LOAN DEFAULT RISK ANALYSIS AND PREDICTION USING BANK LOAN DATASET

Balqis Nur Baity Oka Widani

Portfolio

2025

1
2
3
4
5
6
7
8
9

Loan Default Risk Analysis and Prediction using Bank Loan Dataset

This analysis aims to detect default risks and predict key features affecting loan decisions. I used Google Colab to perform the analysis. You can view the full analysis by clicking the google colab logo below.

1

2

3

4

5

6

7

8

9

Dataset Overview

This dataset is sourced from **the Loan Default Dataset**. It contains 33 column categories with range index 148670 entries. I focused on six feature to predict default risk : **Credit Score, Loan Amount, Rate of Interest, Income, LTV (Loan to Value), and DTIR1 (Debt to Income Ratio Ver 1).**

These six features were analyzed in comparison to the target variable, ‘Status’, where default is represented as 1.

1
2
3
4
5
6
7
8
9

Data Processing

Tools and Library

Google Colab

Phyton Progamming Language

Pandas Library

Numpy Library

Scikit-Learn Library

Matplotlib Library

Seaborn Library

Scipy Library

1
2
3
4
5
6
7
8
9

Data Cleaning

For data cleaning, I ensured that :

- Column names had no spaces and were converted to lowercase.

```
[187] df.columns = df.columns.str.strip().str.replace(' ', '_').str.lower()
```

- Checked for duplicate entries.

```
0s [188] df.duplicated().sum()

np.int64(0)
```

- Handled missing values.

```
0s [190] df.isna().sum() / len(df) * 100
```

Before cleaning

	0
id	0.000000
year	0.000000
loan_limit	2.249277
gender	0.000000
approv_in_adv	0.610749
loan_type	0.000000
loan_purpose	0.090133
credit_worthiness	0.000000
open_credit	0.000000
business_or_commercial	0.000000
loan_amount	0.000000
rate_of_interest	24.509989
interest_rate_spread	24.644515
upfront_charges	26.664425
term	0.027578
neg_ammortization	0.081388
interest_only	0.000000

After cleaning

	0
id	0.0
year	0.0
loan_limit	0.0
gender	0.0
approv_in_adv	0.0
loan_type	0.0
loan_purpose	0.0
credit_worthiness	0.0
open_credit	0.0
business_or_commercial	0.0
loan_amount	0.0
rate_of_interest	0.0
interest_rate_spread	0.0
upfront_charges	0.0
term	0.0
neg_ammortization	0.0
interest_only	0.0

1
2
3
4
5
6
7
8
9

EDA (Exploratory Data Analysis)

To understand the data and detect potential risks, I performed the following steps:

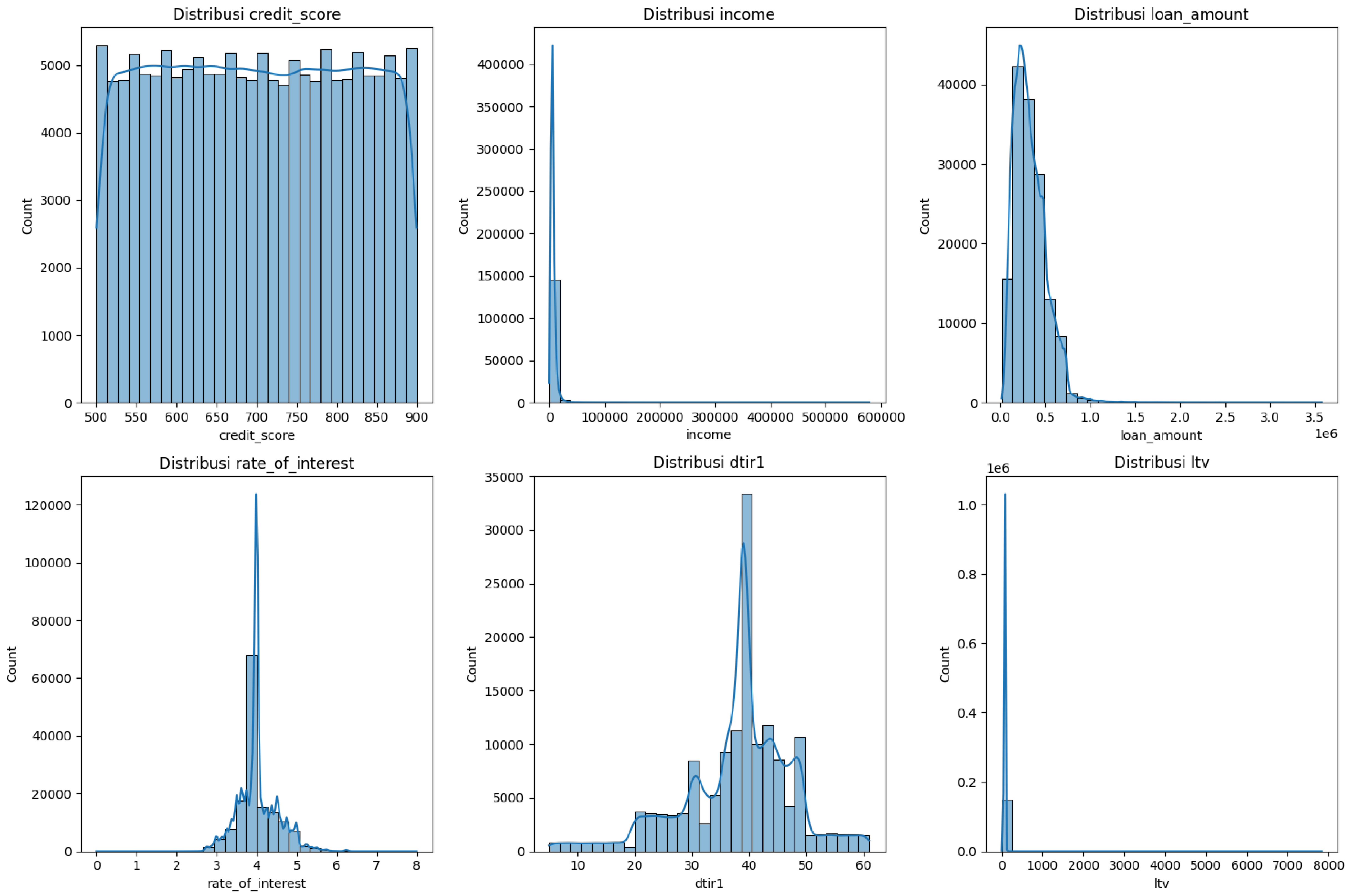
Distribution Analysis: I explored the distribution of key numerical variables such as credit score, income, loan amount, rate of interest, LTV, and DTIR1 using histograms. This helps identify the general spread and patterns in the data.

```
[206] num_cols = ['credit_score', 'income', 'loan_amount', 'rate_of_interest', 'dtir1', 'ltv']

      plt.figure(figsize=(15, 10))
      for i, col in enumerate(num_cols):
          plt.subplot(2, 3, i + 1)
          sns.histplot(df[col], bins=30, kde=True)
          plt.title(f'Distribusi {col}')
      plt.tight_layout()
      plt.show()
```

1
2
3
4
5
6
7
8
9

Result:



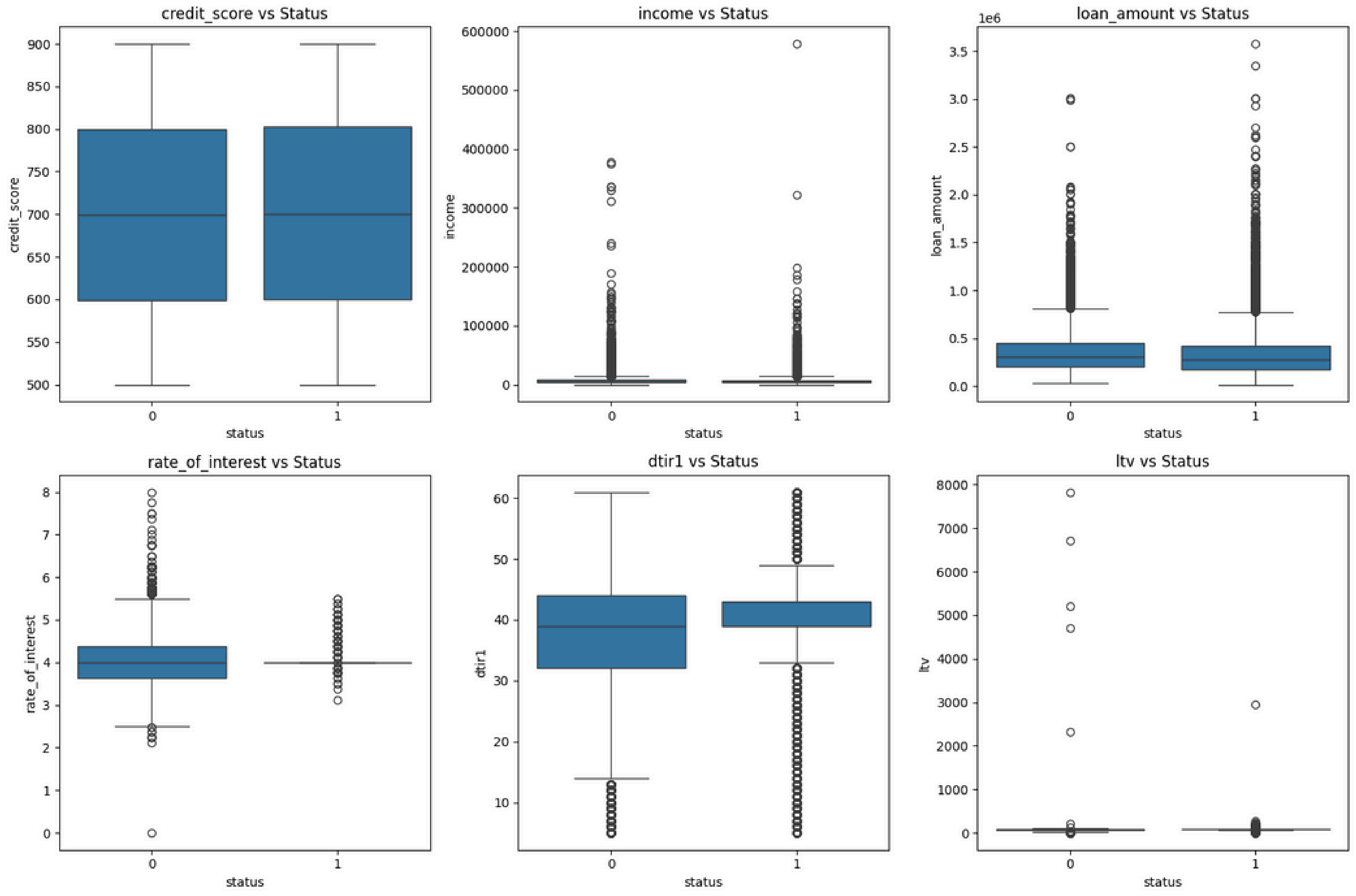
1
2
3
4
5
6
7
8
9

- **Credit Score** mostly falls between 500 and 900, indicating borrowers generally have moderate to good credit history.
- **Income** is concentrated in the lower to middle range, suggesting most applicants are from mid-income groups.
- **Loan Amount** values are mostly below 500,000, meaning smaller loan requests are more common.
- **Rate of Interest** is centered around 4%, showing a relatively stable lending rate.
- **DTIR1 (Debt-to-Income Ratio)** mostly ranges between 30 and 50, still within a manageable risk zone.
- **LTV (Loan-to-Value)** shows high variability, with some values exceeding 1000, which may indicate potential risk and needs further attention.

Outlier & Anomaly Detection: I used boxplots to detect outliers and anomalies that could skew the analysis or indicate unusual behavior.

```
[207] num_cols = ['credit_score', 'income', 'loan_amount', 'rate_of_interest', 'dtir1', 'ltv']

      plt.figure(figsize=(15, 10))
      for i, col in enumerate(num_cols):
          plt.subplot(2, 3, i + 1)
          sns.boxplot(x='status', y=col, data=df)
          plt.title(f'{col} vs Status')
      plt.tight_layout()
      plt.show()
```



- **Credit score** appears similar between default and non-default → likely not a strong predictor.
- **Income & loan amount** are higher in default group → potential risk factors.
- Higher **interest rates, DTIR, and LTV** are more common in defaulters → may increase default risk.

1
2
3
4
5
6
7
8
9

Target-Based Analysis: I conducted a grouped comparison using groupby('status').mean() to understand how each feature behaves across default (1) and non-default (0) categories.

```
[209] df.groupby('status')[['credit_score', 'income', 'loan_amount', 'rate_of_interest', 'dtir1', 'ltv']].mean()
```

	credit_score	income	loan_amount	rate_of_interest	dtir1	ltv
status						
0	699.523793	7102.046041	334990.774875	4.044931	37.482965	72.064812
1	700.600344	6215.851961	319275.184912	3.991968	39.331423	75.815339

Groupby output

- **Credit Score**: Not a strong predictor of default.
- **Income, DTIR1, and LTV**: Higher values are associated with increased default risk.
- **Loan Amount**: A potential risk factor for default.
- **Rate of Interest**: Appears less significant based on mean comparison.

1 2 3 4 5 6 7 8 9

Hypothesis Testing

A t-test was used to evaluate which features are statistically significant in influencing default risk.

```
def ttest(column):
  status_0 = df[df['status'] == 0][column]
  status_1 = df[df['status'] == 1][column]

  stat, p = stats.ttest_ind(status_0, status_1)
  print(f'Processing column: {column}')
  print(f't-statistic: {stat}')
  print(f'p-value: {p}')

  if p < 0.05:
    print(f'{column} is significant to default')
  else:
    print(f'{column} is not significant to default')
  print()
  return status_0, status_1
```

```
[212] for col in num_cols:
        result = ttest(col)
```

```
Processing column: credit_score
t-statistic: -1.5437361122829274
p-value: 0.12265440254169067
credit_score is not significant to default

Processing column: income
t-statistic: 23.415899383258807
p-value: 4.882969759627964e-121
income is significant to default

Processing column: loan_amount
t-statistic: 14.208539142794285
p-value: 8.69062767980642e-46
loan_amount is significant to default

Processing column: rate_of_interest
t-statistic: 18.040546368030164
p-value: 1.1195908811448796e-72
rate_of_interest is significant to default

Processing column: dtir1
t-statistic: -31.892407549643437
p-value: 1.9292349024820293e-222
dtir1 is significant to default

Processing column: ltv
t-statistic: -16.462003778070088
p-value: 7.788581645278733e-61
ltv is significant to default
```

T-test results show that only Credit Score is not statistically significant to default, which aligns with the previous analysis.
All other features tested — Income, Loan Amount, Rate of Interest, DTIR1, and LTV — are statistically significant and contribute to the potential of default.
The T-test results are consistent with the boxplot analysis.

1
2
3
4
5
6
7
8
9

Modeling

For this dataset, I used Random Forest as the modeling algorithm. The ROC AUC Score is 98%, indicating that the model has excellent predictive accuracy. The Random Forest model successfully distinguishes between default and non-default cases, showing that the risk of default is generally low in the dataset based on the selected features.

```
[213] rf_model = RandomForestClassifier(random_state=42)
      rf_model.fit(X_train, y_train)

      y_pred_rf = rf_model.predict(X_test)
      y_proba_rf = rf_model.predict_proba(X_test)[:, 1]

      print(confusion_matrix(y_test, y_pred_rf))
      print(classification_report(y_test, y_pred_rf))
      print(f"ROC AUC Score: {roc_auc_score(y_test, y_proba_rf):.4f}")
```

```
[[21103  1391]
 [ 1216  6024]]
              precision    recall  f1-score   support

           0       0.95      0.94      0.94     22494
           1       0.81      0.83      0.82      7240

    accuracy                           0.91     29734
   macro avg       0.88      0.89      0.88     29734
weighted avg       0.91      0.91      0.91     29734

ROC AUC Score: 0.9758
```

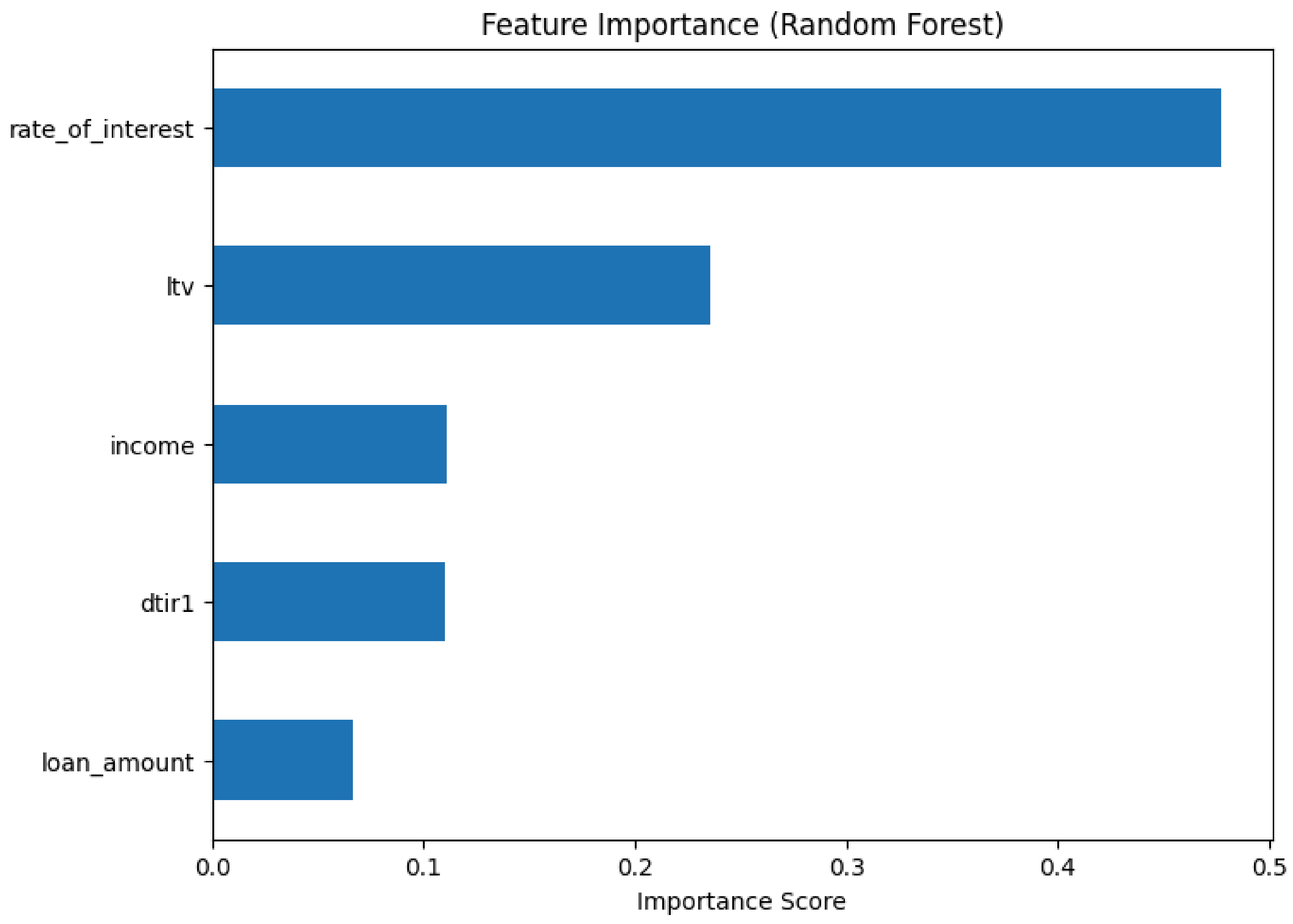
Features Importance

1
2
3
4
5
6
7
8
9

Feature importance shows that Rate of Interest is the most influential feature (≈ 0.5), indicating a strong impact on predicting default.

Meanwhile, Loan Amount has the lowest importance (< 0.1), suggesting less influence in the model.

```
[214] feature_importance = pd.Series(rf_model.feature_importances_, index=features)
      feature_importance.sort_values().plot(kind='barh', figsize=(8,6))
      plt.title("Feature Importance (Random Forest)")
      plt.xlabel("Importance Score")
      plt.show()
```



Summary

- **Overall Risk Profile :** Although features like Income, Loan Amount, Rate of Interest, DTIR1, and LTV show statistical significance in relation to default, the model predicts that the majority of borrowers fall into a low-risk category, indicating well-managed credit distribution.
- **Most Influential Factor** : Rate of Interest holds the highest feature importance, suggesting that higher interest rates increase the possibility of default. Interest rate setting should be aligned with borrower risk profiles.
- **Operational Recommendation :** The verification or credit approval team should apply stricter assessments for applicants with high loan-to-income ratios or LTV scores. These borrowers should be closely evaluated or offered limited credit.

Summary

- **Loan Structuring Strategy :** Offering longer tenures and fixed interest options can help at-risk borrowers maintain consistent payments and avoid default, especially those with tight DTIR margins.
- **Potential for Risk Tiering :** The Random Forest model enables borrower segmentation by default risk, which can support risk-based pricing, tailored monitoring, and early intervention strategies.

Thank you

Contact Details

Phone : *+6281216876268*

Github : [*https://github.com/balqisn*]

LinkedIn : *https://www.linkedin.com/in/balqis nurbaityokawidani/*

Email : *balqis.1542@gmail.com*