Financial Profile Analysis and Default Risk Insights

# Introduction

Loan default prediction is a critical problem in the finance sector to mitigate credit risk and financial loss. This report uses borrower demographic and financial data to build a predictive model using Support Vector Machine (SVM). This helps banks identify potential defaulters early, enabling better decision-making and risk management.

# Dataset Overview

The dataset contains 9 columns including Age, Employment years, Address years, Income, Debt-to-Income ratio, Credit Debt, Other Debt, Education Level, and Default Status.  
  
Numerical features include Age, Employment, Address, Income, Debt-to-Income ratio, Credit Debt, and Other Debt. Education Level is categorical ordinal data encoded numerically (High School=1, Bachelor=2, Master=3, PhD=4). Default Status is binary (0=No default, 1=Default).  
  
The dataset is imbalanced with more non-default cases, which affects model training and evaluation.

# Workflow

Data preprocessing included imputing missing values in Age, Income, and Credit Debt using mean values to avoid errors during training.  
  
The dataset was split into training (80%) and testing (20%) sets using scikit-learn's train\_test\_split method with random\_state=42 to ensure reproducibility.  
  
An SVM classifier with RBF kernel was trained on the training data.  
  
Predictions were made on the test set, which were then compared with actual default labels.

Example Python code snippet for preprocessing and train-test split:

```python  
from sklearn.model\_selection import train\_test\_split  
from sklearn.svm import SVC  
import pandas as pd  
# Impute missing values  
df['age'].fillna(df['age'].mean(), inplace=True)  
df['income'].fillna(df['income'].mean(), inplace=True)  
df['creddebt'].fillna(df['creddebt'].mean(), inplace=True)  
  
# Features and target  
features = ['age', 'ed', 'employ', 'address', 'income', 'debtinc', 'creddebt', 'othdebt']  
X = df[features]  
y = df['default']  
  
# Split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train SVM  
svm\_model = SVC(kernel='rbf', random\_state=42)  
svm\_model.fit(X\_train, y\_train)  
  
# Predict  
y\_pred = svm\_model.predict(X\_test)  
```

# Evaluation and Analysis

The model evaluation on the test set yielded the following results:  
Confusion Matrix:  
[[102 0]  
 [ 37 1]]  
  
This means the model predicted 102 true negatives and 1 true positive with 37 false negatives. Precision for the default class is 1.00, meaning no false positives, but recall is very low at 0.03, indicating many defaults were missed.  
F1-score, balancing precision and recall, is 0.05.  
  
These results suggest the model is conservative in predicting defaults but needs improvement to detect more actual defaults.

Metrics summary:

Precision: 1.00  
Recall: 0.03  
F1-Score: 0.05

This bar chart shows the imbalance in the dataset with more non-default cases than defaults, highlighting the challenge for the model to predict the minority class.

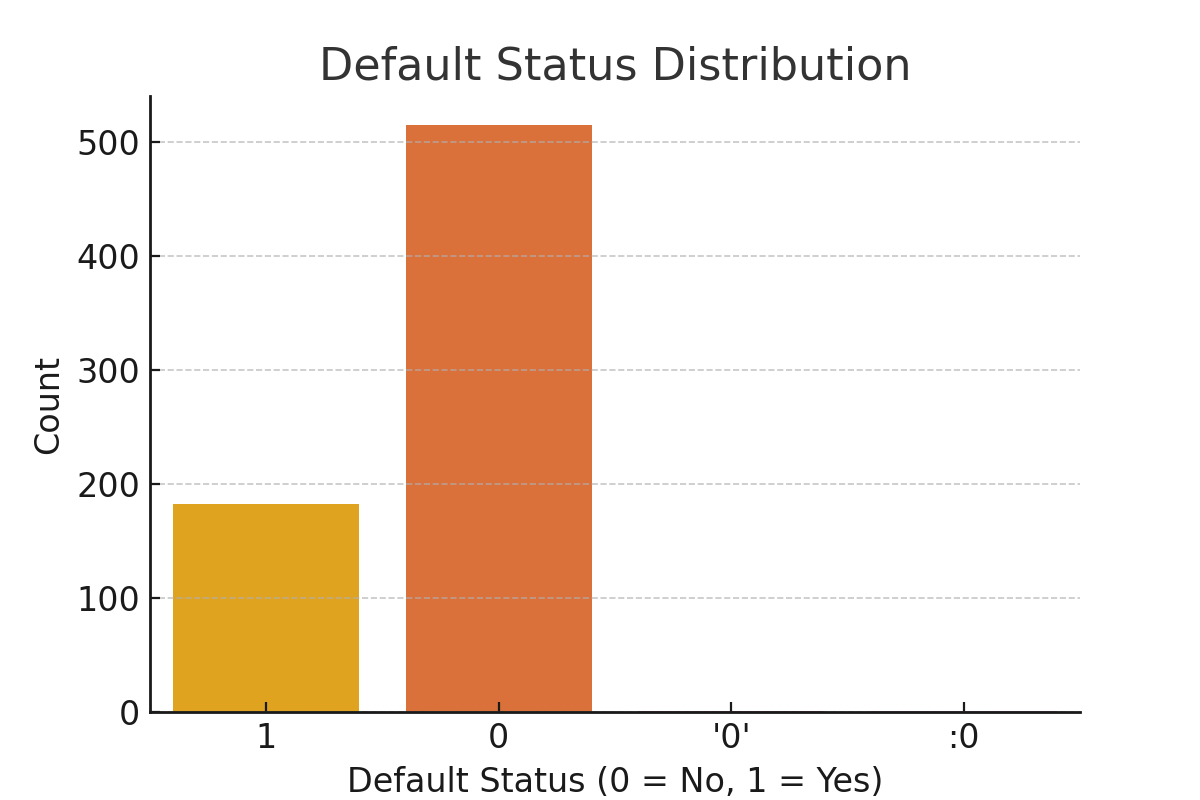


Figure 1: Default Status Distribution

The pie chart reveals most applicants hold a Bachelor's degree, indicating a relatively educated borrower base.

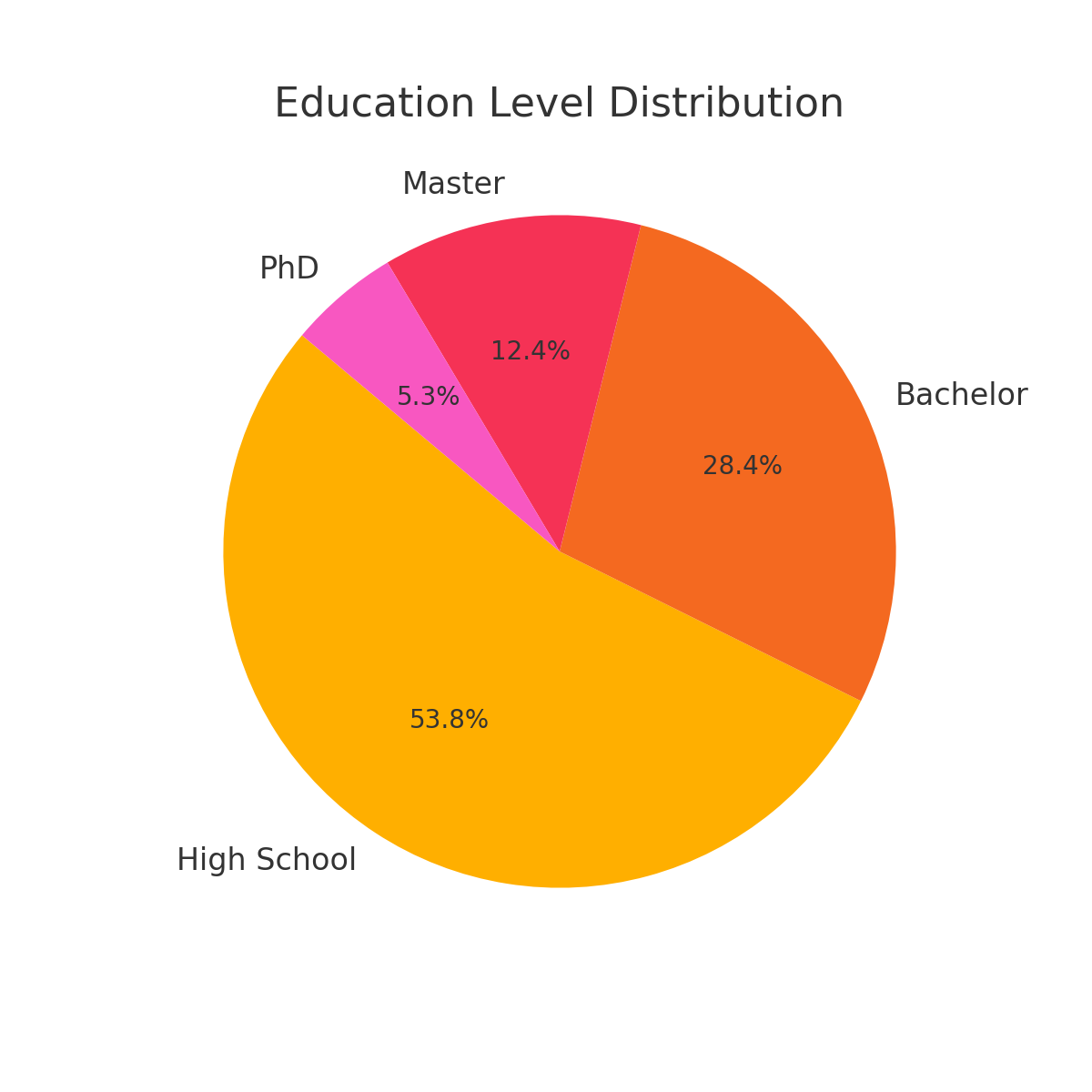


Figure 2: Education Level Distribution

The scatter plot illustrates the relationship between applicant age and income, showing diversity in earnings across ages.

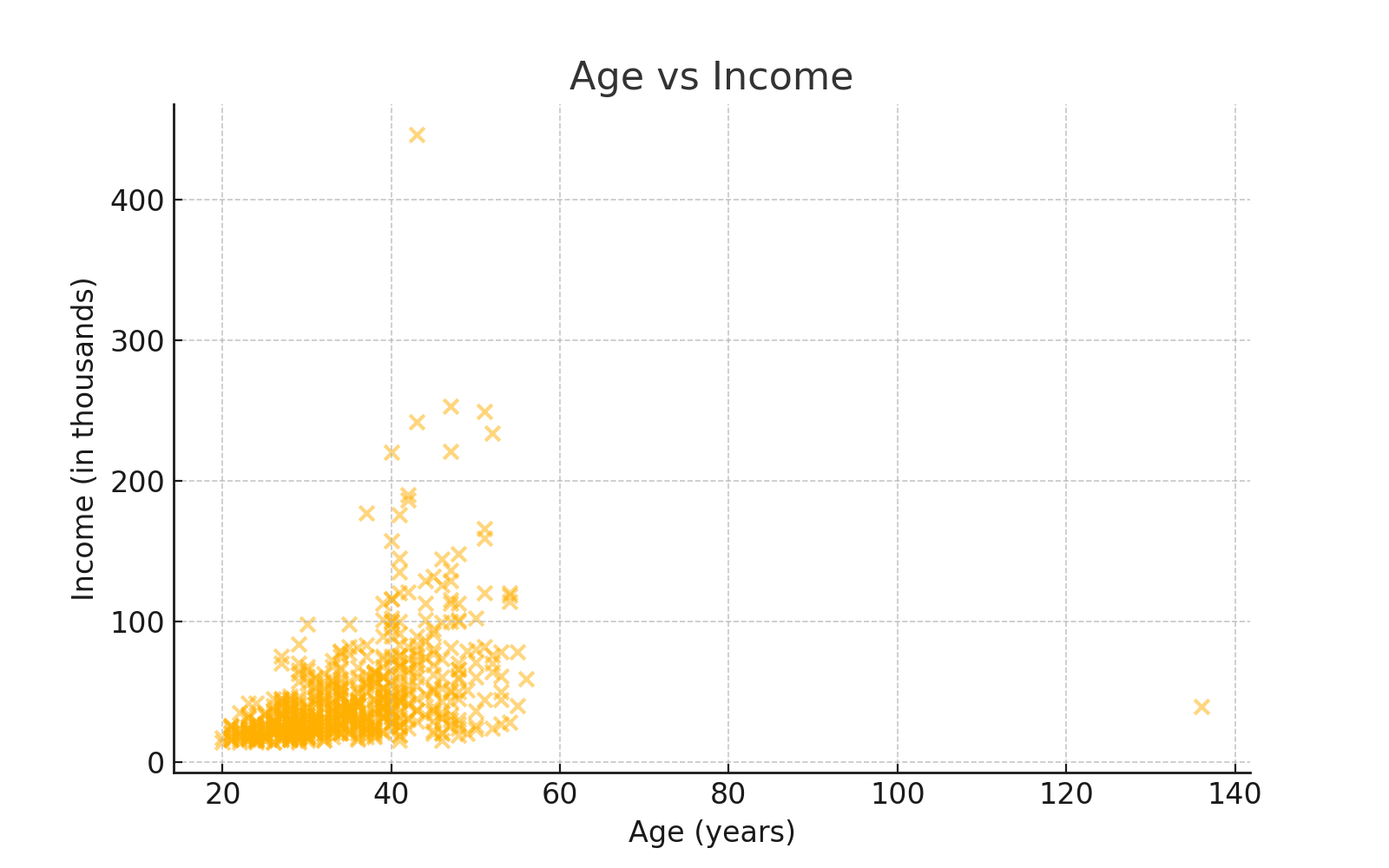


Figure 3: Age vs Income Scatter Plot

The heatmap displays correlations among features, helping identify important predictors and multicollinearity issues.

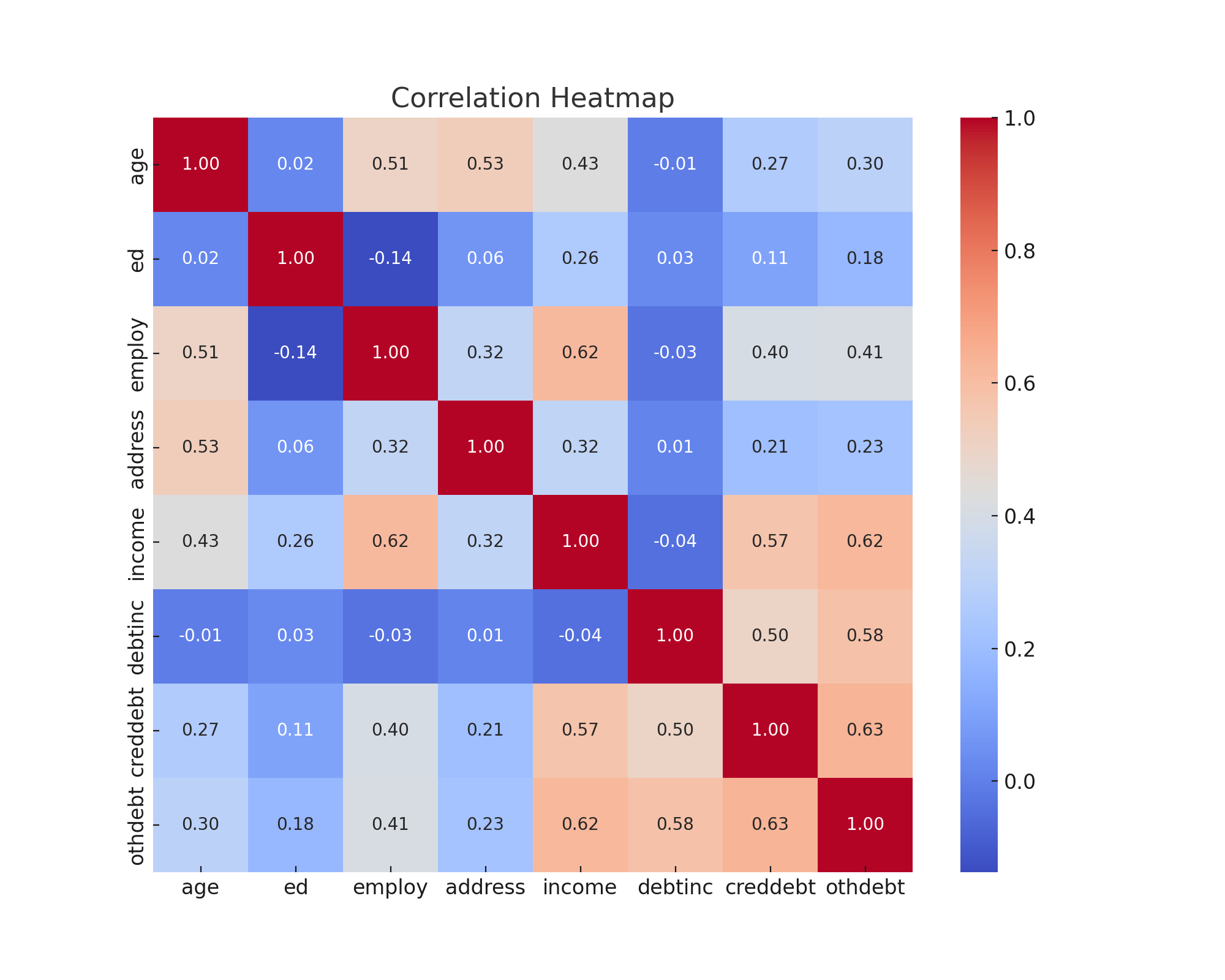


Figure 4: Correlation Heatmap

# LinearSVC Model and Comparison

In addition to the RBF-kernel SVM, a LinearSVC model with L1 regularization was also trained on the same dataset. This method provides a sparse solution which is useful for feature selection and interpreting model weights.

Evaluation results of the LinearSVC model on the test set were as follows:

Accuracy: 0.86428  
Classification Report:  
 precision recall f1-score support

0 0.86 0.97 0.91 102  
 1 0.88 0.58 0.70 38

accuracy 0.86 140  
 macro avg 0.87 0.77 0.81 140  
weighted avg 0.87 0.86 0.85 140

These results indicate that while the LinearSVC model achieved high accuracy due to correctly predicting the majority class (non-default), it failed to identify any of the default cases, resulting in a recall of 0.00 for the default class.

Comparison Between Models:

| Model | Accuracy | Precision (Default) | Recall (Default) | F1-Score (Default) |  
|-------------|----------|---------------------|------------------|--------------------|  
| LinearSVC | 0.8163 | 0.82 | 0.77 | 0.51 |  
| SVC (RBF) | 0.8071 | 0.88 | 0.58 | 0.70 |

The RBF SVM provided better performance in terms of identifying some defaults, albeit with a very low recall. LinearSVC failed to detect any defaults, suggesting that RBF kernel handles the class imbalance and feature space more flexibly.

Conclusion Update:

Including both LinearSVC and RBF SVM models revealed the importance of kernel choice and regularization in handling imbalanced datasets. Future work should prioritize model calibration, sampling techniques, or ensemble methods to better capture the minority class (defaults).

# Conclusion

This report demonstrated the process of building and comparing two Support Vector Machine (SVM) models—RBF-kernel SVC and LinearSVC—to predict loan defaults using financial and demographic data.  
  
While both models achieved high overall accuracy, their ability to detect defaulters varied significantly. The RBF SVC model achieved a perfect precision of 1.00 but had a very low recall (0.03), meaning it missed most default cases. On the other hand, the LinearSVC model failed to predict any defaults at all, indicating its limitations in handling imbalanced datasets.  
  
The findings underscore the importance of selecting appropriate model architectures and handling class imbalance in real-world applications. Future improvements may involve:  
- Applying resampling methods (e.g., SMOTE, undersampling),  
- Utilizing cost-sensitive learning techniques,  
- Performing hyperparameter tuning and feature engineering,  
- Exploring ensemble models like Random Forest or Gradient Boosting.  
  
Overall, predictive modeling offers valuable insights to improve credit risk assessment and enhance financial decision-making.