# STAT207 Final Project - Predicting Loan Approval

Haotian Kang, Rongsheng Zhang, Tangyue Gong, Aditya Agarwal

```
#Imports here
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import KFold
from sklearn.metrics import roc_curve
from sklearn.cluster import KMeans
```

## 1. Introduction

In this project, we are trying to predict the likelihood of a person gaining approval for a loan based on a variety of chracteristics. This data is useful for financial institutions who lend consumers money like banks, credit card companies, BNPL apps, etc.

I think for this type of research we would need to prioritize minimizing false positive outcomes. This is as we don't want to lend money to people who would be unable to repay. Overstating loan approval probability can have negative consequences as consumers don't tend to behave rationally with debt and are often unable to pay back loans they take.

We saw this in the 2008 housing crisis, where people took on a lot of debt and invested in real estate even if they were not financially able to pay it back (1).

We also see these negative effects today, with student loan defaults and delinquincies seeing a sharp rise. (2).

**Research Goal Statement**

Primary Research Goal - To build a predictive model that will effectively predict loan approval status for new datasets.

Secondary Research Goal - To build a model that can yield interpretative insights about the nature of the relationship between the variables in our dataset

## 2. Dataset Discussion

The dataset used in this project was obtained from kaggle on the 23rd of April at the following link: https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data?resource=download.

The rows in this dataset represent loan applications, their details, and whether or not they were approved. This data was collected by synthesizing existing Credit Risk and Financial Risk for Loan Approval datatsets on kaggle. This dataset is not inclusive of all the different varieties of loan applicants. One example of a major category omitted is auto loan applicants. Since the number of loan applicants in the US is so vast and diverse, being inclusive of all different types of applicants is not possible. Based on the answer to our research question, the person in our research motivation may try to imporve or change certain parts of their application before applying for a loan. They also may be encourage or discouraged from taking on debt in general based on their profile and their ability to change different aspects of their profile.

Variables

- Response Variable: "loan_status" indicator variable. 1: loan approved, 0: loan rejected
- Explanatory Variables
  - "person_age" - Applicants age. Numerical Explanatory Variable
  - "person_gender" - Applicants gender. Categorical Explanatory Variable
  - "person_income" - Applicants yearly income. Numerical Explanatory Variable
  - "loan_amnt" - Dollar amount of loan requested. Numerical Explanatory Variable
  - "loan_intent" - Reason for loan. Categorical Explanatory Variable
  - "previous_loan_defaults_on_file" - Wether or not the applicant has previously defaulted on a loan. Categorical Explanatory Variable
  - "loan_int_rate" - Loan interest rate. Numerical Explanatory Variable

These 7 explanatory variables were selected as they are expected to provide significant predictive power to the model while not providing issues with multicollinearity as they are not strongly related to each other.

```
df = pd.read_csv("loan_data.csv")
df
```

```
       person_age person_gender person_education  person_income  \
0            22.0        female           Master        71948.0
1            21.0        female      High School        12282.0
2            25.0        female      High School        12438.0
3            23.0        female         Bachelor        79753.0
4            24.0          male           Master        66135.0
```

```
...          ...          ...              ...          ...
44995       27.0         male        Associate      47971.0
44996       37.0       female        Associate      65800.0
44997       33.0         male        Associate      56942.0
44998       29.0         male         Bachelor      33164.0
44999       24.0         male      High School      51609.0

       person_emp_exp person_home_ownership  loan_amnt        loan_intent  \
0                   0                  RENT    35000.0           PERSONAL
1                   0                   OWN     1000.0          EDUCATION
2                   3              MORTGAGE     5500.0            MEDICAL
3                   0                  RENT    35000.0            MEDICAL
4                   1                  RENT    35000.0            MEDICAL
...               ...                   ...        ...                ...
44995               6                  RENT    15000.0            MEDICAL
44996              17                  RENT     9000.0    HOMEIMPROVEMENT
44997               7                  RENT     2771.0  DEBTCONSOLIDATION
44998               4                  RENT    12000.0          EDUCATION
44999               1                  RENT     6665.0  DEBTCONSOLIDATION

       loan_int_rate  loan_percent_income  cb_person_cred_hist_length  \
0              16.02                 0.49                         3.0
1              11.14                 0.08                         2.0
2              12.87                 0.44                         3.0
3              15.23                 0.44                         2.0
4              14.27                 0.53                         4.0
...              ...                  ...                         ...
44995          15.66                 0.31                         3.0
44996          14.07                 0.14                        11.0
44997          10.02                 0.05                        10.0
44998          13.23                 0.36                         6.0
44999          17.05                 0.13                         3.0

       credit_score previous_loan_defaults_on_file  loan_status
0               561                           No            1
1               504                          Yes            0
2               635                           No            1
3               675                           No            1
4               586                           No            1
...             ...                          ...          ...
44995           645                           No            1
44996           621                           No            1
44997           668                           No            1
44998           604                           No            1
44999           628                           No            1
```

```
[45000 rows x 14 columns]
```

```
ognumrows = df.shape[0]
ognumrows
```

```
45000
```

## 3. Dataset Cleaning

Looking at our dataset, we don't observe any implicit missing values. This can be verified by observing the datatypes of the numerical variables, which are all int or float. Similarly looking at the unique values of the categorical variables we are considering, we don't observe any values like "none" or blank spaces, indicating we don't have any implicit missing values.

Looking at our categorical explanatory variables, we can see in all the variables we are considering for our project that each level for these variables has a significant amount of observatins, indicating none of them should be dropped from our dataset.

Considering the scatterplot of our numerical explanatory variables, we notice outliers spcifically for observations with extremely high ages ($> 100$). The pro of dropping these observations with extremely high ages is that it will make our model better at predicting loan approval for the overwhelming majority of applicants. The con is that the model will be worse at predicting loan aprovale for applicants older than 100 years old. Considering this age group is extremely rarely seen among loan applicants it is beneficial for our analysis to drop these outliers. Dropping these outliers dropped 7 rows from our dataset.

```
df.dtypes
```

```
person_age                      float64
person_gender                    object
person_education                 object
person_income                   float64
person_emp_exp                    int64
person_home_ownership            object
loan_amnt                       float64
loan_intent                      object
loan_int_rate                   float64
loan_percent_income             float64
cb_person_cred_hist_length      float64
credit_score                      int64
previous_loan_defaults_on_file   object
loan_status                       int64
dtype: object
```

```
print(df["person_gender"].unique())
print(df["loan_intent"].unique())
```

```
print(df["previous_loan_defaults_on_file"].unique())
```

```
['female' 'male']
['PERSONAL' 'EDUCATION' 'MEDICAL' 'VENTURE' 'HOMEIMPROVEMENT'
 'DEBTCONSOLIDATION']
['No' 'Yes']
```

```
df["person_gender"].value_counts()
```

```
person_gender
male      24841
female    20159
Name: count, dtype: int64
```

```
df["loan_intent"].value_counts()
```

```
loan_intent
EDUCATION            9153
MEDICAL              8548
VENTURE              7819
PERSONAL             7552
DEBTCONSOLIDATION    7145
HOMEIMPROVEMENT      4783
Name: count, dtype: int64
```

```
df["previous_loan_defaults_on_file"].value_counts()
```

```
previous_loan_defaults_on_file
Yes    22858
No     22142
Name: count, dtype: int64
```

```
df["person_age"].max()
```

```
144.0
```

```
df = df[df["person_age"] <= 100]
numrows1 = df.shape[0]
ognumrows - numrows1
```

```
7
```

```
df.head()
```

```
   person_age person_gender person_education  person_income  person_emp_exp  \
0        22.0        female           Master        71948.0               0
1        21.0        female      High School        12282.0               0
2        25.0        female      High School        12438.0               3
3        23.0        female         Bachelor        79753.0               0
4        24.0          male           Master        66135.0               1

  person_home_ownership  loan_amnt loan_intent  loan_int_rate  \
```

```
0                  RENT    35000.0    PERSONAL          16.02
1                   OWN     1000.0   EDUCATION          11.14
2              MORTGAGE     5500.0     MEDICAL          12.87
3                  RENT    35000.0     MEDICAL          15.23
4                  RENT    35000.0     MEDICAL          14.27

   loan_percent_income  cb_person_cred_hist_length  credit_score  \
0                 0.49                         3.0           561
1                 0.08                         2.0           504
2                 0.44                         3.0           635
3                 0.44                         2.0           675
4                 0.53                         4.0           586

  previous_loan_defaults_on_file  loan_status
0                             No            1
1                            Yes            0
2                             No            1
3                             No            1
4                             No            1
```

```
numvars = ['person_age', 'person_income', 'loan_amnt', 'loan_int_rate']
catvars = ['person_gender', 'previous_loan_defaults_on_file', 'loan_intent']
features = numvars + catvars
response = 'loan_status'
```

## 4. Preliminary Analysis

In this section, we will explore the relationships between our response variable, loan approval status, and the explanatory variables. We will investigate both the individual relationships between the response and each explanatory variable, as well as the relationships between pairs of explanatory variables. Additionally, we will examine potential interaction effects between numerical and categorical explanatory variables to gain a comprehensive understanding of the factors influencing loan approval.

### Relationships between the Response Variable and the Explanatory Variables

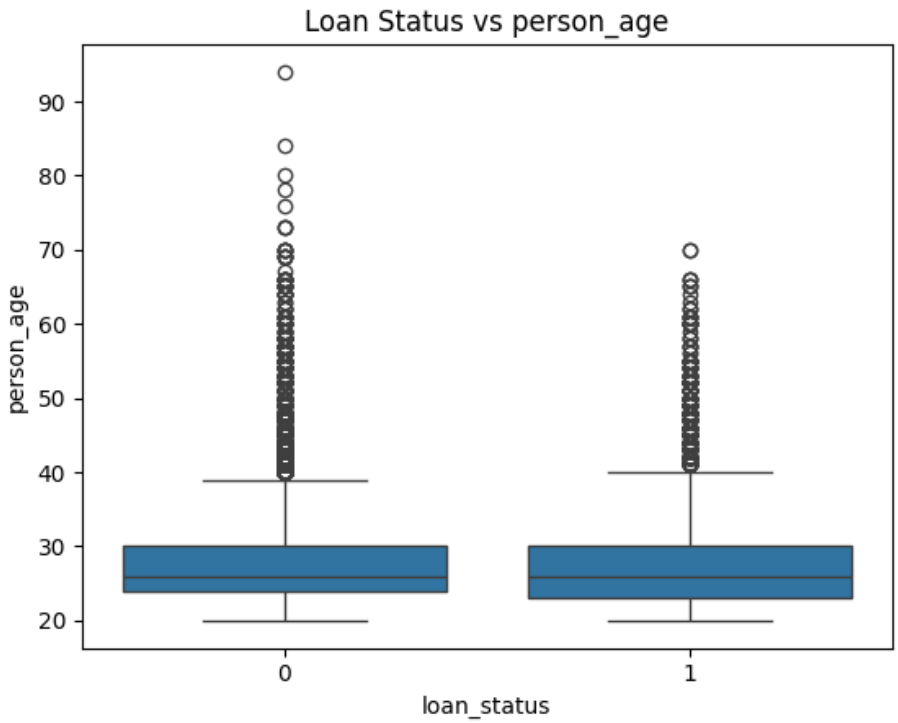
To visualize the relationships between the response variable (loan_status) and each explanatory variable, we will use appropriate plots such as box plots for numerical variables and bar plots for categorical variables. These visualizations will help us identify which explanatory variables exhibit strong or weak relationships with loan approval.

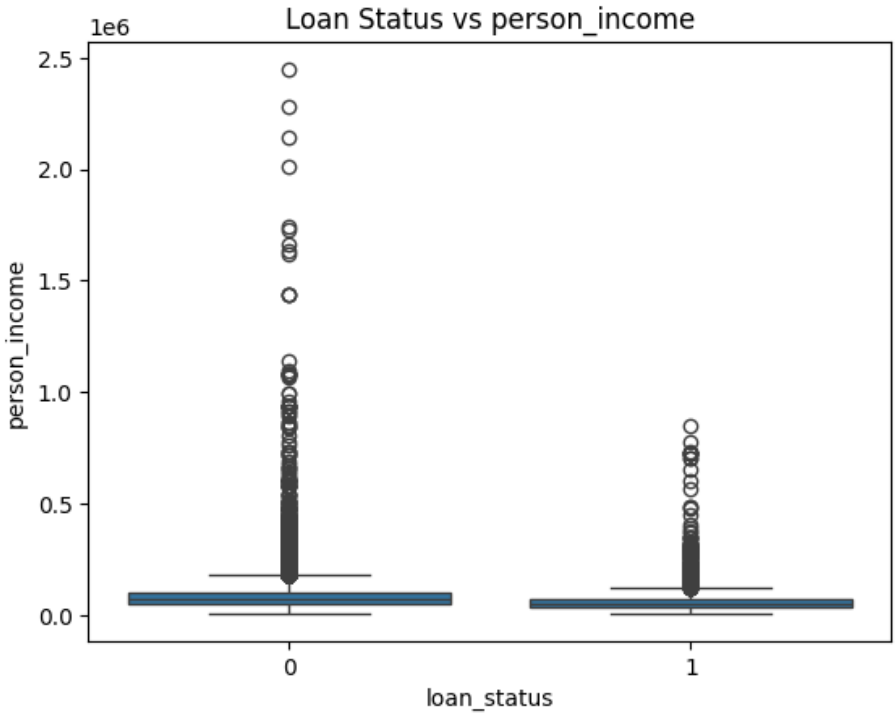
```
df.head()
```

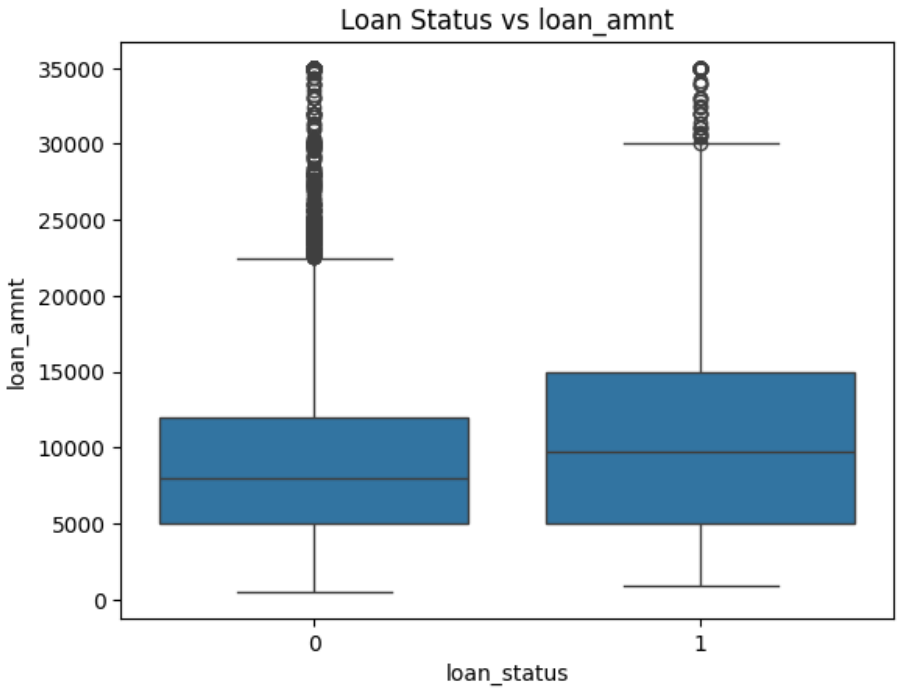
```
   person_age person_gender person_education  person_income  person_emp_exp  \
0        22.0        female           Master        71948.0               0
```

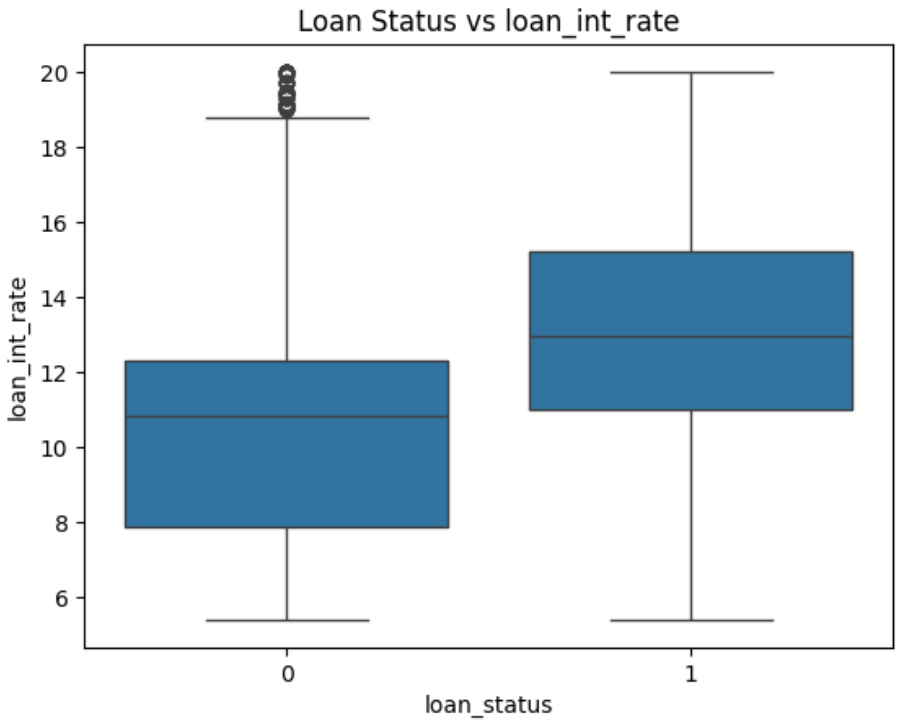
```
1        21.0        female      High School        12282.0                   0
2        25.0        female      High School        12438.0                   3
3        23.0        female         Bachelor        79753.0                   0
4        24.0          male           Master        66135.0                   1

  person_home_ownership  loan_amnt loan_intent  loan_int_rate  \
0                  RENT    35000.0    PERSONAL          16.02
1                   OWN     1000.0   EDUCATION          11.14
2              MORTGAGE     5500.0     MEDICAL          12.87
3                  RENT    35000.0     MEDICAL          15.23
4                  RENT    35000.0     MEDICAL          14.27

   loan_percent_income  cb_person_cred_hist_length  credit_score  \
0                 0.49                         3.0           561
1                 0.08                         2.0           504
2                 0.44                         3.0           635
3                 0.44                         2.0           675
4                 0.53                         4.0           586

  previous_loan_defaults_on_file  loan_status
0                             No            1
1                            Yes            0
2                             No            1
3                             No            1
4                             No            1
```

```python
for var in numvars:
  sns.boxplot(x='loan_status', y=var, data=df)
  plt.title(f"Loan Status vs {var}")
  plt.show()
```


Loan Status vs person_age
person_age
90
80
70
60
50
40
30
20
0
1
loan_status


Loan Status vs person_income
1e6
2.5
2.0
1.5
1.0
0.5
0.0
person_income
0
1
loan_status


Loan Status vs loan_amnt
35000
30000
25000
20000
15000
10000
5000
0
loan_amnt
0
1
loan_status




Side-by-side boxplots were created for numerical explanatory variables (person_age, person_income, loan_amnt, loan_int_rate) against loan_status.

**Person Age:** The distribution of person_age was quite similar across both approved and rejected loans, with slightly younger applicants being more common in both groups. This suggests that person_age has only a **weak association** with loan approval.

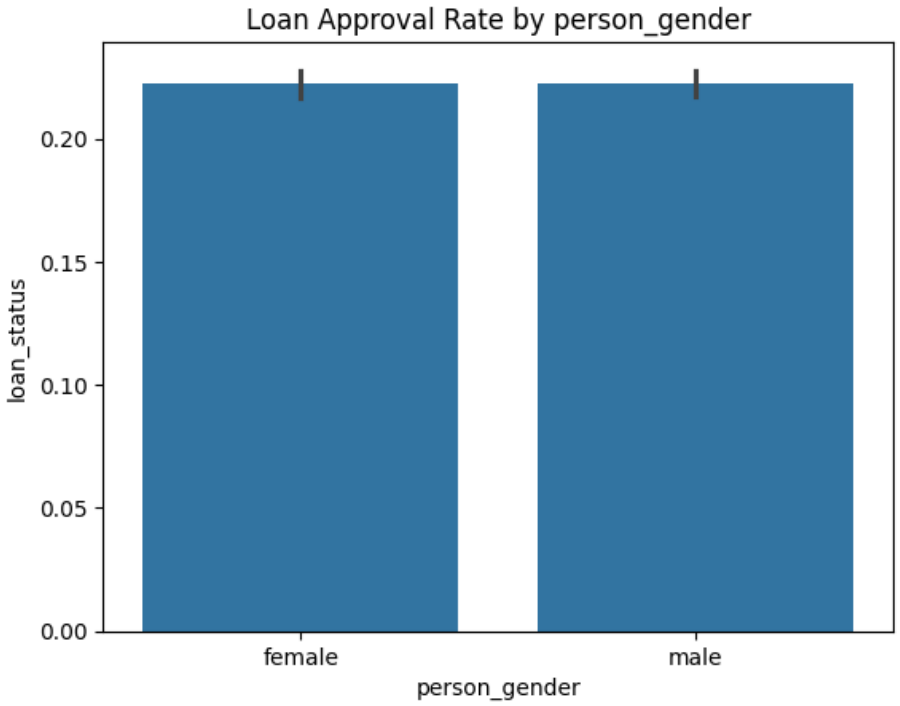
**Person Income:** Although there are many extreme income outliers, overall, the distributions of person_income for approved and rejected applicants were quite similar. Thus, person_income also appears to have a **weak association** with loan approval.

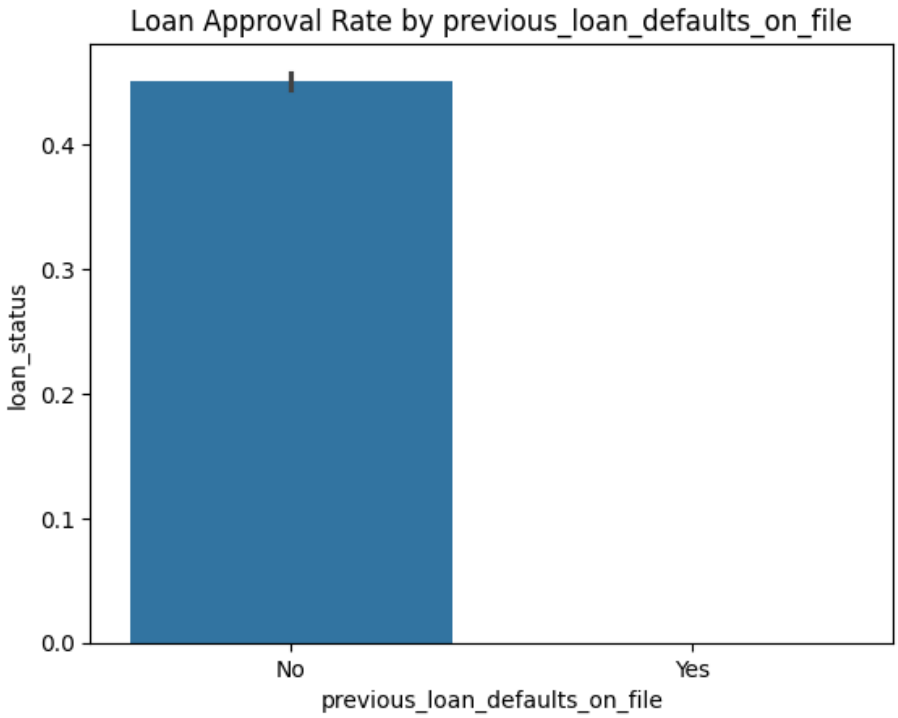
**Loan Amount:** The boxplots show that applicants requesting higher loan_amnt amounts were more likely to be approved. The median loan amount was noticeably higher among approved applicants, which indicates that loan_amnt has a **moderate positive** relationship with loan approval.

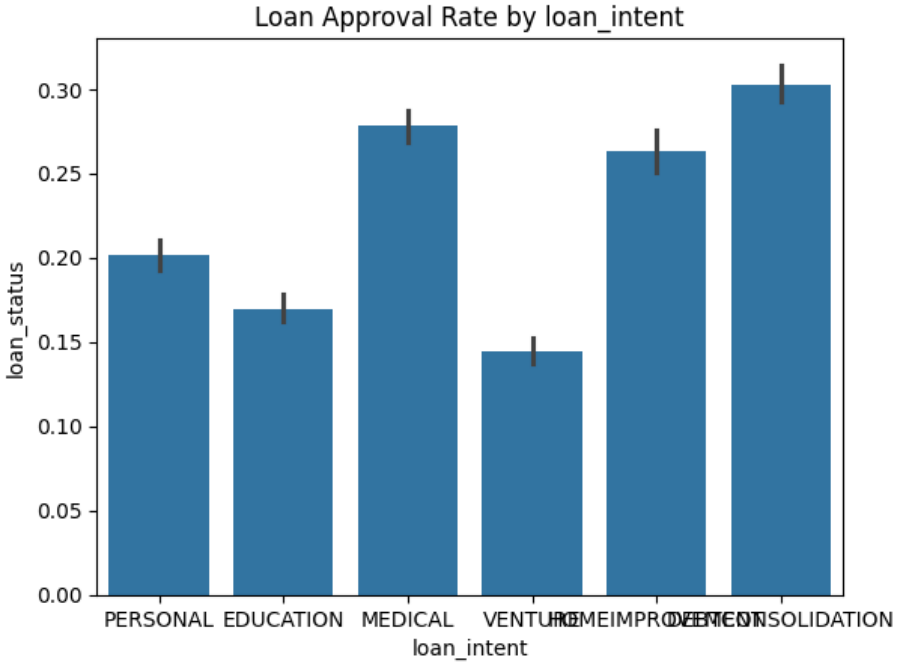
**Loan Interest Rate:** The distribution of loan interest rate was quite different across both approved and rejected loans. This suggests that loan interest rate has a **strong association** with loan approval.

```
for var in catvars:
  sns.barplot(x=var, y='loan_status', data=df)
```

```
plt.title(f"Loan Approval Rate by {var}")
plt.show()
```




Loan Approval Rate by previous_loan_defaults_on_file
loan_status
0.0
0.1
0.2
0.3
0.4
No
Yes
previous_loan_defaults_on_file


Loan Approval Rate by loan_intent
loan_status
0.00
0.05
0.10
0.15
0.20
0.25
0.30
PERSONAL
EDUCATION
MEDICAL
VENTURE
HOMEIMPROVEMENT
DEBTCONSOLIDATION
loan_intent

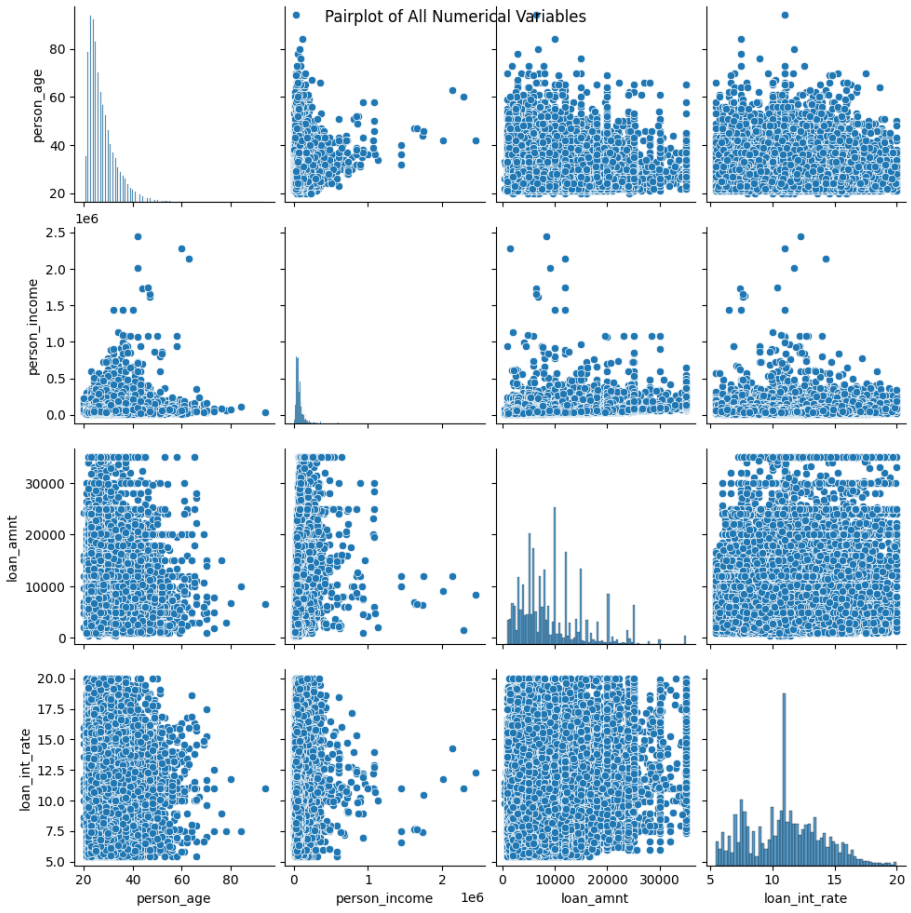
**Person Gender:** Gender seems to have **minimal influence** on loan approval rates.

**Loan intent:** Loan intent categories show varying approval rates, implying that the purpose of the loan may be a **significant factor** in the decision-making progress.

**Previous Loan Defaults:** Applicants with previous defaults almost never got approved, as shown by the barplot. Thus, previous_loan_defaults_on_file is a **very strong categorical predictor**: having a past default drastically lowers approval chances.

### Relationships between Explanatory Variable Pairs

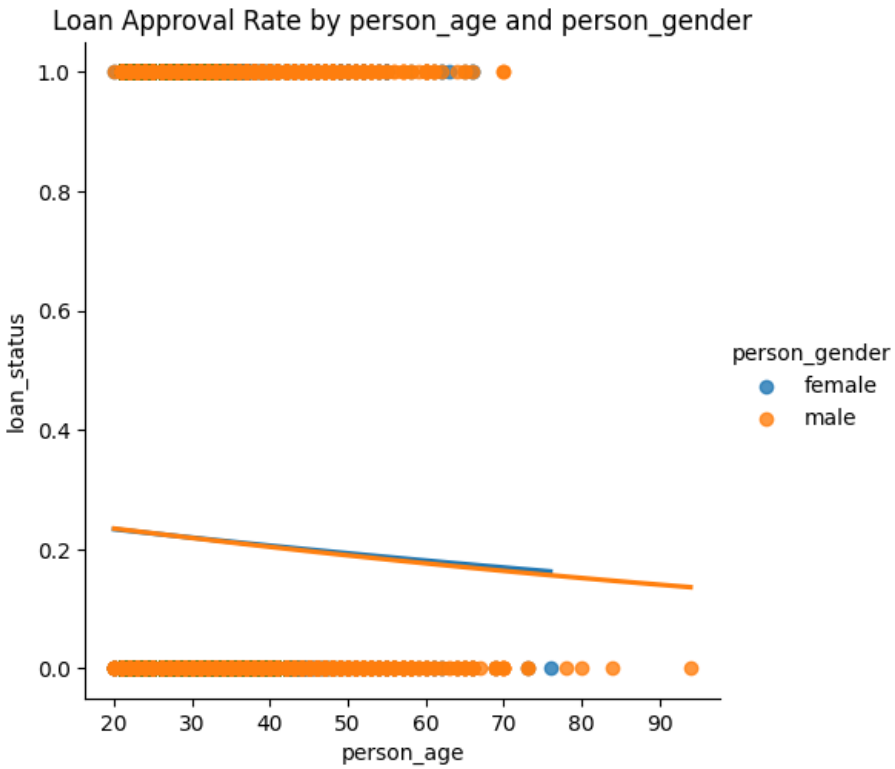
```
sns.pairplot(df[numvars])
plt.suptitle("Pairplot of All Numerical Variables")
plt.show()
```

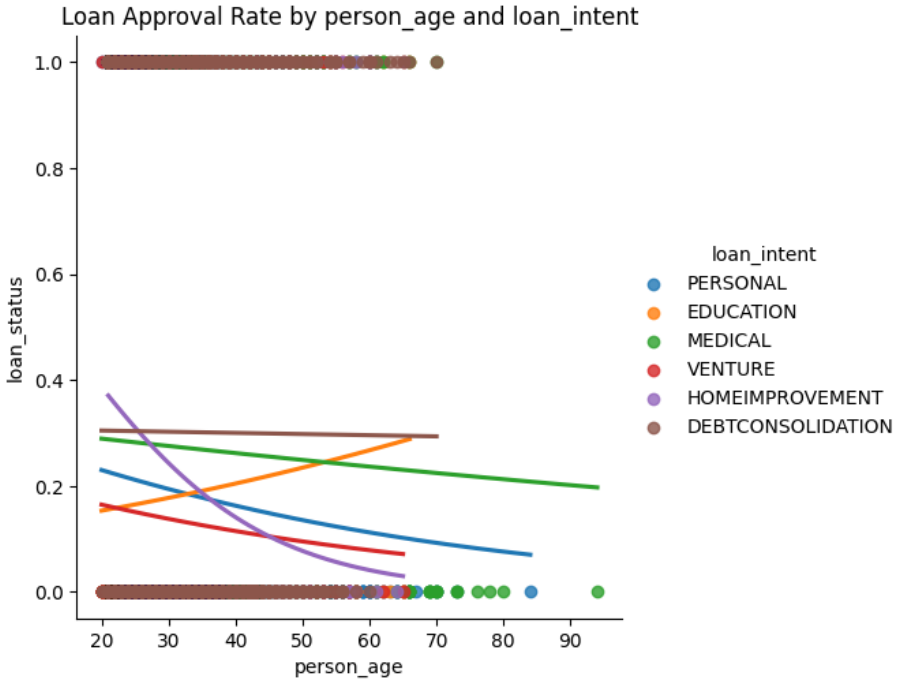


Based on the pairplot, there are **no strong linear relationships** between any pair of numerical explanatory variables. This indicates that multicollinearity should not be a major concern when building the logistic regression model. Each explanatory variable likely provides independent information about the likelihood of loan approval.

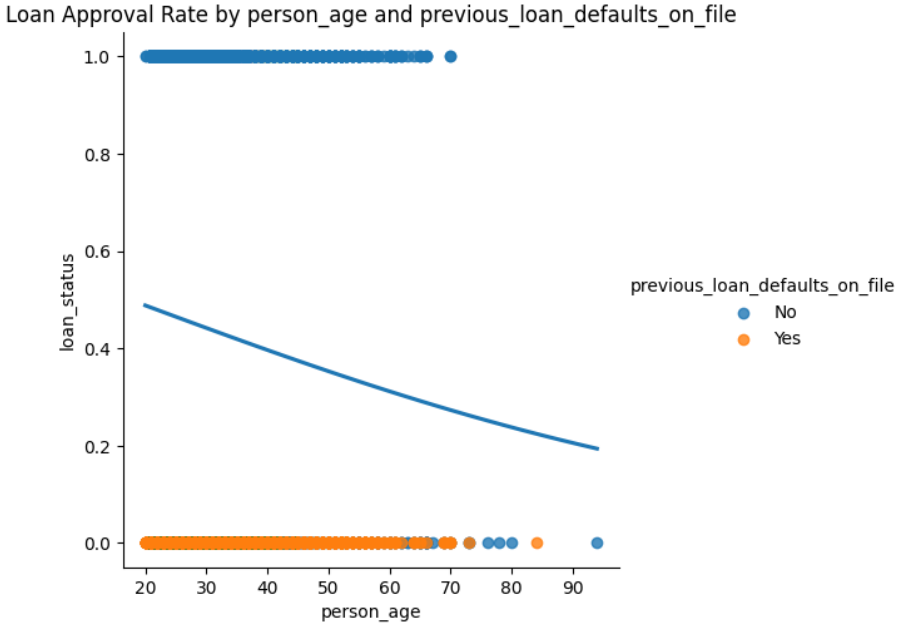
### Interaction Effects

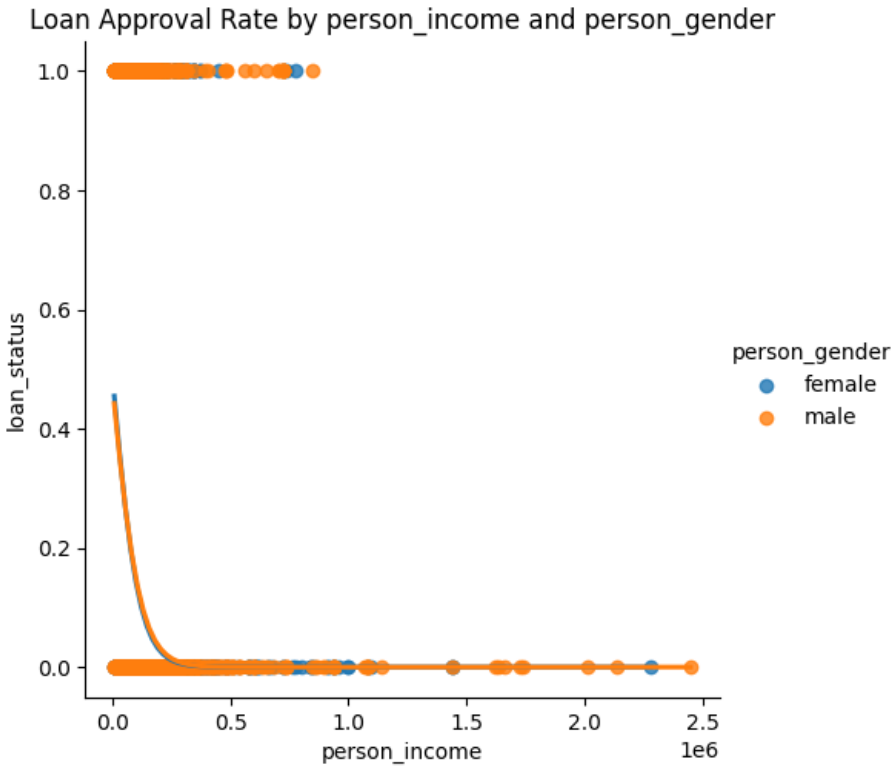
```
for num_var in numvars:
  for cat_var in ['person_gender', 'loan_intent', 'previous_loan_defaults_on_file']:
    sns.lmplot(x=num_var, y="loan_status", hue=cat_var, data=df, logistic=True, ci=False)
    plt.title(f"Loan Approval Rate by {num_var} and {cat_var}")
    plt.show()
```

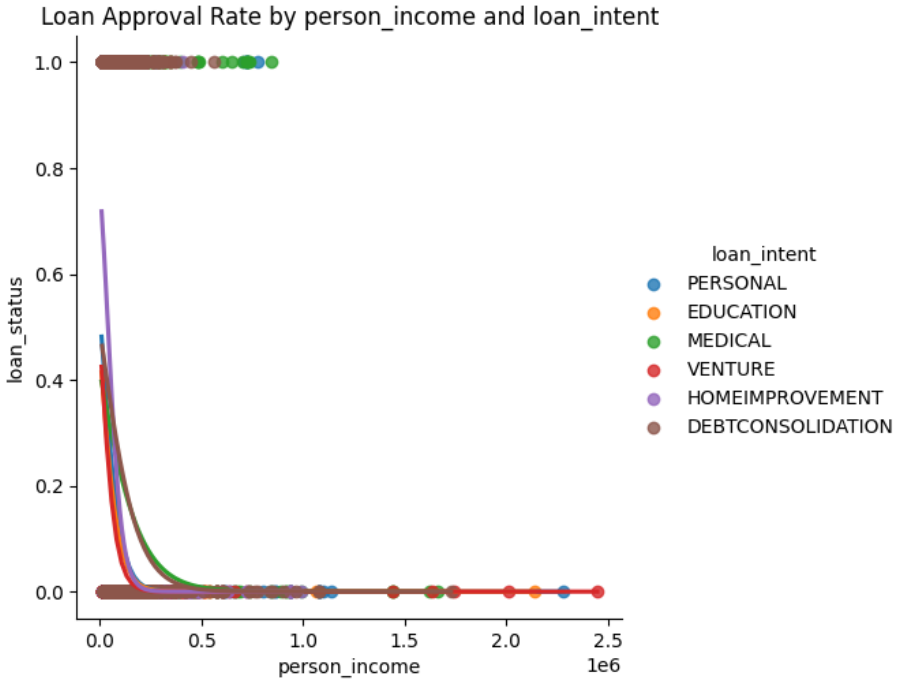




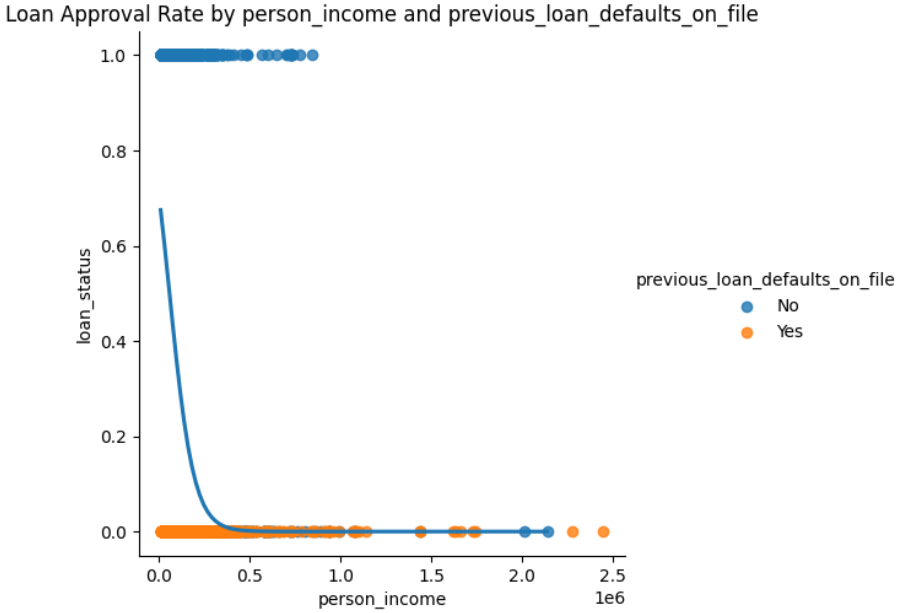

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/numpy/lib/na
  return function_base._ureduce(a,
```

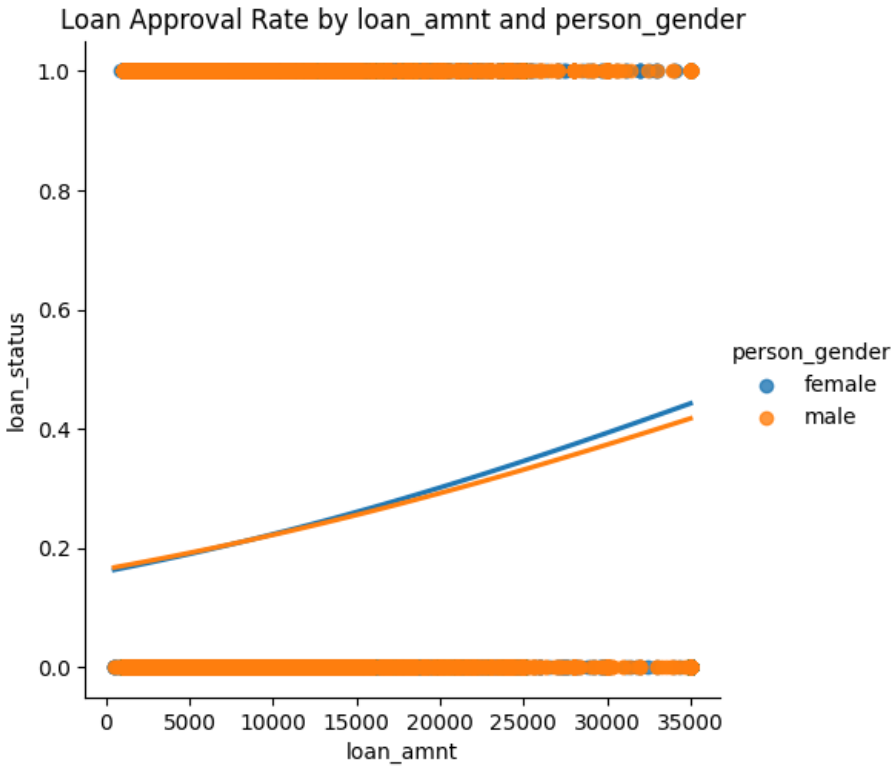


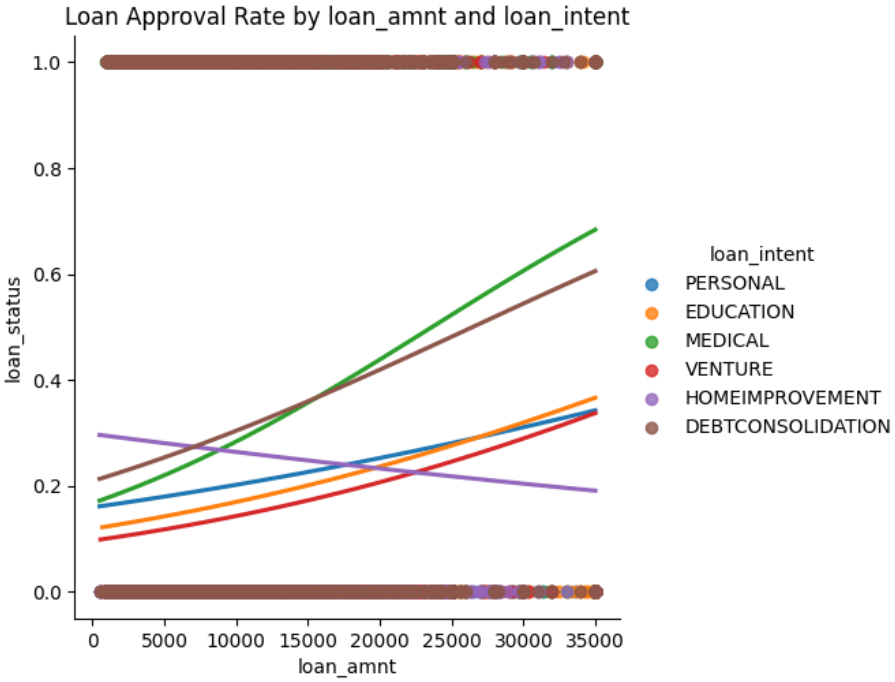

Loan Approval Rate by person_income and person_gender
loan_status
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
person_income
1e6
person_gender
female
male



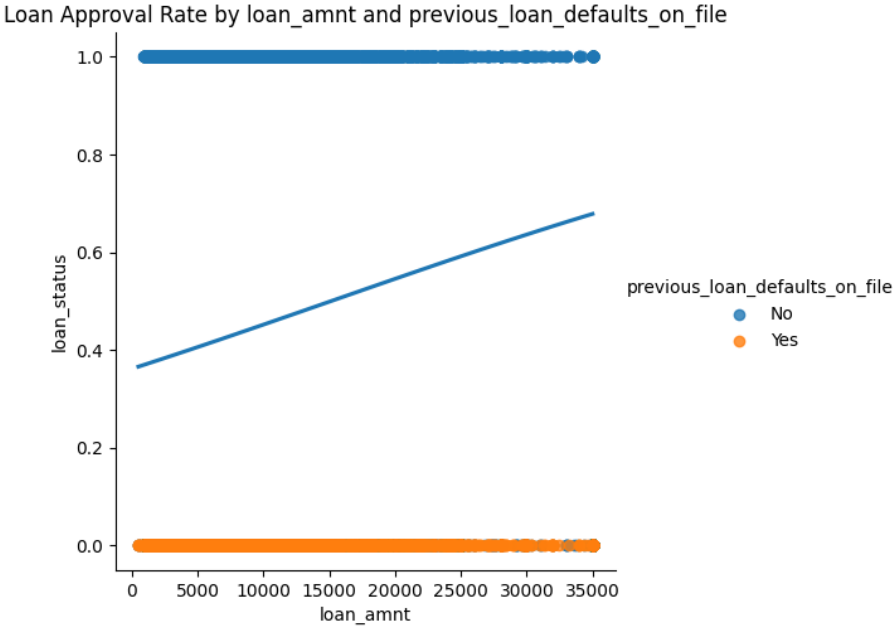

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/numpy/lib/na
  return function_base._ureduce(a,
```

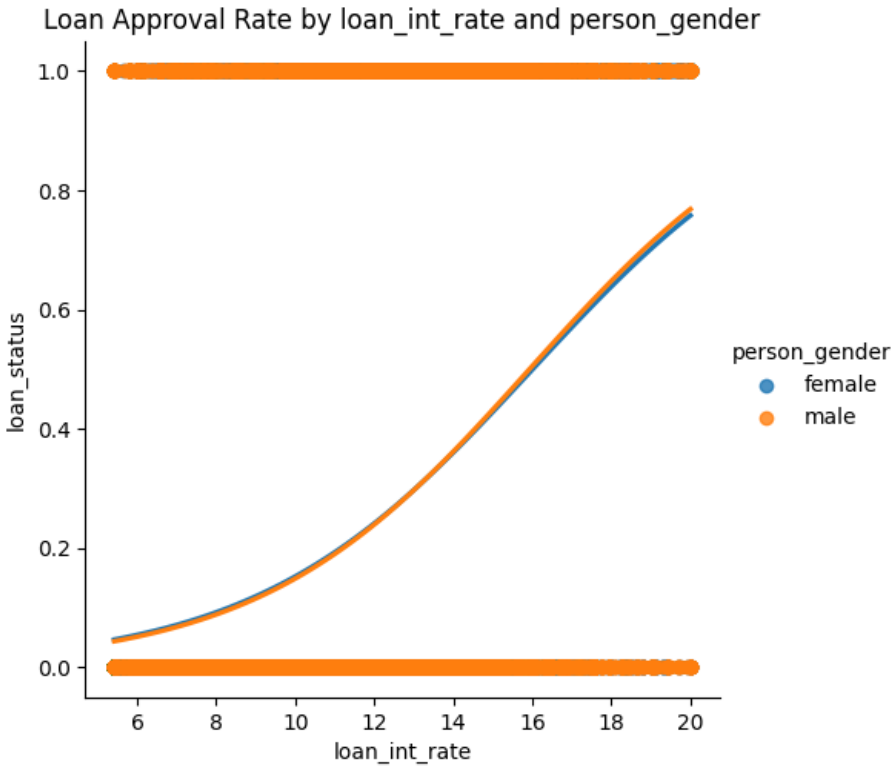


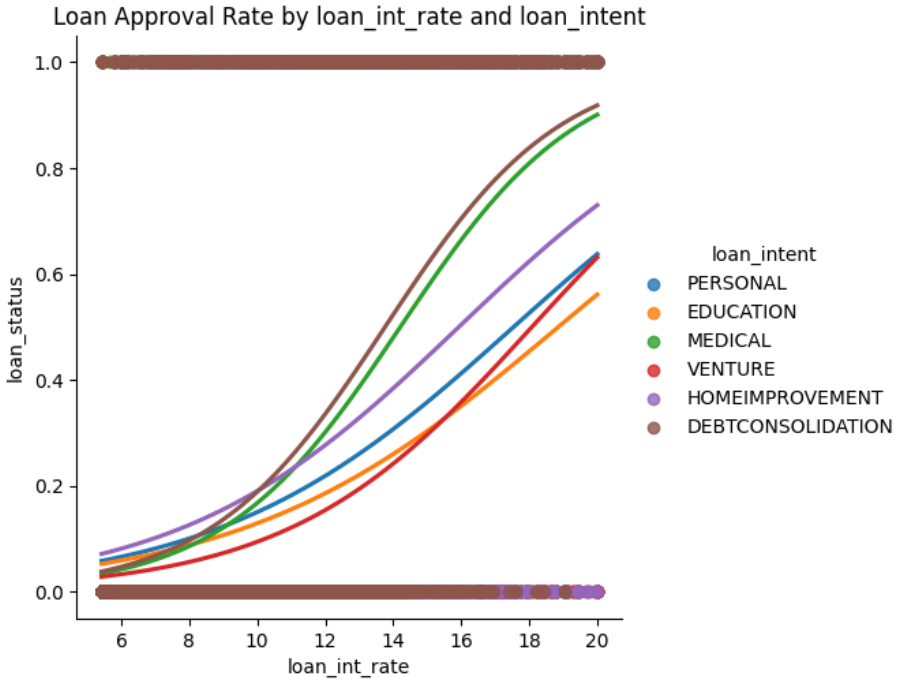

Loan Approval Rate by loan_amnt and person_gender
loan_status
1.0
0.8
0.6
0.4
0.2
0.0
0
5000
10000
15000
20000
25000
30000
35000
loan_amnt
person_gender
female
male



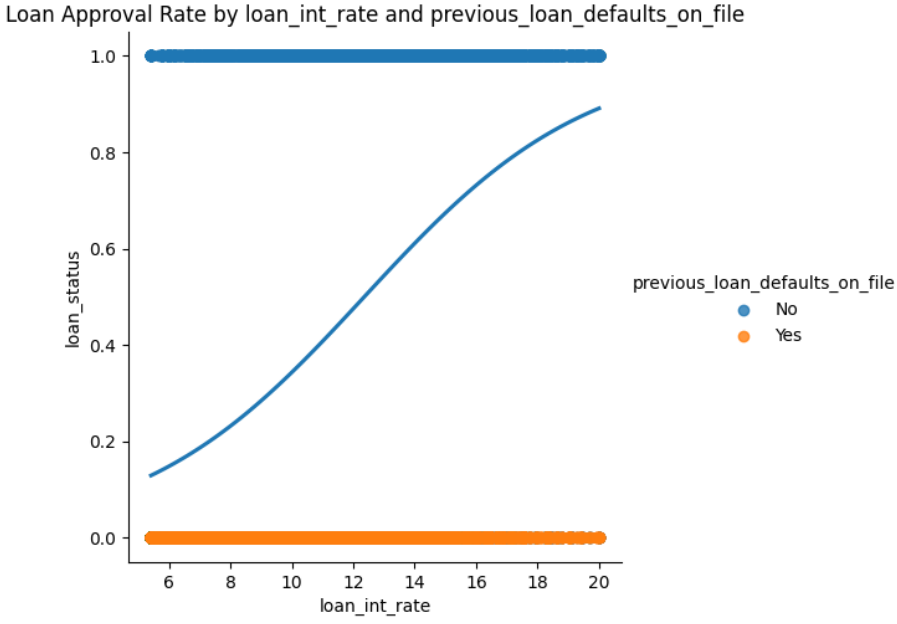

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/numpy/lib/na
  return function_base._ureduce(a,
```




Loan Approval Rate by loan_int_rate and person_gender
loan_status
1.0
0.8
0.6
0.4
0.2
0.0
6
8
10
12
14
16
18
20
loan_int_rate
person_gender
female
male




```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/numpy/lib/na
  return function_base._ureduce(a,
```



We assessed interaction effects between each numerical explanatory variable and categorical explanatory variable (gender, loan intent, previous defaults). Below are the results based on the interaction plots::

**person_age and person_gender:** The interaction plot suggests **no significant interaction**. The slopes for males and females are very similar and close to parallel.

**person_age and loan_intent**: Slopes differ more noticeably across different loan intents, suggesting that **loan intent may interact with age** to impact loan approval.

**person_age and previous_loan_defaults_on_file:** A strong interaction is observed. The approval rates for applicants with and without previous defaults behave very differently with age. This indicates a **strong interaction**.

**person_income and person_gender:** Both genders show a downward trend, with slightly differences in slopes, suggesting a **weak interaction**.

**person_income and loan_intent:** The slopes vary across different loan intents, suggesting **a possible interaction** between income and loan purpose.

**person_income and previous_loan_defaults_on_file:** Clear difference in slopes between applicants with and without defaults, indicating **a strong interaction**.

**loan_amnt and person_gender:** Very similar slopes between males and females, suggesting **no significant interaction**.

**loan_amnt and loan_intent:** Different loan intents have visibly different slopes, indicating **interaction between loan amount and loan purpose**.

**loan_amnt and previous_loan_defaults_on_file**: Strongly different approval patterns for applicants with/without defaults. **A strong interaction** effect is present.

**loan_int_rate and person_gender:** Slopes are similar between genders, indicating **no interaction**.

**loan_int_rate and loan_intent:** There are variations in slopes across different loan intents, with some loan types showing a stronger effect of credit score on approval. This suggests **a potential interaction** between credit score and loan purpose.

**loan_int_rate and previous_loan_defaults_on_file: A very strong interaction** is observed. Applicants with previous defaults have extremely low approval probabilities regardless of credit score, while those without defaults show higher approval probabilities that decrease as credit score worsens.

## 5. Model Data Preprocessing

First, our response variable loan_status **was already a binary (0/1) indicator** variable, where 1 represents loan approval and 0 represents loan rejection, so no transformation was necessary for the response variable.

Next, we created our **features matrix and target array**. We deleted the variables of person_age and person_gender since there are no large interaction with the response variable. The features matrix included both the numerical explanatory variables. The target array contained the corresponding values of the loan_status variable.

It was important to scale all numerical explanatory variables. We applied **standard scaling** to the numerical features to ensure they were on comparable scales. For the categorical explanatory variables, we created **indicator (dummy) variables**, and then turn True/False into 1/0 variables.

```
from sklearn.preprocessing import StandardScaler
df_model = df[features + [response]].copy()

df_scaled = df_model.copy()

scaler = StandardScaler()
df_scaled[numvars] = scaler.fit_transform(df_scaled[numvars])
df_scaled = pd.get_dummies(df_scaled, columns=catvars, drop_first=True)

#Turn all True/False into 1/0
for col in df_scaled.columns:
  if df_scaled[col].dtype == "bool":
    df_scaled[col] = df_scaled[col].astype(int)

X_scaled = df_scaled.drop(columns=["loan_status"])
y = df_scaled["loan_status"]
```

```
X_scaled.head()
```

```
   person_age  person_income  loan_amnt  loan_int_rate  person_gender_male  \
0   -0.972715      -0.125715   4.025004       1.682992                   0
1   -1.141929      -1.067987  -1.359230       0.044832                   0
2   -0.465073      -1.065523  -0.646611       0.625573                   0
3   -0.803501      -0.002455   4.025004       1.417798                   0
4   -0.634287      -0.217516   4.025004       1.095537                   1

   previous_loan_defaults_on_file_Yes  loan_intent_EDUCATION  \
0                                   0                      0
1                                   1                      1
2                                   0                      0
3                                   0                      0
```

```
4                                   0                     0

   loan_intent_HOMEIMPROVEMENT  loan_intent_MEDICAL  loan_intent_PERSONAL  \
0                            0                    0                     1
1                            0                    0                     0
2                            0                    1                     0
3                            0                    1                     0
4                            0                    1                     0

   loan_intent_VENTURE
0                    0
1                    0
2                    0
3                    0
4                    0
```

```
y.head()
```

```
0    1
1    0
2    1
3    1
4    1
Name: loan_status, dtype: int64
```

## 6. Feature Selection with k-Fold Cross-Validation

We select LASSO to find out whether there is a feature that doesn't provide enough predictive power. We choose LASSO instead of Ridge and elactic net because LASSO tends to set slopes of variables with little predictive power zero, from which we can easily identify redundant variables.

We try lambdas ranging from 0.00001 to 0.1 with 100 trials and find that lambda of 0.00001 generated the best model. The best model has an auc of 0.9388. The slopes of the best model demonstrates that all variables bring enough predictive power into the model.

```
lambda_range = np.linspace(0.00001, 0.1, 100)
auc_scores = []
models = []
for lam in lambda_range:
    log_mod = LogisticRegression(penalty='l1', solver='liblinear', C=1/lam, max_iter=1000)
    cross_val = KFold(n_splits=5, shuffle=True, random_state=207)
    test_fold_auc=cross_val_score(log_mod, X_scaled, y, cv=cross_val, scoring="roc_auc")
    auc_scores.append(np.mean(test_fold_auc))
    models.append(log_mod)

best_index = np.argmax(auc_scores)
```

```
best_lambda = lambda_range[best_index]
best_auc = auc_scores[best_index]
best_mod = models[best_index]
```

```
plt.figure(figsize=(8, 5))
plt.plot(lambda_range, auc_scores, marker='o')
plt.xlabel('Lambda')
plt.ylabel('AUC')
plt.legend()
plt.grid(True)
plt.show()
```

```
print("Best Lambda:", best_lambda)
print("Highest Average AUC:", best_auc)
```

```
  Cell In[19], line 10
    print("Best Lambda:" best_lambda)
          ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

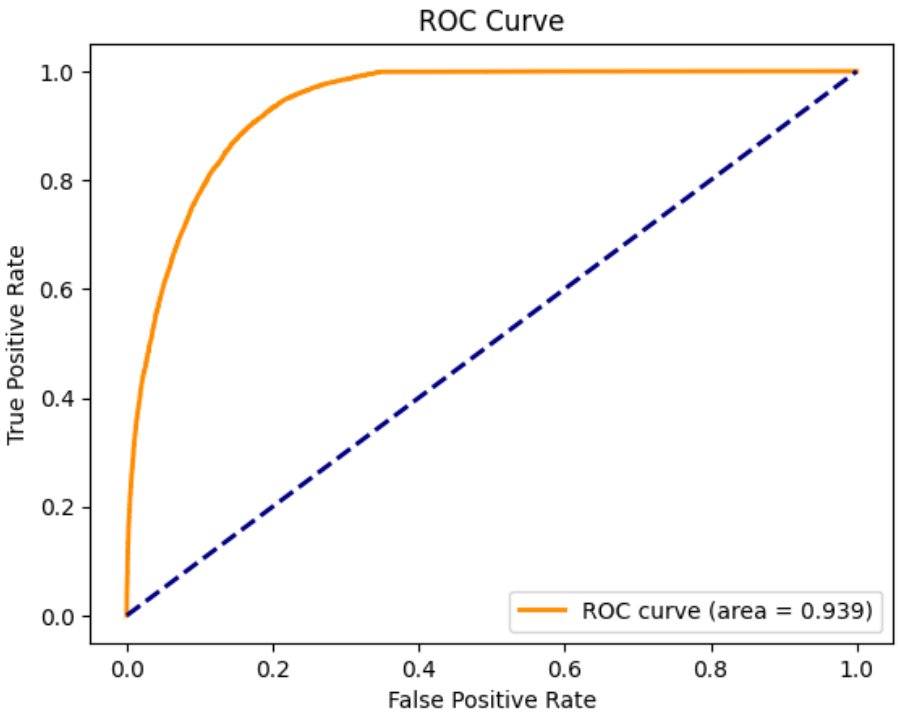
```
best_mod.fit(X_scaled, y)
best_mod.coef_[0]
```

```
array([ -0.07651247,  -1.65765415,   0.62972975,   0.87464123,
         0.03121984, -16.44922874,  -0.8696228 ,  -0.13032501,
        -0.28654369,  -0.69664404,  -1.13540222])
```

```
pred_y = best_mod.predict_proba(X_scaled)[:, 1]
```

```
def plot_roc(fpr, tpr, auc, lw=2):
    plt.plot(fpr, tpr, color='darkorange', lw=lw,
             label='ROC curve (area = '+str(round(auc,3))+')')
    plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve')
    plt.legend(loc="lower right")
    plt.show()
```

```
def plot_roc_curve(a, b):
    fpr, tpr, thresholds = roc_curve(a, b)
    auc = roc_auc_score(y_true=a, y_score=b)
    plot_roc(fpr, tpr, auc)

plot_roc_curve(y, pred_y)
```




```
full_mod = LogisticRegression(penalty='none', max_iter=1000)
cross_val = KFold(n_splits=5, shuffle=True, random_state=207)
test_fold_auc=cross_val_score(log_mod, X_scaled, y, cv=cross_val, scoring="roc_auc")
print(test_fold_auc.mean())
print(test_fold_auc.std())
```

```
0.9382195869291478
0.0012878140753577842
```

## 7. Best Model Discussion

According to the average test auc score, the LASSO model outperforms the full model. Therefore, LASSO is the best model.

```
from sklearn.model_selection import train_test_split

df_train, df_test = train_test_split(df_model, test_size=0.2, random_state=555)
df_train.head()
```

```
       person_age  person_income  loan_amnt  loan_int_rate person_gender  \
29512        39.0        37327.0    10375.0           8.90          male
1677         21.0        35455.0     8000.0           8.90          male
26979        30.0       145301.0    15000.0          10.99        female
```

```
2250         21.0        38606.0    13000.0          13.11          male
16683        22.0        74488.0    12000.0          13.49          male

      previous_loan_defaults_on_file      loan_intent  loan_status
29512                            Yes        EDUCATION            0
1677                             Yes        EDUCATION            0
26979                             No  HOMEIMPROVEMENT            0
2250                              No        EDUCATION            1
16683                            Yes          MEDICAL            0
```

```
scaler = StandardScaler()
df_train[numvars] = scaler.fit_transform(df_train[numvars])
df_train = pd.get_dummies(df_train, columns=catvars, drop_first=True)
for col in df_train.columns:
  if df_train[col].dtype == "bool":
    df_train[col] = df_train[col].astype(int)

df_train.head()
```

```
       person_age  person_income  loan_amnt  loan_int_rate  loan_status  \
29512    1.915444      -0.662247   0.125732      -0.706484            0
1677    -1.141818      -0.691190  -0.249989      -0.706484            0
26979    0.386813       1.007103   0.857398      -0.004933            0
2250    -1.141818      -0.642473   0.541002       0.706689            1
16683   -0.971970      -0.087714   0.382803       0.834244            0

       person_gender_male  previous_loan_defaults_on_file_Yes  \
29512                   1                                  1
1677                    1                                  1
26979                   0                                  0
2250                    1                                  0
16683                   1                                  1

       loan_intent_EDUCATION  loan_intent_HOMEIMPROVEMENT  \
29512                      1                            0
1677                       1                            0
26979                      0                            1
2250                       1                            0
16683                      0                            0

       loan_intent_MEDICAL  loan_intent_PERSONAL  loan_intent_VENTURE
29512                    0                     0                    0
1677                     0                     0                    0
26979                    0                     0                    0
2250                     0                     0                    0
16683                    1                     0                    0
```

```
X_train = df_train.drop(columns=["loan_status"])
y_train = df_train["loan_status"]
```

```
model = LogisticRegression(penalty='l1', solver='liblinear', C=1/best_lambda, max_iter=1000)
df_slopes = pd.DataFrame(model.coef_[0], index=X_train.columns, columns=["Slope"])
df_slopes
```

| | Slope |
|---|---|
| person_age | -0.098622 |
| person_income | -1.717790 |
| loan_amnt | 0.630811 |
| loan_int_rate | 0.883997 |
| person_gender_male | 0.021128 |
| previous_loan_defaults_on_file_Yes | -15.565764 |
| loan_intent_EDUCATION | -0.852173 |
| loan_intent_HOMEIMPROVEMENT | -0.124539 |
| loan_intent_MEDICAL | -0.283975 |
| loan_intent_PERSONAL | -0.732838 |
| loan_intent_VENTURE | -1.134097 |

```
model.intercept_
```

```
array([-0.21679644])
```

Equation:

$$\hat{p} = \frac{1}{1 + \exp\left(-\begin{bmatrix} -1.18539112 \\ -0.022838 \cdot \text{person\_age} \\ -0.000026 \cdot \text{person\_income} \\ +0.000098 \cdot \text{loan\_amnt} \\ +0.260390 \cdot \text{loan\_int\_rate} \\ +0.010904 \cdot \text{person\_gender\_male} \\ -14.967081 \cdot \text{previous\_loan\_defaults\_on\_file\_Yes} \\ -0.927160 \cdot \text{loan\_intent\_EDUCATION} \\ -0.193122 \cdot \text{loan\_intent\_HOMEIMPROVEMENT} \\ -0.341704 \cdot \text{loan\_intent\_MEDICAL} \\ -0.782793 \cdot \text{loan\_intent\_PERSONAL} \\ -1.179984 \cdot \text{loan\_intent\_VENTURE} \end{bmatrix}\right)}$$

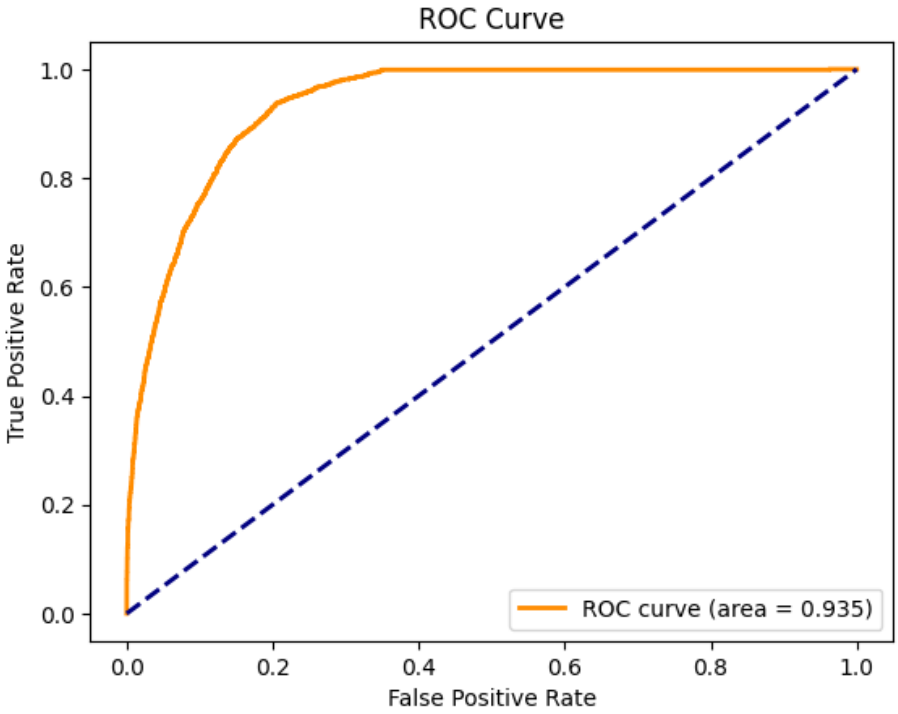
```
for i in range(len(numvars)):
    for m in range(i + 1, len(numvars)):
        if abs(df_train[numvars].corr()[numvars[i]][numvars[m]]) >= 0.7:
            print(numvars[i], numvars[m], df_train[numvars].corr()[numvars[i]][numvars[m]])
```

```
df_train[numvars].corr()
```

| | person_age | person_income | loan_amnt | loan_int_rate |
|---|---|---|---|---|
| person_age | 1.000000 | 0.143927 | 0.048725 | 0.014395 |
| person_income | 0.143927 | 1.000000 | 0.306158 | -0.003958 |
| loan_amnt | 0.048725 | 0.306158 | 1.000000 | 0.143510 |
| loan_int_rate | 0.014395 | -0.003958 | 0.143510 | 1.000000 |

There is no issue of multicollinearity. Since the training data is scaled and there is no issue of multicollinearity, we can interpret the magnitudes of the slopes as indicating how important the corresponding explanatory variable is when it comes to predicting out response variable. previous_loan_defaults_on_file is the most important explanatory variable as indicated by the slopes. As the lambda value is low and no slope is set to zero, we believe that there is no issue of overfitting.

```
df_test[numvars] = scaler.fit_transform(df_test[numvars])
df_test = pd.get_dummies(df_test, columns=catvars, drop_first=True)
for col in df_test.columns:
  if df_test[col].dtype == "bool":
    df_test[col] = df_test[col].astype(int)

X_test = df_test.drop(columns=["loan_status"])
y_test = df_test["loan_status"]

y_test_pred = model.predict_proba(X_test)[:, 1]
plot_roc_curve(y_test, y_test_pred)
print(roc_auc_score(y_true=y_test, y_score=y_test_pred))
```




```
0.9352743756826806
```

The test auc score of this model is 0.935, which is pretty high. Since this is to predict whether we approve the loan, we should be careful not to provide it to someone who will never pay back. Therefore, we should choose a threshold with low false positive rate. However, we don't want to deny all clients. Therefore, we want the threshold with the lowest false positive rate that has a true positive rate above 60.

```
fpr, tpr, thresholds = roc_curve(y_test, y_test_pred)
df_res = pd.DataFrame({"FPR": fpr, "TPR": tpr, "Thresholds": thresholds})
best = df_res[df_res['TPR'] > 0.6].nsmallest(1, 'FPR')
print('Threshold', best['Thresholds'].values[0])
print('FPR', best['FPR'].values[0])
print('TPR', best['TPR'].values[0])
```

```
Threshold 0.5968519140342083
FPR 0.05297729544480937
TPR 0.6002004008016032
```

The threshold that satisfies the requirement is 0.597, which has a true positive rate of 0.6 and false positive rate of 0.05. This means that 60% of all the actual positive cases are correctly predicted as positive. Meanwhile, the False Positive Rate is 5%, meaning that 5% of the actual negative cases are incorrectly

predicted as positive.

## 8. Additional Analysis/Insight

To further investigate the characteristics of loan applicants, we conducted a cluster analysis using K-Means clustering. This analysis, outside the scope of our initial research goals, aims to identify distinct groups of applicants based on their profile features. This additional analysis can offer deeper insights into applicant behavior and patterns, which can help refine our understanding of loan approval determinants and better tailor our model for specific subgroups.

```
KMeans_model = KMeans(n_clusters=3, random_state=207)
```

```
cluster_labels = KMeans_model.fit_predict(X_scaled)
```

```
df['cluster'] = cluster_labels
```

```
/var/folders/kj/xx0t04xd2j3fqt15rhd_rzcr0000gn/T/ipykernel_10617/2329045320.py:1: SettingWit
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guic
  df['cluster'] = cluster_labels
```

```
cluster_summary = df.groupby('cluster')[numvars + ['loan_status']].mean()
```

```
cluster_summary
```

| cluster | person_age | person_income | loan_amnt | loan_int_rate | loan_status |
|---|---|---|---|---|---|
| 0 | 28.787065 | 73870.252341 | 7449.259528 | 8.205212 | 0.103022 |
| 1 | 29.549471 | 133890.468783 | 19319.571794 | 12.275271 | 0.274154 |
| 2 | 26.042338 | 61273.942107 | 7135.830852 | 12.893825 | 0.303528 |

Based on this result, we can summarize the characteristics of our clusters as below:

**Cluster 0:** This cluster is composed of applicants with the lowest yearly income and loan amount, as well as a low interest rate and moderate age. This group has the lowest loan approval rate of just 10%, suggesting these applicants are generally considered high-risk.

**Cluster 1:** This group has the highest yearly income and loan amount, along with moderate age and interest rate. This cluster has a moderate loan approval rate of 27%, indicating a moderate risk profile.

**Cluster 2:** This group is characterized by the youngest age and lowest yearly income, but moderate loan amounts and high interest rates. Despite this, they have the highest loan approval rate of 30%, suggesting they might represent a specific niche of applicants who are considered more favorably despite their youth and income level.

This additional analysis extends our secondary research goal by providing more interpretative insights about the nature of the relationships between the variables in our dataset. The person considering taking on an educational loan, as mentioned in our introduction, can use these profiles to gauge where they fall in the broader loan applicant pool. If they fall into cluster 0, they may consider lowering their requested loan amount, or if they are in cluster 2, they may use that information to make further research into if there are specific characteristics in that group that can further increase their chances of loan approval. Financial institutions could use these cluster profiles to better understand the reasons behind loan application outcomes. For instance, the high approval rate in Cluster 2, despite the low yearly income and moderate interest rates, may point to the influence of other factors not explicitly captured in our analysis, such as quality of degree sought or university of attendance.

## 9. Conclusion

In this project, we aimed to build a predictive model for loan approval status and gain interpretative insights into the factors influencing these decisions. We successfully constructed a LASSO logistic regression model that achieved a high average test AUC score of 0.935. This indicates a strong ability to discriminate between loan applications that are likely to be approved versus those likely to be rejected. Our feature selection process, guided by K-fold cross-validation, ensured that the model utilized the most relevant variables while mitigating overfitting. Further insights were gained from a cluster analysis, which allowed us to identify different groups of applicants based on their characteristics and approval patterns.

**Recommendation**

Given the high AUC score, I would cautiously recommend this model to be used by the U of I grad a few years out of college that we mentioned in our introduction. The model's high AUC score on test data suggests that it generalizes well to new, unseen data, which is critical for making reliable predictions. However, the model should be used as an advisory tool rather than a definitive decision-maker. The individual should weigh the model's prediction alongside their personal circumstances and financial goals. It should serve as a valuable piece of information in a broader decision-making process, and not a directive.

**Shortcomings/Caveats**

While our LASSO logistic regression model demonstrates strong predictive performance, it is crucial to acknowledge the limitations of our analysis and potential areas for improvement. Though a test AUC score of 0.935 indicates strong discriminatory power on unseen data, we cannot definitively claim that our model achieves the absolute highest possible average test AUC score. This is because we explored only a limited set of model types. If we add backwards elimination with cross-validation and forward selection with cross-validation, we can improve or double-check our result. Furthermore, while we considered some

interaction effects, a deeper investigation into higher-order interactions or non-linear relationships between explanatory variables might further enhance the model's predictive capabilities. Additionally, we acknowledge the absence of formal statistical testing for overfitting; although the low lambda value suggests a minimal risk of overfitting, future work should include these tests to rigorously assess this possibility. In addition to the limitations of model exploration and evaluation, our study's scope could be expanded by incorporating more feature selecting, particularly financial ratios, to capture a more holistic view of an applicant's financial situation. Addressing these shortcomings will pave the way for future refinements and a more comprehensive understanding of loan approval prediction.

**Future Work**

Based on what we learned, one promising direction for future work would be to incorporate financial ratio data and explore a broader range of features related to an applicant's financial situation. Key ratios such as debt-to-income, loan-to-value, and credit utilization are commonly used in credit risk assessment. Including these metrics would enhance the model's ability to capture applicants' overall financial health and improve its accuracy in predicting loan approval status. Furthermore, analyzing these ratios within different clusters could lead to a more nuanced understanding of financial risk factors among various applicant segments.

## References

(1) Mian, Atif, and Amir Sufi. House of Debt: How They (and You) Caused the Great Recession, and How We Can Prevent It from Happening Again. University of Chicago Press, 2014. https://press.uchicago.edu/ucp/books/book/chicago/H/bo20832545.html

(2) Stein, Tristan, and Emily Wielk. "Student Loan Default: How Policy and Politics Are Failing Borrowers." Bipartisan Policy Center, 16 Apr. 2025, https://bipartisanpolicy.org/blog/student-loan-default-how-policy-and-politics-are-failing-borrowers/.