CREDIT RISK ANALYSIS

Predictive model: Logistic regression with L1 regularization

NOVEMBER 26, 2025

LELETHU CONJWAYO

Contents

Introduction

Credit risk is defined as the potential for financial loss a creditor faces due to a debtor’s failure to repay a loan or meet contractual obligations. The failure to fulfill the terms of the loan or the financial contract is known as borrower default. Credit risk modelling is crucial for financial institutions to reduce loss , make informed lending decisions and overall balance risk and reward.

The purpose of this project is to build a predictive model that identifies whether a loan applicant is likely to default or not. In this analysis a logistic regression model with LASSO (L1) regularization was used.

Logistic regression is a classification model used to predict the probability of an observation belonging to particular class , the model outputs a probability between 0 and 1 which represents how likely the positive outcome is to occur.

The model then uses a decision threshold to convert this probability into a class label, if the predicted probability is greater than or equal to the threshold, the model predicts the positive class (in our case high risk of default) if it is below the threshold, it predicts the negative class (low risk of default).

LASSO regularization works by adding a penalty term to a model’s loss function which shrinks the coefficients to be smaller and some ultimately zero thus having no effect on the model. We chose this because it performs ‘automatic’ feature selection , in essence it ‘deletes’ the variables which have less effect on our target variable.

Method

The dataset used in this analysis comes from the Kaggle website and is titled 'credit_risk_dataset', the dataset contains information about the applicant's financial history like personal income ,loan percent income, loan history characteristics , age and the target dependent variable – loan status, which indicated whether the applicant defaulted (1) or not (0).

Data processing:

All data processing and visualization were done using python.

Handling categorical variables

The first step was to convert categorical variables into numerical variables via one hot encoding.

One hot encoding converts each category into a new binary column where '1' indicates the presence of that category for a given row and '0' indicates its absence. This prevents the model from mis interpreting that data by creating a false sense of hierarchy between categories.

Scaling numerical variables

Numerical variables were standardized using Standard Scaler (Z- score normalization) which works by transforming the variables to have a mean of 0 and standard deviation of 1.

Scaling improves performance of models like logistic regression because it prevents features with large values from dominating those with smaller ones. In essence it ensure that all variables contribute equally to the model's performance regardless of their original scales.

Handling Missing values

There were missing values in 2 of the numerical variables columns :

1.Person employment length

2.Loan interest rate.

To fix this we imputed the missing values with estimated data , for both of the variables we visualized the data using the Quantile-Quantile plot which works by comparing the quantiles of the observed data to the quantiles of the normal distribution. For person employment length we decided to impute the missing values with the median because it was not normal and for the loan interest rate we imputed with the mean since it was mostly normal.

Modelling

Constructed a logistic regression model with Lasso regularization.

As stated above logistic regression has a decision threshold , the standard threshold =0.5 but after training the model the accuracy was high (approximately 86%) but the recall for the positive class was low (approximately 64%) . In credit risk analysis this is a problem because recall is the most important metric because in our case it measures the percentage of actual clients that default , if the model is not good at identifying that this can lead to just 'bad' things happening at the company (financially of-course).

Because of that I prioritized high recall while still keeping precision and overall accuracy of the model good enough. To find the best threshold I used the ROC curve which illustrates the performance of the precision and recall of the model at varying threshold values.

Ultimately the chosen threshold was =0.3 , where there was a higher recall , good enough precision and accuracy.

RESULTS

DESCRIPTIVE ANALYSIS

Before we get to the modelling results, lets analyze the numerical variables that are most relevant to loan status. Although the dataset includes both categorical and numerical variables, I choose to focus on the numerical variables for this part. This is because they provide measurable insights into clients’ financial capacity and repayment behavior, which are essential for understanding default risk. Numerical variables give a more intuitive understanding of patterns in loan performance.

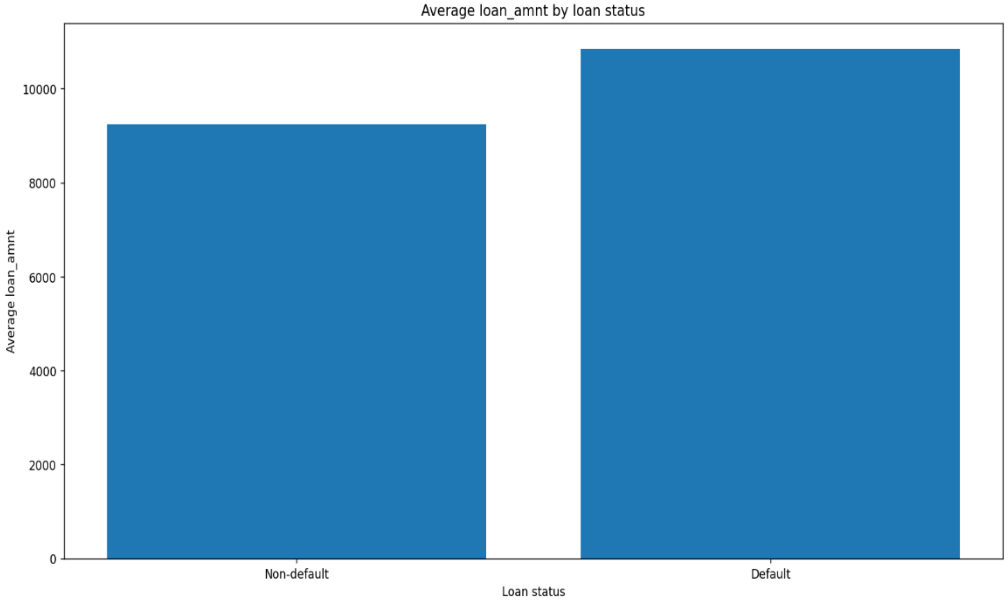


Figure 1: Average loan amount by loan status

Figure 1 illustrates that clients are less likely to default when their loan amounts range between R8,000 and R10,000 and more likely to default when their loan amounts are higher.

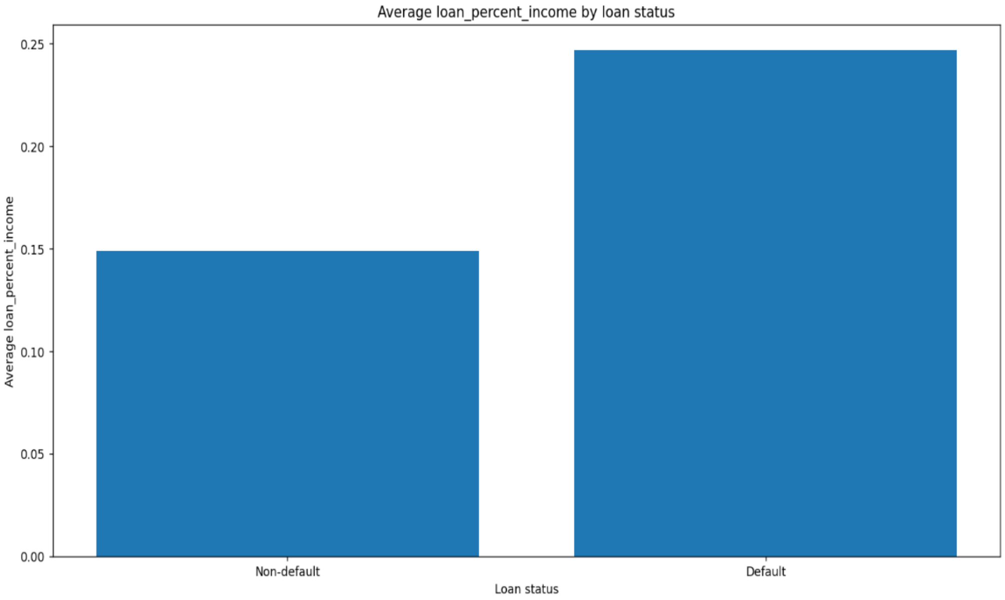


Figure 2: Average loan percentage income by loan status

Similarly, Figure 2 shows that clients are more likely to default when their loan-to-income ratio is higher, i.e. when the loan represents a larger proportion of their income. On the other hand, defaults are less frequent when the loan takes up a smaller percentage of their income.

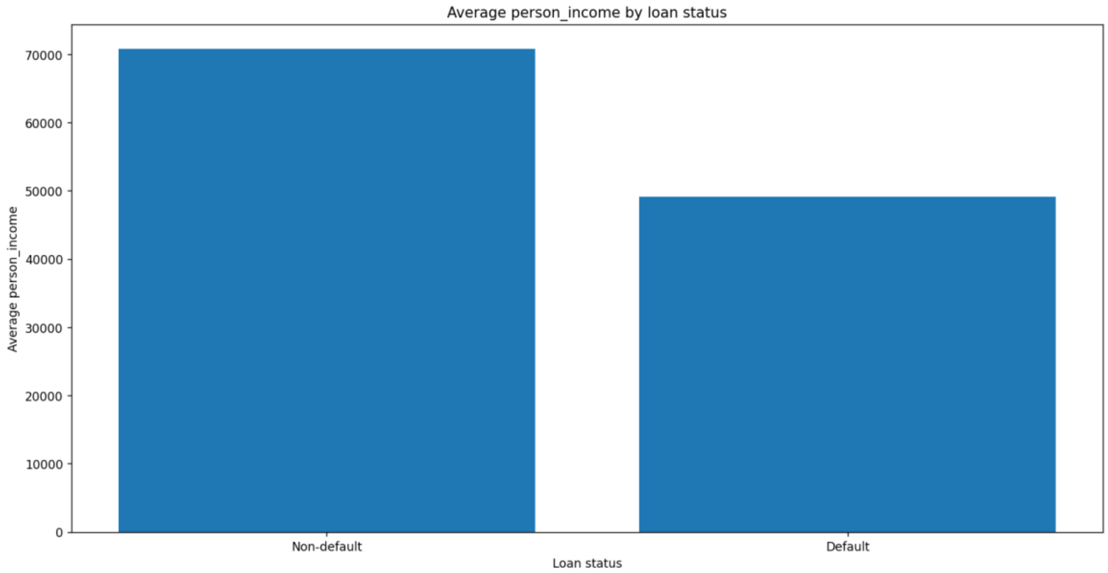


Figure 3: Average personal income by loan status

Figure 3 shows that clients with higher personal incomes above approximately R 70 ,000 are less likely to default whereas those earning between R40,000 and R50,000 are more likely to default. Since income directly affects the loan to income ratio this is expected.

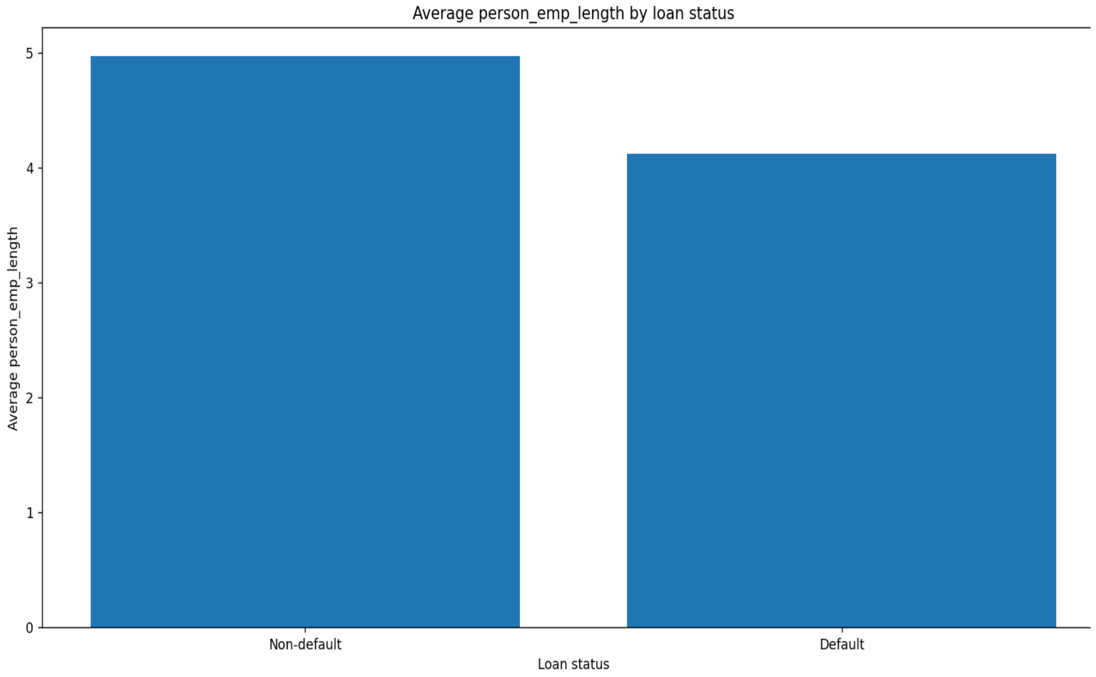


Figure 4: Average person employment length by loan status

Figure 4 shows that clients with longer employment (>=5 years) are less likely to default, while those with shorter employment durations are more likely to default. This is expected because longer employment leads to higher income and more financial stability which reduces loan to income ratio.

Interpretation:

Although these are different variables, they are closely connected. Employment length affects income; income affects the loan to income ratio and the loan to income ratio is strongly linked to default risk. Larger loans result in higher monthly payments and when the debt is large relative to a client's income, there is a higher risk that they will struggle to keep up with payments. Psychologically, large loans can feel overwhelming, causing stress to clients because often clients overestimate their ability to repay a large loan when taking it out, which contributes to defaults. A solution to this is to carefully assess a client's debt-to-income ratio before approving a loan. By evaluating their income and their credit history carefully before approving a loan, this would help us better ensure that we reduce the likelihood of a default.

Quantile-Quantile plots for the NaN values:

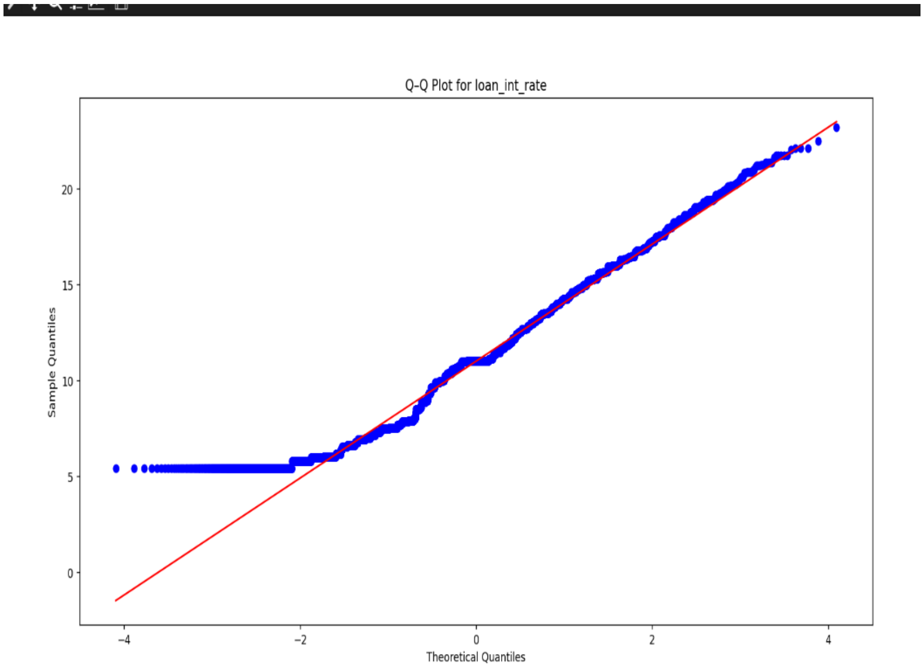


Figure 5: Quantile-Quantile plot for loan interest rate

The observed data points show that the points align somewhat closely with the reference line , suggesting that the variable is approximately normally distributed which as stated above is why I imputed missing values using the mean.

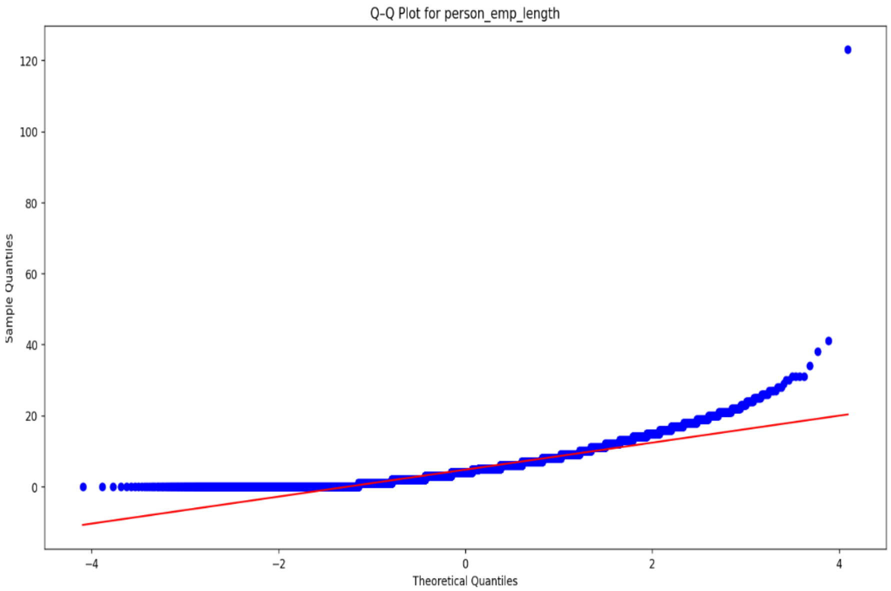
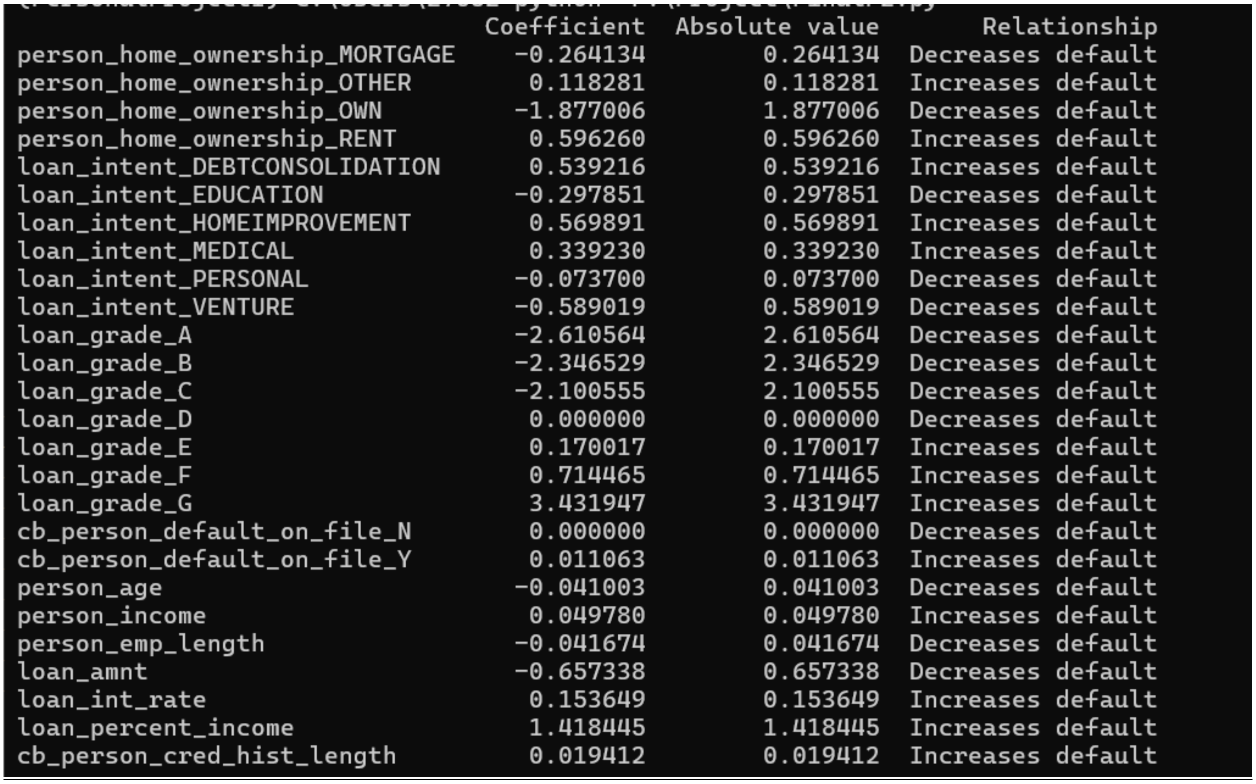


Figure 6: Quantile -Quantile plot for employment length

The observed data appoints deviate noticeably from the reference line indicating that the variable is not normally distributed. Because of this I imputed the missing values using the median rather than the mean , as the median is the standard to skewness and outliers.

Model Metrics:

Coefficients table:



```
                                  Coefficient  Absolute value       Relationship
person_home_ownership_MORTGAGE    -0.264134          0.264134  Decreases default
person_home_ownership_OTHER        0.118281          0.118281  Increases default
person_home_ownership_OWN         -1.877006          1.877006  Decreases default
person_home_ownership_RENT         0.596260          0.596260  Increases default
loan_intent_DEBTCONSOLIDATION      0.539216          0.539216  Increases default
loan_intent_EDUCATION             -0.297851          0.297851  Decreases default
loan_intent_HOMEIMPROVEMENT        0.569891          0.569891  Increases default
loan_intent_MEDICAL                0.339230          0.339230  Increases default
loan_intent_PERSONAL              -0.073700          0.073700  Decreases default
loan_intent_VENTURE               -0.589019          0.589019  Decreases default
loan_grade_A                      -2.610564          2.610564  Decreases default
loan_grade_B                      -2.346529          2.346529  Decreases default
loan_grade_C                      -2.100555          2.100555  Decreases default
loan_grade_D                       0.000000          0.000000  Decreases default
loan_grade_E                       0.170017          0.170017  Increases default
loan_grade_F                       0.714465          0.714465  Increases default
loan_grade_G                       3.431947          3.431947  Increases default
cb_person_default_on_file_N        0.000000          0.000000  Decreases default
cb_person_default_on_file_Y        0.011063          0.011063  Increases default
person_age                        -0.041003          0.041003  Decreases default
person_income                      0.049780          0.049780  Increases default
person_emp_length                 -0.041674          0.041674  Decreases default
loan_amnt                         -0.657338          0.657338  Decreases default
loan_int_rate                      0.153649          0.153649  Increases default
loan_percent_income                1.418445          1.418445  Increases default
cb_person_cred_hist_length         0.019412          0.019412  Increases default
```

Figure 7: Coefficients table showing the relationship between each coefficient and relationship to default

After applying L1 regularization , we got to see our data in a different perspective regarding which variables strongly influence the probability of loan default. One of the notable findings is that Loan Grade G has the largest positive coefficient ,which shows that the relationship between loan grade G and default is very strong and thus increases default risk Another influential variable is loan percent income which also has a relatively large positive coefficient. Which as discussed in the previous section , clients who allocate a higher proportion of their income toward their loan payments face a higher likelihood of default.

ROC CURVE:

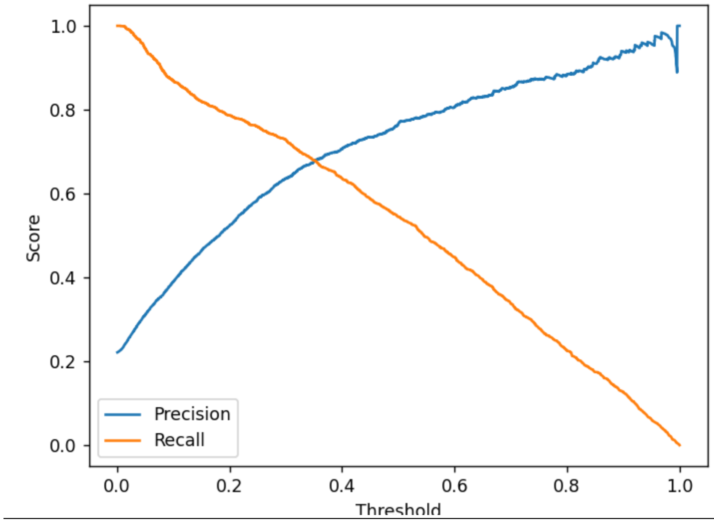


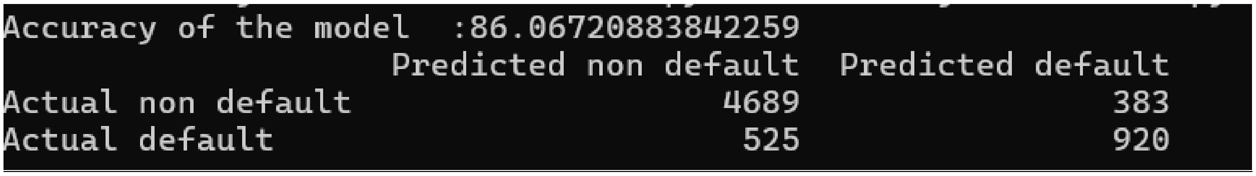
Figure 8: ROC curve

The figure above shows how precision and recall change across different classification thresholds. The point where both precision and recall remain relatively stable occurs at a threshold of approximately 0.4. At thresholds greater than 0.4 , precision increases meaning the model will accurately measure the number of clients who have defaulted and this sounds great but it also reduces recall which is not a good idea in credit risk analysis.

Because recall is the priority in this model , the focus shifts to where recall is the highest , this occurs at a threshold of approximately 0.30 where the model correctly identifies a larger proportion of true defaulters. Think of it like this : precision asks" If my model said 1000 people defaulted how many of them according to the original data did default/" and recall asks "out of the actual people who defaulted , how many did my model predict accurately?" and the latter is the most important part because the goal is to catch all defaulters.

Confusion matrix & Accuracy:

Threshold >=0.4



```
Accuracy of the model  :86.06720883842259
                    Predicted non default  Predicted default
Actual non default                   4689                383
Actual default                        525                920
```

Figure 10: Confusion matrix and accuracy at a threshold of at least 0.4

True positives =920 , clients who actually defaulted and were correctly predicted.

False negatives=525, clients who defaulted but were incorrectly predicted as non defaulters.

False positives = 383 , clients predicted to default but who actually did not.

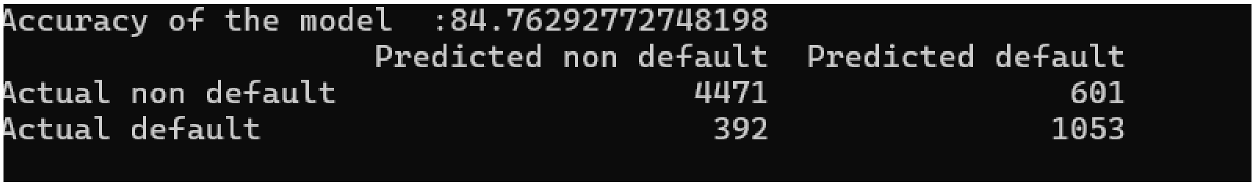
True negatives= 4689 , clients who did not default and were correctly predicted.

At a threshold of at least 0.4 we have a recall of approximately 64 %.

We have a precision of approximately 71%.

Overall accuracy 86.07%

Threshold >= 0.3



```
Accuracy of the model  :84.76292772748198
                    Predicted non default  Predicted default
Actual non default                   4471                601
Actual default                        392               1053
```

Figure 9: Confusion matrix and accuracy at a threshold of at least 0.3

True positives =1053clients who actually defaulted and were correctly predicted.

False negatives=392 clients who defaulted but were incorrectly predicted as non defaulters.

False positives = 601 , clients predicted to default but who actually did not.

True negatives= 4471 , clients who did not default and were correctly predicted.

At a threshold of at least 0.3 we have a recall of approximately 73 %.

We have a precision of approximately 64%.

Overall accuracy 84.76%

The overall accuracy of the model dips by approximately 2 % but we achiever a higher recall at a threshold of at least 0.3.The confusion matrix confirms that the model is effective at identifying risky clients while keeping false negatives relatively low.

Conclusion

Overall , the report shows that loan defaults are influenced by factors such as loan amount, personal income etc. but the model highlights that loan grade and loan percent income are the strongest predictors of default, with other variables playing smaller but still relevant roles. By prioritizing recall in the model , most high risk clients are correctly identified, allowing the company to make informed decisions.

REFERENCES

1. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.geeksforgeeks.org/machine-learning/implementation-of-lasso-regression-from-scratch-using-python/&ved=2ahUKEwjk86jh45CRAxXedUEAHUppGywQFnoECCgQAQ&usg=AOvVaw2ttNH-_d0LKBXAAJjp-AVH
2. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression&ved=2ahUKEwjk86jh45CRAxXedUEAHUppGywQFnoECCUQAQ&usg=AOvVaw2lCg5znysZ8SRapzn7NJKt
3. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.datacamp.com/tutorial/one-hot-encoding-python-tutorial&ved=2ahUKEwji46_945CRAxXHQUEAHTCeKjYQFnoECBwQAQ&usg=AOvVaw1Y2sLwLLkcpwLA4e9v_CIP
4. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.kaggle.com/datasets&ved=2ahUKEwie4pGO5JCRAxV4VkEAHb6WAVoQFnoECA0QAQ&usg=AOvVaw1GgLsCxInmUeOry-vCnO9A

APPENDIX A: CODE FOR THE MODEL

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix , classification_report,accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder


#LOADING THE DATA
data_directory = "F:\Project\credit_risk_dataset.csv\credit_risk_dataset.csv"
data=pd.read_csv(data_directory)
data_frame= pd.DataFrame(data)

#CONVERTING CATEGORICAL TO NUMERICAL VARIABLES
numerical=[
'person_age','person_income','person_emp_length','loan_amnt','loan_int_rate','loan_perce
nt_income','cb_person_cred_hist_length']
categorical=
['person_home_ownership','loan_intent','loan_grade','cb_person_default_on_file']

#DESCRIPRIVE ANALYSIS
'''for var in numerical:
    average= data_frame.groupby('loan_status')[var].mean()
    plt.figure(figsize=(5,5))
    plt.bar(['Non-default','Default'],average)
    plt.ylabel(f'Average {var}')
    plt.xlabel('Loan status')
    plt.title(f'Average {var} by loan status')
    plt.show()
'''


'''Now we use one hot encoding for converting'''
encoder= OneHotEncoder(sparse_output=False)
one_hot_encoded_data=encoder.fit_transform(data_frame[categorical])
one_hot_encoded_data_frame=pd.DataFrame(
                        one_hot_encoded_data,
                columns=encoder.get_feature_names_out(categorical))

data_frame['person_emp_length']=data_frame['person_emp_length'].fillna(data_frame['person_emp_length'].median())
data_frame['loan_int_rate']=data_frame['loan_int_rate'].fillna(data_frame['loan_int_rate'].mean())
'''Then we scale the numeric ones'''
scaler= StandardScaler()
scaled_data=scaler.fit_transform(data_frame[numerical])
scaled_data_frame=pd.DataFrame(
           scaled_data,
           columns=numerical
          )

#Preparing data for the model
X= pd.concat([one_hot_encoded_data_frame,scaled_data_frame],axis=1)
Y=data_frame['loan_status']
X_train,X_test,Y_train,Y_test= train_test_split(X,Y,test_size=0.2,random_state=42)

#The model
model_obj=LogisticRegression(penalty='l1',solver='liblinear',C=1.0,random_state=42)
model_obj.fit(X_train,Y_train)

#The results
coefficients=pd.Series(model_obj.coef_[0],
              index=X_train.columns
            )
coefficients_dataframe=pd.DataFrame({
            'Coefficient':coefficients,
            'Absolute value': coefficients.abs(),
            'Relationship':coefficients.apply(lambda x: 'Increases default'if x > 0 else ('Decreases default'))
          })
#print(coefficients_dataframe)
threshold=0.3
y_pred_new=(model_obj.predict_proba (X_test)[:,1]>=threshold).astype(int)

#Perfomance metrics
#y_pred = model_obj.predict(X_test)
accuracy=accuracy_score(Y_test,y_pred_new)
```

```
print("Accuracy of the model  :" + str (accuracy*100))

'''Get confusion matrix to see where the model messed up '''
confusion_matrix_c= confusion_matrix(Y_test,y_pred_new, labels=[0,1])
confusion_matrix_decorated=pd.DataFrame(confusion_matrix_c,
                                        index=['Actual non default','Actual default'],
                                        columns=['Predicted non default', 'Predicted default'])
print(confusion_matrix_decorated)
```