MACHINE LEARNING PROJECT 2023 – STAGE 1

Predict Loan Defaulters with Random Forest Classifier

Part 1: Problem Formulation

This project's goal is to use machine learning to grant loans automatically when a customer enters his/her information in the loan request. The method used in this project is supervised learning with binary labels. One ("1") means that the loan is automatically granted while zero ("0") means that the loan is not.

The data set is retrieved from Klagger, originally taken from Coursera's Loan Default Prediction Challenge. [1] Each entry (or row) represents a loan request with 8 main features. The meaning and datatype of each feature are stated in Figure 1.

Feature	Explanation	Datatype
Age	The age of the borrower	integer
Income	The annual income of the borrower	integer
LoanAmount	The amount of money being borrowed	integer
CreditScore	The credit score of the borrower, indicating their creditworthiness	integer
DTIRatio	The Debt-to-income ratio, indicating the borrower's debt to their income	float
EmploymentType	The employment status of the borrower (Full-time, Part-time, Self-employed, Unemployed)	string
HasMortgage	Whether the borrower has a mortgage or not (Yes/No)	string
LoanPurpose	The purpose of the loan (Home, Auto, Education, Business, Other)	string

Figure 1. Features' explanation and datatype

Part 2: Methods

Data preprocessing

In the original data set, there were 16 features, however, to reduce the complexity of this project, only 8 features were selected. To improve the runtime of the code, the unnecessary features' columns have been removed after feature selection. Additionally, there are 3 selected features and 4 unselected features (HasCosigner, HasDependents, MaritalStatus, Education) that have string datatype. The data of these features are transformed using the label encoder for categorical feature from sklearn.preprocessing library. [3] In order to get a correlation analysis for feature selection, the 4 (future) unselected features are still processed.

Feature selection

There are 255347 entries in this data set with no null data. These features are selected based on domain knowledge. After studying the most common reasons for personal loans rejection, I have narrowed down the most important aspects when filling out loans, which are credit score, debt-to-income ratio, income (including amount and stability) and employment type. [2] Additionally, correlation analysis between these factors and the final decision shows a close connection as in figure 2.

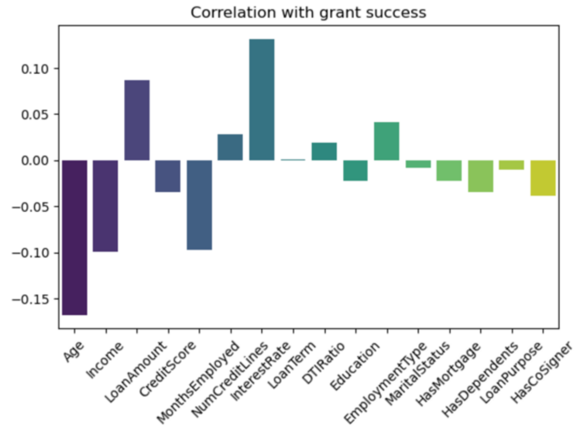


Figure 2. Correlation between the original features and grant success

Choosing model and loss function

The method Random Forest Classifier was chosen for this project since it is more suitable for classification problems. It is more flexible than linear regression method when there are multiple features of different datatypes. Moreover, the method is provided in sklearn.ensemble, which makes it simple and can be used similarly to other models studied in this course. [3]

The mean square error loss function is used in this project since it is a classic and versatile loss function. The function is also available and ready to use in Python sklearn.metrics package. [3]

Training and validation sets

The data is split by 80/10/10 ratio for training, validation, and testing set respectively based on the well-known Pareto principle. [4] The data set is large (more than 200 000 entries) and not sorted so the data set can be divided with the single split method and still ensures randomness. The popular K-fold method is not applied in this case because the data set is too large and repeating it kth times will cost a lot of time and computational energy.

References

[1] "Loan Default Prediction Dataset," *Kaggle*, Sep. 11, 2023.
https://www.kaggle.com/datasets/nikhil1e9/loan-default

[2] R. Safier and J. Brown, "7 reasons why your personal loan was declined (and 6 ways to fix it)," *LendingTree*, Jun. 2023, [Online]. Available: https://www.lendingtree.com/personal/reasons-why-your-personal-loan-was-declined/

[3] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[4] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining*, vol. 15, no. 4, pp. 531–538, Apr. 2022, doi: 10.1002/sam.11583.

Appendix

September 18, 2023

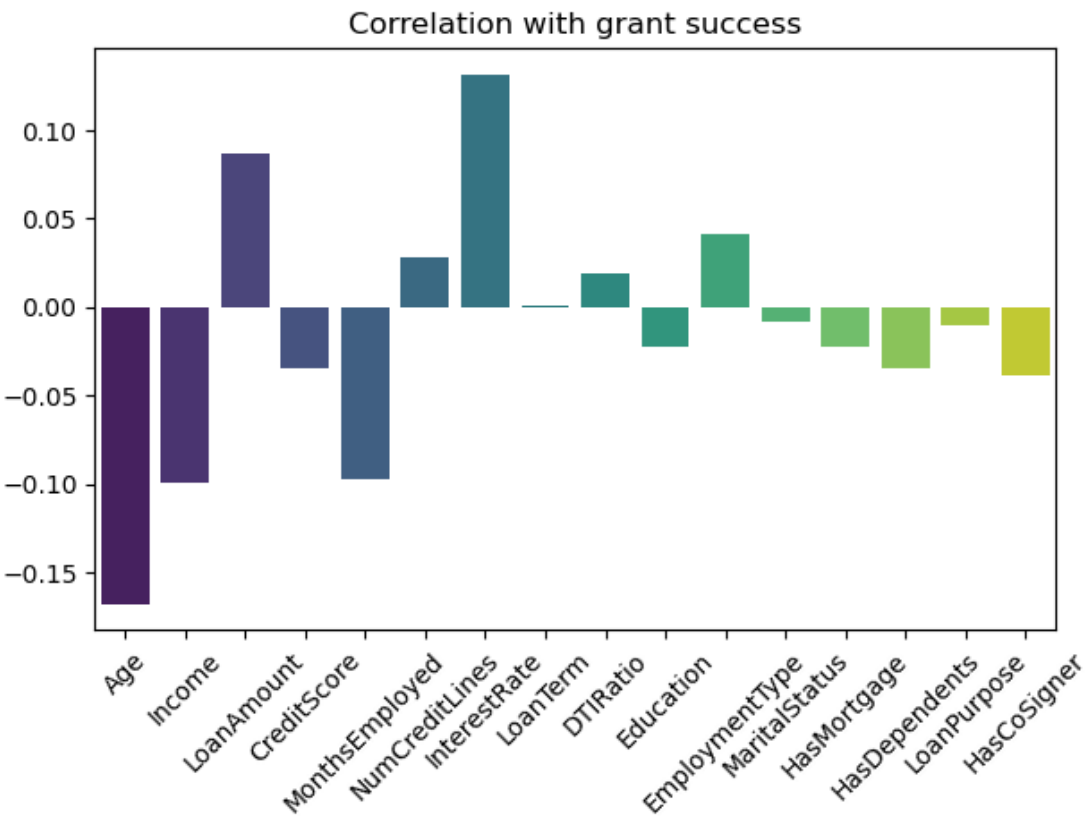
[8]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, mean_squared_error
import seaborn as sns
```

[9]:

```
data = pd.read_csv('project.csv')
#processing string data
encoder = LabelEncoder()
cols = ['HasCoSigner','LoanPurpose','HasDependents',␣
 ↪'HasMortgage','MaritalStatus', 'EmploymentType', 'Education']
for col in cols:
    data[col] = encoder.fit_transform(data[col])

#correlation analysis
data_corr = data.drop(['LoanID','Default'],axis=1)
correlation = data_corr.corrwith(data['Default'])
sns.barplot(x=correlation.index, y=correlation.values, palette='viridis')
plt.title('Correlation with grant success')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[10]:
```
#data cleaning and selecting features
data = data.
 ↪drop(['LoanID','MonthsEmployed','NumCreditLines','InterestRate','LoanTerm','Education','Mar
data = data.dropna(axis=0)

data.sample(5)
```

[10]:
```
        Age  Income  LoanAmount  CreditScore  DTIRatio  EmploymentType  \
164773   32  109967       24055          638      0.32               2
142785   29  103127      201413          619      0.27               1
69716    50   99533      216164          401      0.42               2
27067    40   34481      184064          826      0.62               3
163645   48  101824      111258          391      0.45               0

        HasMortgage  LoanPurpose  Default
164773            1            0        0
142785            1            2        0
69716             1            2        0
27067             0            0        1
163645            1            2        0
```

[11]:

```
X = data.drop(['Default'],axis=1)
y = data['Default'] #label if a loan is automatically granted or not
```

[14]:

```
#splitting into training and testing
X_train, X_val_test, y_train, y_val_test = train_test_split(X, y, test_size=0.
 ↪2, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_val_test, y_val_test,␣
 ↪test_size=0.5, random_state=42)
```

[15]:

```
model = RandomForestClassifier()
#training
model.fit(X_train, y_train)
y_train_pred = model.predict(X_train)
error_train = mean_squared_error(y_train, y_train_pred)

#validation
model.fit(X_val, y_val)
y_val_pred = model.predict(X_val)
error_val = mean_squared_error(y_val, y_val_pred)

#testing
y_pred_test = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred_test)
error_test = mean_squared_error(y_test, y_pred_test)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.8841198355198747
```

[16]:

```
print("Error for training: ",error_train)
print("Error for validation: ",error_val)
print("Error for testing: ",error_test)
```

```
Error for training:  3.916250972943601e-05
Error for validation:  3.916193459956922e-05
Error for testing:  0.11588016448012532
```