484 1977 4188 1759 4288 3052 7333 2071 5316 2929 5050
 2214 2137 2957 3692 2014 2718 3459 4895 3316 4200 2698 2330 1538 4860
 6277 2281 3254 2980 1863 3089 9323 2439 2237 1820 3522 4344 3497 2045
 1916 4600 3625 2178 2383 4885 3858 4191 1907 3416 3917 4408 3244 3975
 2479 3418 3430 5703 3173 3850 150 3727 5000 4283 2221 4009 2971 6250
 4735 2491 3716 3189 3155 3463 3812 3315 5819 2510 3406 6050 9703 6608
 2882 1809 3427 6045 5250 4931 2060 7200 5166 4708 2876 3237 2833 2620
 3900 2750 3103 4100 3927 2301 1811 3158 3704 4124 3075 4400 3153 5417
 4416 4666 2875 1625 3762 2917 2927 2507 5746 3399 3717 2058 3541 4342
 3166 4917 4333 4384 2935 5503 4160 2378 3083 2666 5500 2423 3813 3875
 4723 4750 3013 6822 6216 5124 3062 2764 4817 8750 4310 3069 5391 7167
 4566 2346 5488 1993 3100 3276 3180 3033 3902 1500 2889 2755 1963 4547
 2167 2213 3867 2253 2995 1025 3246 5829 2720 7250 4606 5935 2920 2717
 8624 6500 2425 1926 7142 4707 3466 4652 3539 3340 2309 1958 3948 2483
 7085 3859 4301 4354 7740 5191 2947 210 2653 4691 2873 3095 2083 3547
 2435 2699 3691 3597 3326 4625 2895 645 3159 4050 3814 3583 3598 3283
 2130 2031 3074 3400 2192 5677 3775 2679 6783 4281 3588 2550 3617 4608
 2138 3652 2239 3017 3358 2526 2785 3593 5468 2667 3675 5800 4467 6406
 3087 3229 1782 1836 1880 2787 2297 2165 2726 3833 2987 3232 2900 4106]
 CoapplicantIncome : [1.50800000e+03 0.00000000e+00 2.35800000e+03 1.51600000e+03
 7.00000000e+02 1.84000000e+03 2.84000000e+03 1.08600000e+03
 3.50000000e+03 1.91100000e+03 1.91700000e+03 2.25300000e+03
 1.04000000e+03 2.08300000e+03 1.66700000e+03 1.33000000e+03
 1.45900000e+03 1.66800000e+03 1.21300000e+03 2.33600000e+03
 3.44000000e+03 2.27500000e+03 1.64400000e+03 1.16700000e+03
 1.59100000e+03 2.20000000e+03 2.85900000e+03 3.79600000e+03
 2.25400000e+03 1.87500000e+03 3.02200000e+03 2.16600000e+03
 2.53100000e+03 2.00000000e+03 2.11800000e+03 2.90000000e+03
 1.82000000e+03 2.30200000e+03 9.97000000e+02 3.54100000e+03
 3.26300000e+03 1.03000000e+03 7.54000000e+02 2.33300000e+03
 1.39800000e+03 8.98000000e+03 1.92900000e+03 1.43000000e+03
 2.03400000e+03 4.48600000e+03 1.42500000e+03 8.30000000e+02
 1.04100000e+03 1.28000000e+03 3.33300000e+03 1.76900000e+03
 7.36000000e+02 1.96400000e+03 1.61900000e+03 1.45100000e+03
 5.06300000e+03 2.13800000e+03 2.36500000e+03 2.81600000e+03
 2.50000000e+03 1.08300000e+03 1.25000000e+03 3.02100000e+03
 9.83000000e+02 1.80000000e+03 1.77500000e+03 2.38300000e+03
 1.71700000e+03 2.79100000e+03 2.05400000e+03 2.59800000e+03
 1.77900000e+03 5.00000000e+03 1.98300000e+03 1.30000000e+03
 4.41700000e+03 4.33300000e+03 1.84300000e+03 1.86800000e+03
 2.16700000e+03 2.20900000e+03 1.38700000e+03 1.56000000e+03
 1.85700000e+03 2.22300000e+03 1.84200000e+03 8.00000000e+02
 9.85799988e+02 1.66600000e+03 3.05300000e+03 2.41600000e+03
 2.92500000e+03 2.93400000e+03 1.75000000e+03 1.80300000e+03
 1.86300000e+03 2.40500000e+03 1.64000000e+03 2.13400000e+03

```

1.89000000e+02 2.98500000e+03 2.45100000e+03 1.79300000e+03
4.49000000e+03 2.33000000e+03 2.16800000e+03 5.05000000e+02
3.03300000e+03 1.98700000e+03 9.23000000e+02 4.99600000e+03
1.60000000e+03 2.41700000e+03 2.11500000e+03 1.62500000e+03
1.40000000e+03 4.84000000e+02 2.00000000e+04 2.40000000e+03
2.03300000e+03 2.77300000e+03 1.41700000e+03 1.71900000e+03
4.30000000e+03 1.61200008e+01 2.34000000e+03 1.85100000e+03
1.99300000e+03 1.21000000e+03 1.37600000e+03 1.71000000e+03
1.25500000e+03 1.45600000e+03 1.73300000e+03 2.46600000e+03
1.66400000e+03 2.91700000e+03 1.50000000e+03 1.87200000e+03
3.15000000e+03 2.43600000e+03 2.78500000e+03 2.15700000e+03
9.13000000e+02 1.70000000e+03 2.85700000e+03 3.68300000e+03
5.30200000e+03 1.48300000e+03 1.28700000e+03 2.03500000e+03
6.66600000e+03 3.42800000e+03 1.63200000e+03 1.74200000e+03
1.42400000e+03 1.30200000e+03 2.04200000e+03 5.36000000e+02
2.84500000e+03 2.52400000e+03 6.63000000e+02 1.78300000e+03
2.01600000e+03 4.26600000e+03 1.03200000e+03 2.42000000e+02
2.06400000e+03 4.61000000e+02 2.21000000e+03 2.73900000e+03
2.23200000e+03 3.38370000e+04 1.52200000e+03 3.41600000e+03
3.30000000e+03 1.95000000e+03]
LoanAmount : [128. 66. 120. 141. 95. 70. 109. 114. 17. 125. 100. 76. 13
3. 115.
104. 116. 112. 122. 110. 35. 74. 106. 144. 80. 47. 75. 134. 96.
88. 44. 97. 135. 99. 126. 136. 81. 113. 130. 111. 50. 131. 25.
137. 94. 139. 118. 85. 93. 102. 84. 129. 30. 98. 121. 63. 87.
101. 67. 73. 108. 58. 48. 90. 124. 55. 59. 127. 72. 60. 138.
42. 140. 123. 150. 78. 54. 89. 143. 105. 132. 56. 117. 71. 46.
103. 45. 65. 53. 62. 148. 149. 119. 107. 145. 26. 9. 36. 61.
146. 142. 40.]
Loan_Amount_Term : [360. 120. 240. nan 180. 60. 300. 480. 84. 12. 36.]
Credit_History : [ 1. nan 0.]
Property_Area : ['Rural' 'Urban' 'Semiurban']
Loan_Status : ['N' 'Y']

```

It is clear that quite a bit of cleaning needs to take place. All nan values must be replaced. To keep the data integrity, nan values will be replaced with their own unique numerical values.

2. Gender will be 0 for male, 1 for female, 2 for nan.
3. Married will be 1 for yes and 0 for no.
4. Dependent options will be 1,2, 3 for 3+, and 4 for nan.
5. Education will be 1 for graduate, 0 for non.
6. Self Employed will be 1 for yes, 2 for no, and 3 for nan.
7. Loan Term will be 0 for nan.
8. Credit History will be 2 for nan.
9. Property Area will be 0 for Rural, 1 for Semiurban, and 2 for Urban

The other columns are already in numerical format > 0, so no need to change.

```
In [85]: train_data = pd.DataFrame(columns={'Gender':[], 'Married':[], 'Dependents':[],
                                           'CoapplicantIncome':[], 'LoanAmount':[],
                                           })
```

```
gender = [0 if x == 'Male' else x for x in loan_data.Gender.tolist()]
gender = [1 if x == 'Female' else x for x in gender]
gender = [3 if np.isnan(x) else x for x in gender]
train_data['Gender'] = gender

married = [1 if x == 'Yes' else 0 for x in loan_data.Married.tolist()]
train_data['Married'] = married

dependents = [3 if x == '3+' else x for x in loan_data.Dependents.tolist()]
dependents = [int(x) if x in ['0', '1', '2', '3'] else 4 for x in dependents]
train_data['Dependents'] = dependents

education = [1 if x == 'Graduate' else 0 for x in loan_data.Education.tolist()]
train_data['Education'] = education

self_employed = [1 if x == 'Yes' else x for x in loan_data.Self_Employed.tolist()]
self_employed = [0 if x == 'No' else x for x in self_employed]
self_employed = [3 if np.isnan(x) else x for x in self_employed]
train_data['Self_Employed'] = self_employed

train_data['ApplicantIncome'] = loan_data.ApplicantIncome.tolist()
train_data['CoapplicantIncome'] = loan_data.CoapplicantIncome.tolist()
train_data['LoanAmount'] = loan_data.LoanAmount.tolist()
train_data['Loan_Amount_Term'] = [0 if np.isnan(x) else x for x in loan_data.Loan_Amount_Term.tolist()]
train_data['Credit_History'] = [2 if np.isnan(x) else x for x in loan_data.Credit_History.tolist()]

property_area = [0 if x == 'Rural' else x for x in loan_data.Property_Area.tolist()]
property_area = [1 if x == 'Semiurban' else x for x in property_area]
property_area = [3 if x == 'Urban' else x for x in property_area]
train_data['Property_Area'] = property_area
train_data.index = loan_data.Loan_ID

print(train_data)
```

	Gender	Married	Dependents	Education	Self_Employed	\
Loan_ID						
LP001003	0	1	1	1		0
LP001005	0	1	0	1		1
LP001006	0	1	0	0		0
LP001008	0	0	0	1		0
LP001013	0	1	0	0		0
...
LP002953	0	1	4	1		0
LP002974	0	1	0	1		0
LP002978	1	0	0	1		0
LP002979	0	1	4	1		0
LP002990	1	0	0	1		1

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
Loan_ID				
LP001003	4583	1508.0	128.0	360.0
LP001005	3000	0.0	66.0	360.0
LP001006	2583	2358.0	120.0	360.0
LP001008	6000	0.0	141.0	360.0
LP001013	2333	1516.0	95.0	360.0
...
LP002953	5703	0.0	128.0	360.0
LP002974	3232	1950.0	108.0	360.0
LP002978	2900	0.0	71.0	360.0
LP002979	4106	0.0	40.0	180.0
LP002990	4583	0.0	133.0	360.0

	Credit_History	Property_Area
Loan_ID		
LP001003	1.0	0
LP001005	1.0	3
LP001006	1.0	3
LP001008	1.0	3
LP001013	1.0	3
...
LP002953	1.0	3
LP002974	1.0	0
LP002978	1.0	0
LP002979	1.0	0
LP002990	0.0	1

[381 rows x 11 columns]

Train the NFM

For curiosity, before throwing any assumptions, let's see how an NFM performs at categorizing vs the loan status in the dataset. Can it blindly perform well based on the clean dataset as it is?

```
In [37]: from sklearn.decomposition import NMF
nmf = NMF(n_components = 2)
```

```
approval_df = nmf.fit_transform(train_data)
approval_df = pd.DataFrame(approval_df, index=loan_data.Loan_ID)

print(approval_df)
```

	0	1
Loan_ID		
LP001003	2.504061	1.121004
LP001005	0.003904	0.752453
LP001006	3.921491	0.609687
LP001008	0.000000	1.498427
LP001013	2.523722	0.561495
...
LP002953	0.000000	1.424546
LP002974	3.241872	0.777734
LP002978	0.004264	0.727629
LP002979	0.000000	1.023864
LP002990	0.000000	1.146262

[381 rows x 2 columns]

```
In [38]: train_pred = pd.DataFrame(columns={'Loan_ID':[], 'Loan_Status':[], 'Pred':[]})

for loan_id in loan_data.Loan_ID:
    loan_status = loan_data[loan_data['Loan_ID'] == loan_id]['Loan_Status'].
    pred_list = approval_df.loc[loan_id].tolist()

    train_pred.loc[len(train_pred)] = [
        loan_id,
        loan_status,
        pred_list.index(max(pred_list))
    ]

print(train_pred)
```

	Loan_ID	Loan_Status	Pred
0	LP001003	N	0
1	LP001005	Y	1
2	LP001006	Y	0
3	LP001008	Y	1
4	LP001013	Y	0
..
376	LP002953	Y	1
377	LP002974	Y	0
378	LP002978	Y	1
379	LP002979	Y	1
380	LP002990	N	1

[381 rows x 3 columns]

```
In [39]: x = [0, 1]

y_cats = {}

for status in loan_data.Loan_Status.unique():
    cat_df = train_pred[train_pred['Loan_Status'] == status].copy()
    y_cats[status] = []
```

```

for pred in x:
    y_cats[status].append(len(cat_df[cat_df['Pred'] == pred].index))

y_cats[status] = np.array(y_cats[status])

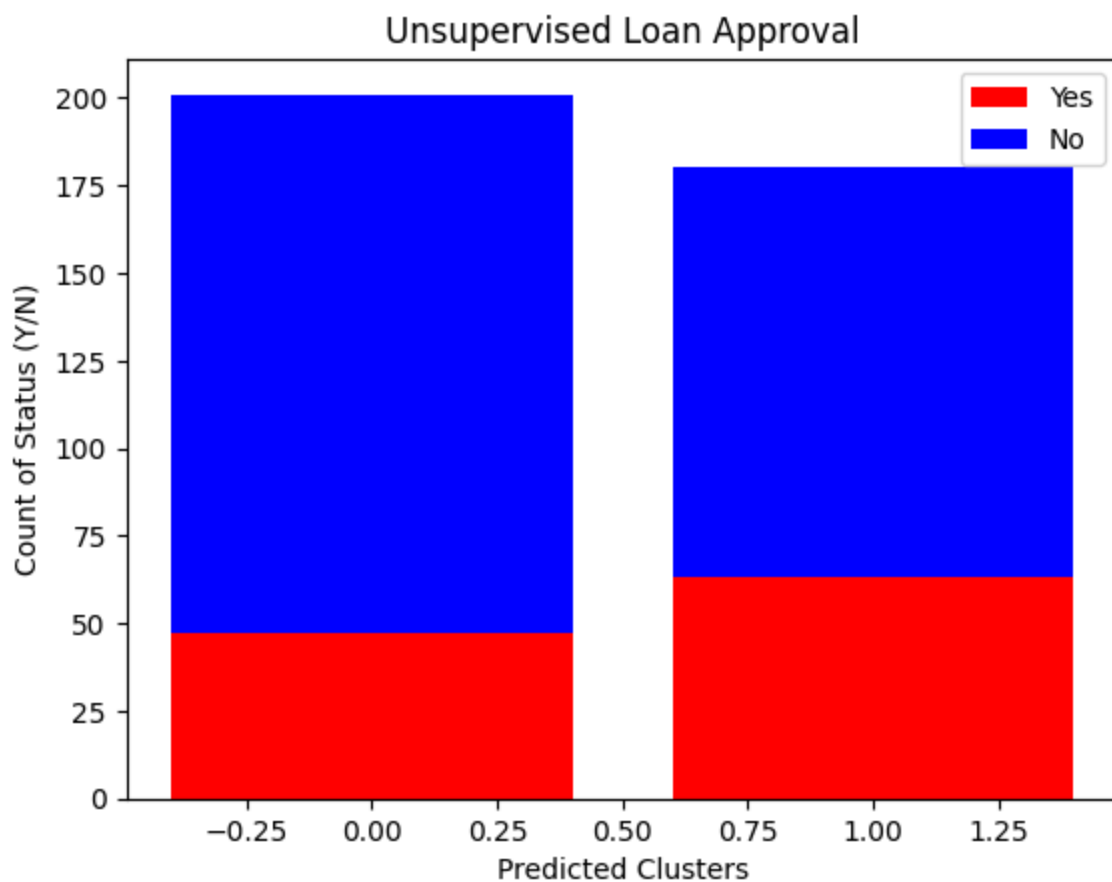
y_true = []
for idx in train_pred.index:
    status = train_pred['Loan_Status'][idx]
    y_true.append(
        list(y_cats[status]).index(max(y_cats[status]))
    )

plt.bar(x, y_cats['Y'], color = 'r')
plt.bar(x, y_cats['Y'], bottom = y_cats['N'], color='b')
plt.xlabel("Predicted Clusters")
plt.ylabel("Count of Status (Y/N)")
plt.legend(["Yes", "No"])
plt.title("Unsupervised Loan Approval")
plt.show()

from sklearn.metrics import accuracy_score

print(f'Accuracy = {accuracy_score(y_true, train_pred["Pred"]):0.2f}')

```



Accuracy = 0.57

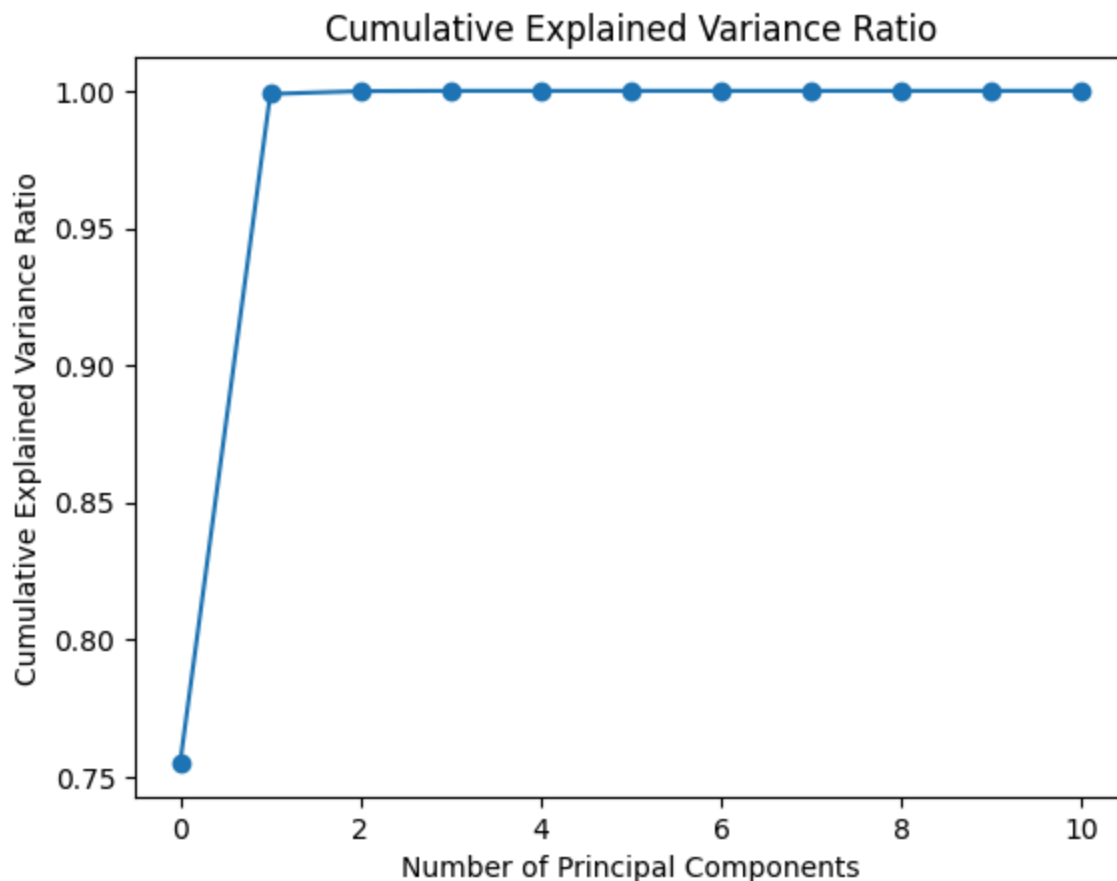
That answers the question... No, it performs quite poorly. What else can we learn from the data to improve the accuracy?

EDA 2 - Can we learn from PCA?

```
In [52]: from sklearn.decomposition import PCA

pca = PCA(svd_solver='full')
pca.fit(train_data)

cumulative_variance_ratio = np.cumsum(pca.explained_variance_ratio_)
plt.plot(cumulative_variance_ratio, marker='o')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.title('Cumulative Explained Variance Ratio')
plt.show()
```



Wow! Just 2 principle components account for the vast majority of the explained variance. We only have 11 components. It seems simple enough to loop through each pair and see which two components perform the best.

```
In [73]: results = {}
for col1 in train_data.columns:
    for col2 in train_data.columns:
        if col1 != col2 and col2 + ' ' + col1 not in results.keys():
            new_train_data = train_data[[col1, col2]].copy()
            nmf = NMF(n_components=2, max_iter=500)
            approval_df = nmf.fit_transform(new_train_data)
```



```

approval_df = pd.DataFrame(approval_df, index=loan_data.Loan_ID)

new_train_pred = pd.DataFrame(columns={'Loan_ID':[], 'Loan_Status':[]})

for loan_id in loan_data.Loan_ID:
    loan_status = loan_data[loan_data['Loan_ID'] == loan_id]['Loan_Status'].iloc[0]
    pred_list = approval_df.loc[loan_id].tolist()

    new_train_pred.loc[len(new_train_pred)] = [
        loan_id,
        loan_status,
        pred_list.index(max(pred_list))
    ]

status_df = new_train_pred[new_train_pred['Loan_Status'] == 'Y']
count0 = status_df['Pred'].tolist().count(0)
count1 = status_df['Pred'].tolist().count(1)
if count0 > count1:
    y_true = [0 if x == 'Y' else 1 for x in new_train_pred.Loan_Status]
else:
    y_true = [1 if x == 'Y' else 0 for x in new_train_pred.Loan_Status]

results[col1 + ' ' + col2] = accuracy_score(y_true, new_train_pred.Pred)

print(f'{col1 + " " + col2} : {results[col1 + " " + col2]:0.2f}')

```

Gender Married : 0.63
Gender Dependents : 0.64
Gender Education : 0.63
Gender Self_Employed : 0.65
Gender ApplicantIncome : 0.64
Gender CoapplicantIncome : 0.58
Gender LoanAmount : 0.63
Gender Loan_Amount_Term : 0.64
Gender Credit_History : 0.66
Gender Property_Area : 0.65
Married Dependents : 0.52
Married Education : 0.65
Married Self_Employed : 0.56
Married ApplicantIncome : 0.57
Married CoapplicantIncome : 0.57
Married LoanAmount : 0.57
Married Loan_Amount_Term : 0.58
Married Credit_History : 0.54
Married Property_Area : 0.52
Dependents Education : 0.54
Dependents Self_Employed : 0.66
Dependents ApplicantIncome : 0.57
Dependents CoapplicantIncome : 0.50
Dependents LoanAmount : 0.57
Dependents Loan_Amount_Term : 0.56
Dependents Credit_History : 0.71
Dependents Property_Area : 0.61
Education Self_Employed : 0.64
Education ApplicantIncome : 0.58
Education CoapplicantIncome : 0.52
Education LoanAmount : 0.61
Education Loan_Amount_Term : 0.62
Education Credit_History : 0.54
Education Property_Area : 0.62
Self_Employed ApplicantIncome : 0.65
Self_Employed CoapplicantIncome : 0.57
Self_Employed LoanAmount : 0.64
Self_Employed Loan_Amount_Term : 0.66
Self_Employed Credit_History : 0.65
Self_Employed Property_Area : 0.65
ApplicantIncome CoapplicantIncome : 0.47
ApplicantIncome LoanAmount : 0.55
ApplicantIncome Loan_Amount_Term : 0.62
ApplicantIncome Credit_History : 0.75
ApplicantIncome Property_Area : 0.49
CoapplicantIncome LoanAmount : 0.71
CoapplicantIncome Loan_Amount_Term : 0.65
CoapplicantIncome Credit_History : 0.58
CoapplicantIncome Property_Area : 0.53
LoanAmount Loan_Amount_Term : 0.49
LoanAmount Credit_History : 0.66
LoanAmount Property_Area : 0.55
Loan_Amount_Term Credit_History : 0.81
Loan_Amount_Term Property_Area : 0.58
Credit_History Property_Area : 0.70

```
In [74]: print(max(results, key=results.get))
```

Loan_Amount_Term Credit_History

Now we see that "Loan_Amount_Term" and "Credit_History" perform the best in terms of accuracy. Let's plot the results.

```
In [77]: new_train_data = train_data[['Loan_Amount_Term', 'Credit_History']].copy()

nmf = NMF(n_components=2, max_iter=500)
approval_df = nmf.fit_transform(new_train_data)
approval_df = pd.DataFrame(approval_df, index=loan_data.Loan_ID)

new_train_pred = pd.DataFrame(columns={'Loan_ID': [], 'Loan_Status': [], 'Pred': []})

for loan_id in loan_data.Loan_ID:
    loan_status = loan_data[loan_data['Loan_ID'] == loan_id]['Loan_Status'].values
    pred_list = approval_df.loc[loan_id].tolist()

    new_train_pred.loc[len(new_train_pred)] = [
        loan_id,
        loan_status,
        pred_list.index(max(pred_list))
    ]

x = [0, 1]

y_cats = {}

for status in loan_data.Loan_Status.unique():
    cat_df = new_train_pred[new_train_pred['Loan_Status'] == status].copy()
    y_cats[status] = []
    for pred in x:
        y_cats[status].append(len(cat_df[cat_df['Pred'] == pred]).index())

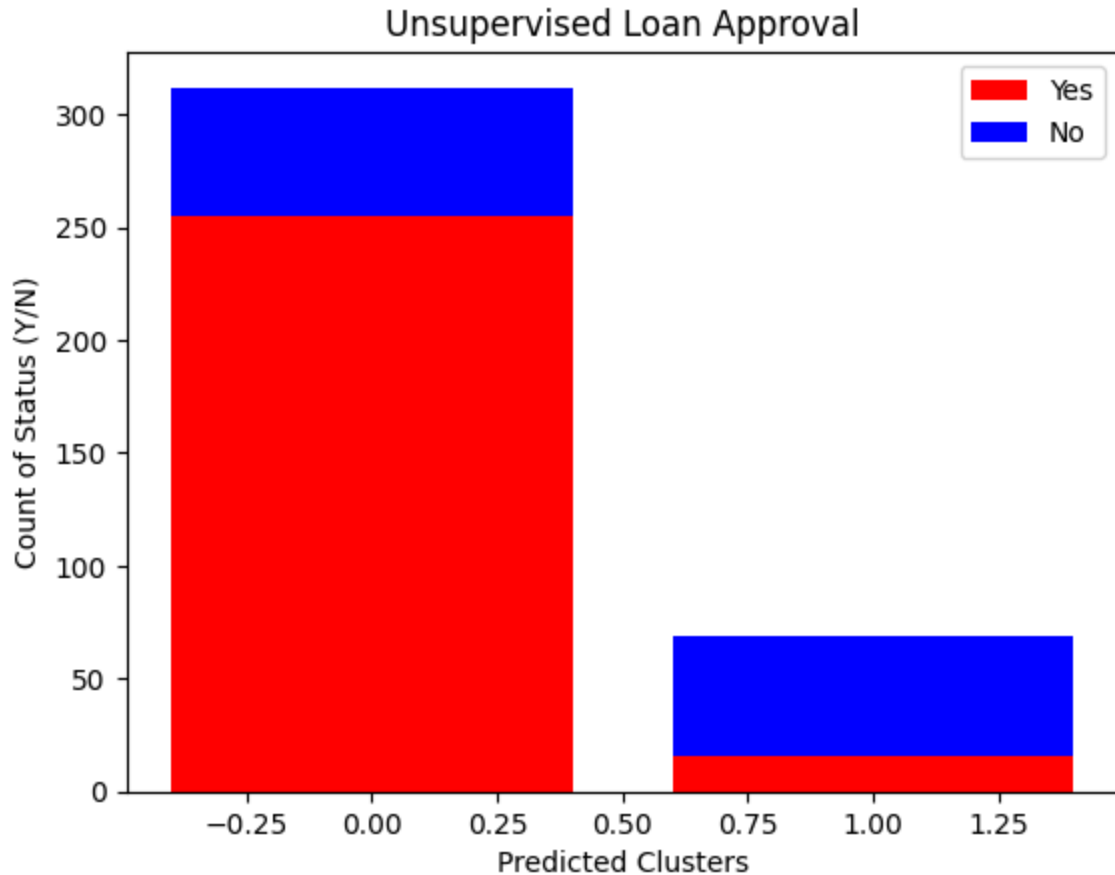
    y_cats[status] = np.array(y_cats[status])

status_df = new_train_pred[new_train_pred['Loan_Status'] == 'Y'].copy()
count0 = status_df['Pred'].tolist().count(0)
count1 = status_df['Pred'].tolist().count(1)
if count0 > count1:
    y_true = [0 if x == 'Y' else 1 for x in new_train_pred.Loan_Status.tolist()]
else:
    y_true = [1 if x == 'Y' else 0 for x in new_train_pred.Loan_Status.tolist()]

plt.bar(x, y_cats['Y'], color = 'r')
plt.bar(x, y_cats['N'], bottom = y_cats['Y'], color='b')
plt.xlabel("Predicted Clusters")
plt.ylabel("Count of Status (Y/N)")
plt.legend(["Yes", "No"])
plt.title("Unsupervised Loan Approval")
plt.show()

from sklearn.metrics import accuracy_score
```

```
print(f'Accuracy = {accuracy_score(y_true, new_train_pred["Pred"]):0.2f}')
```



Accuracy = 0.81

While the accuracy is quite a bit higher, the factorization definitely favors a "Yes" for loan approval.

Recommender System

While the unsupervised method may not be the best at deciding loan approval, it may give a suggestion for loan amount that could increase the odds of being approved. We will begin by building a loan approved dataset and loan unapproved dataset. Then, we will go through each unapproved option and see, based on similarity of an approved one, what the recommended loan amount would be.

```
In [101]: from scipy.spatial.distance import cosine

approved_df = train_data.copy()
approved_df.drop(approved_df.index, inplace=True)
declined_df = train_data.copy()
declined_df.drop(declined_df.index, inplace=True)

loan_id_declined = []
for loan_id in train_data.index:
    if loan_data[loan_data['Loan_ID'] == loan_id]['Loan_Status'].values[0] ==
```

```

        approved_df.loc[len(approved_df)] = train_data.loc[loan_id].tolist()
    else:
        declined_df.loc[len(declined_df)] = train_data.loc[loan_id].tolist()
        loan_id_declined.append(loan_id)

declined_df['Loan_ID'] = loan_id_declined
suggestions_df = pd.DataFrame(columns = {'Loan_ID':[], 'Original_Amount':[], '

for idx1 in declined_df.index:
    print(f'LoanID: {declined_df.Loan_ID[idx1]}, original ask: {declined_df.
    max_sim = [0,0]
    for idx2 in approved_df.index:
        sim = cosine(declined_df.loc[idx1].tolist()[1:], approved_df.loc[idx
        if sim > max_sim[1] and declined_df.LoanAmount[idx1] > approved_df.L
            max_sim = [idx2, sim]

    print(f'Suggested loan amount: {approved_df.LoanAmount[max_sim[0]]}')
    suggestions_df.loc[len(suggestions_df)] = [
        declined_df.Loan_ID[idx1], declined_df.LoanAmount[idx1], approved_df
]

```

LoanID: LP001003, original ask: 128.0
 Suggested loan amount: 66.0
 LoanID: LP001029, original ask: 114.0
 Suggested loan amount: 66.0
 LoanID: LP001036, original ask: 76.0
 Suggested loan amount: 75.0
 LoanID: LP001038, original ask: 133.0
 Suggested loan amount: 66.0
 LoanID: LP001043, original ask: 104.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

Suggested loan amount: 66.0
 LoanID: LP001047, original ask: 116.0
 Suggested loan amount: 66.0
 LoanID: LP001050, original ask: 112.0
 Suggested loan amount: 96.0
 LoanID: LP001086, original ask: 35.0
 Suggested loan amount: 66.0
 LoanID: LP001095, original ask: 74.0
 Suggested loan amount: 66.0
 LoanID: LP001097, original ask: 106.0
 Suggested loan amount: 66.0
 LoanID: LP001109, original ask: 100.0

```

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP001119, original ask: 80.0
Suggested loan amount: 66.0
LoanID: LP001146, original ask: 120.0
Suggested loan amount: 96.0
LoanID: LP001164, original ask: 112.0
Suggested loan amount: 96.0
LoanID: LP001179, original ask: 134.0
Suggested loan amount: 66.0
LoanID: LP001197, original ask: 135.0
Suggested loan amount: 66.0
LoanID: LP001222, original ask: 116.0
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)

```

Suggested loan amount: 96.0
LoanID: LP001228, original ask: 126.0
Suggested loan amount: 66.0
LoanID: LP001241, original ask: 136.0
Suggested loan amount: 96.0
LoanID: LP001250, original ask: 95.0
Suggested loan amount: 66.0
LoanID: LP001255, original ask: 113.0
Suggested loan amount: 66.0
LoanID: LP001259, original ask: 110.0
Suggested loan amount: 66.0
LoanID: LP001370, original ask: 120.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

Suggested loan amount: 66.0
LoanID: LP001489, original ask: 84.0
Suggested loan amount: 75.0
LoanID: LP001493, original ask: 129.0
Suggested loan amount: 66.0
LoanID: LP001528, original ask: 118.0
Suggested loan amount: 66.0
LoanID: LP001532, original ask: 113.0
Suggested loan amount: 66.0
LoanID: LP001565, original ask: 121.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP001577, original ask: 112.0
Suggested loan amount: 96.0
LoanID: LP001579, original ask: 63.0
Suggested loan amount: 66.0
LoanID: LP001586, original ask: 81.0
Suggested loan amount: 66.0
LoanID: LP001603, original ask: 87.0
Suggested loan amount: 66.0
LoanID: LP001630, original ask: 102.0
Suggested loan amount: 66.0
LoanID: LP001634, original ask: 67.0
```



```

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP001641, original ask: 66.0
Suggested loan amount: 66.0
LoanID: LP001702, original ask: 127.0
Suggested loan amount: 66.0
LoanID: LP001711, original ask: 128.0
Suggested loan amount: 66.0
LoanID: LP001722, original ask: 135.0
Suggested loan amount: 66.0
LoanID: LP001732, original ask: 72.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)

```

Suggested loan amount: 66.0
LoanID: LP001736, original ask: 60.0
Suggested loan amount: 66.0
LoanID: LP001754, original ask: 138.0
Suggested loan amount: 66.0
LoanID: LP001800, original ask: 140.0
Suggested loan amount: 66.0
LoanID: LP001813, original ask: 120.0
Suggested loan amount: 66.0
LoanID: LP001836, original ask: 138.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

Suggested loan amount: 96.0
LoanID: LP001849, original ask: 115.0
Suggested loan amount: 66.0
LoanID: LP001854, original ask: 94.0
Suggested loan amount: 66.0
LoanID: LP001864, original ask: 128.0
Suggested loan amount: 66.0
LoanID: LP001883, original ask: 135.0
Suggested loan amount: 96.0
LoanID: LP001925, original ask: 99.0
Suggested loan amount: 96.0
LoanID: LP001938, original ask: 127.0
Suggested loan amount: 66.0
LoanID: LP001945, original ask: 143.0
Suggested loan amount: 96.0
LoanID: LP001963, original ask: 113.0
Suggested loan amount: 66.0
LoanID: LP001964, original ask: 93.0
Suggested loan amount: 66.0
LoanID: LP001994, original ask: 104.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 96.0
LoanID: LP002086, original ask: 110.0
Suggested loan amount: 96.0
LoanID: LP002116, original ask: 46.0
Suggested loan amount: 25.0
LoanID: LP002142, original ask: 105.0
Suggested loan amount: 96.0
LoanID: LP002151, original ask: 67.0
Suggested loan amount: 66.0
LoanID: LP002158, original ask: 100.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP002161, original ask: 81.0
Suggested loan amount: 75.0
LoanID: LP002181, original ask: 133.0
Suggested loan amount: 66.0
LoanID: LP002187, original ask: 96.0
Suggested loan amount: 66.0
LoanID: LP002188, original ask: 124.0
Suggested loan amount: 66.0
LoanID: LP002205, original ask: 111.0
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP002224, original ask: 71.0
Suggested loan amount: 66.0
LoanID: LP002236, original ask: 100.0
Suggested loan amount: 66.0
LoanID: LP002277, original ask: 71.0
Suggested loan amount: 25.0
LoanID: LP002287, original ask: 103.0
Suggested loan amount: 96.0
LoanID: LP002288, original ask: 45.0
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

Suggested loan amount: 66.0
LoanID: LP002296, original ask: 65.0
Suggested loan amount: 66.0
LoanID: LP002318, original ask: 62.0
Suggested loan amount: 25.0
LoanID: LP002357, original ask: 80.0
Suggested loan amount: 75.0
LoanID: LP002362, original ask: 110.0
Suggested loan amount: 66.0
LoanID: LP002367, original ask: 81.0
Suggested loan amount: 75.0
LoanID: LP002379, original ask: 105.0
Suggested loan amount: 66.0
LoanID: LP002435, original ask: 55.0
Suggested loan amount: 66.0
LoanID: LP002443, original ask: 150.0
Suggested loan amount: 66.0
LoanID: LP002446, original ask: 125.0
Suggested loan amount: 66.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

LoanID: LP002448, original ask: 149.0
Suggested loan amount: 66.0
LoanID: LP002493, original ask: 98.0
Suggested loan amount: 66.0
LoanID: LP002500, original ask: 70.0
Suggested loan amount: 66.0
LoanID: LP002505, original ask: 110.0
Suggested loan amount: 66.0
LoanID: LP002517, original ask: 113.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
Suggested loan amount: 66.0
LoanID: LP002530, original ask: 132.0
Suggested loan amount: 96.0
LoanID: LP002545, original ask: 80.0
Suggested loan amount: 66.0
LoanID: LP002556, original ask: 75.0
Suggested loan amount: 66.0
LoanID: LP002585, original ask: 119.0
Suggested loan amount: 66.0
LoanID: LP002618, original ask: 138.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

Suggested loan amount: 66.0
 LoanID: LP002625, original ask: 96.0
 Suggested loan amount: 88.0
 LoanID: LP002637, original ask: 100.0
 Suggested loan amount: 66.0
 LoanID: LP002648, original ask: 70.0
 Suggested loan amount: 66.0
 LoanID: LP002682, original ask: 123.0
 Suggested loan amount: 66.0
 LoanID: LP002684, original ask: 95.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

Suggested loan amount: 88.0
 LoanID: LP002723, original ask: 110.0
 Suggested loan amount: 66.0
 LoanID: LP002739, original ask: 66.0
 Suggested loan amount: 66.0
 LoanID: LP002743, original ask: 99.0
 Suggested loan amount: 96.0
 LoanID: LP002768, original ask: 80.0
 Suggested loan amount: 66.0
 LoanID: LP002776, original ask: 103.0

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide

dist = 1.0 - uv / np.sqrt(uu * vv)

Suggested loan amount: 96.0
 LoanID: LP002789, original ask: 132.0
 Suggested loan amount: 66.0
 LoanID: LP002837, original ask: 123.0
 Suggested loan amount: 66.0
 LoanID: LP002840, original ask: 9.0
 Suggested loan amount: 66.0
 LoanID: LP002841, original ask: 104.0
 Suggested loan amount: 66.0
 LoanID: LP002863, original ask: 150.0
 Suggested loan amount: 66.0
 LoanID: LP002872, original ask: 136.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

Suggested loan amount: 96.0
 LoanID: LP002893, original ask: 90.0
 Suggested loan amount: 66.0
 LoanID: LP002898, original ask: 61.0
 Suggested loan amount: 66.0
 LoanID: LP002911, original ask: 146.0
 Suggested loan amount: 66.0
 LoanID: LP002926, original ask: 106.0
 Suggested loan amount: 66.0
 LoanID: LP002943, original ask: 88.0

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```

```
/Users/dillonwilliams/PycharmProjects/BlackSheep/venv/lib/python3.11/site-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in scalar divide
```

```
    dist = 1.0 - uv / np.sqrt(uu * vv)
```


Suggested loan amount: 66.0
LoanID: LP002990, original ask: 133.0
Suggested loan amount: 96.0

In [102... `print(suggestions_df)`

	Loan_ID	Original_Amount	Suggestion
0	LP001003	128.0	66.0
1	LP001029	114.0	66.0
2	LP001036	76.0	75.0
3	LP001038	133.0	66.0
4	LP001043	104.0	66.0
..
105	LP002898	61.0	66.0
106	LP002911	146.0	66.0
107	LP002926	106.0	66.0
108	LP002943	88.0	66.0
109	LP002990	133.0	96.0

[110 rows x 3 columns]

Conclusion/Discussion

The unsupervised approach did not perform as well as the supervised approach in the previous class. Doing PCA did help get the accuracy up, but it was still a poor performing decision-maker. It would be interesting to see if the recommender system does increase loan approval rates, but that cannot be tested on this dataset.

All in all, it seems unsupervised approaches are best left to exploratory analysis or recommendations, and not decision-making.