EMEM539 - Final Project

NAZARBAYEV
UNIVERSITY

Credit Card Default Prediction & Analysis

Adil Kaskyrbayev
Yerzhan Mukhanov

Meet Our Team

Yerzhan Mukhanov

BSc in Computer Science

Adil Kaskyrbayev

BSc in Computer Science

Agenda

01 INTRODUCTION AND OBJECTIVES
02 DATA OVERVIEW
03 DATA PREPROCESSING
04 METHODS
05 METRICS
06 RESULTS AND INSIGHTS
07 CONCLUSION

Introduction and Objectives

Introduction

How machine learning models can predict credit card defaults, enabling financial institutions to proactively manage credit risk and optimize lending strategies

Objective

Predict likelihood of credit card payment default

Importance

Banks:
Early identification of risky customers
2022y 2.5% World Bank

Impact:
Manage credit risk, optimize lending strategies

Dataset Overview

Source

kaggle.com

Dataset Summary

- 30,000 records of credit card customers
- Features: 25 variables (gender, payment behavior)
- Target: default.payment.next.month (1: Default, 0: No Default) - renamed to DPNM

```python
df = pd.read_csv('/kaggle/input/default-of-credit-card-clients-dataset/UCI_Credit_Card.csv', delimiter=',')
df.dataframeName = 'UCI_Credit_Card.csv'
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   30000 non-null  float64
 2   SEX                         30000 non-null  int64
 3   EDUCATION                   30000 non-null  int64
 4   MARRIAGE                    30000 non-null  int64
 5   AGE                         30000 non-null  int64
 6   PAY_0                       30000 non-null  int64
 7   PAY_2                       30000 non-null  int64
 8   PAY_3                       30000 non-null  int64
 9   PAY_4                       30000 non-null  int64
 10  PAY_5                       30000 non-null  int64
```

Figure 1. Dataset info

Dataset Overview

Features:

- **SEX**: Gender
- **EDUCATION**
- **MARRIAGE**
- **AGE**
- **LIMIT_BAL**: Amount of given credit in dollars
- **PAY_i**: Repayment status started from September, 2005
- **BILL_AMTi**: Amount of bill statement in September and on, 2005 (NT dollar)
- **PAY_AMTi**: Amount of previous payment in September etc, 2005 (NT dollar)

Target value

- default.payment.next.month – Default payment (1=yes, 0=no)

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4
0	20000.0	2	2	1	24	2	2	-1	-1
1	120000.0	2	2	2	26	-1	2	0	0
2	90000.0	2	2	2	34	0	0	0	0
3	50000.0	2	2	1	37	0	0	0	0
4	50000.0	1	2	1	57	-1	0	-1	0

5 rows × 24 columns

Figure 2. Head of the dataset

Data Preprocessing

Feature Transformation

- Normalization and Scaling (LIMIT_BAL,BILL_AMTi, PAY_AMTi)
- Balancing columns (Under-sampling:)
- Renaming columns ('PAY_0':'PAY_1')
- No need in unique()

Encoding Categorical Variables

- The data is already categorised into integer **(-2, -1, 0, 1 etc)**, so we do not have to do this part

Feature Selection

- Correlation matrix analysis
- Dropped not important features, such as ID
- Not NULL check

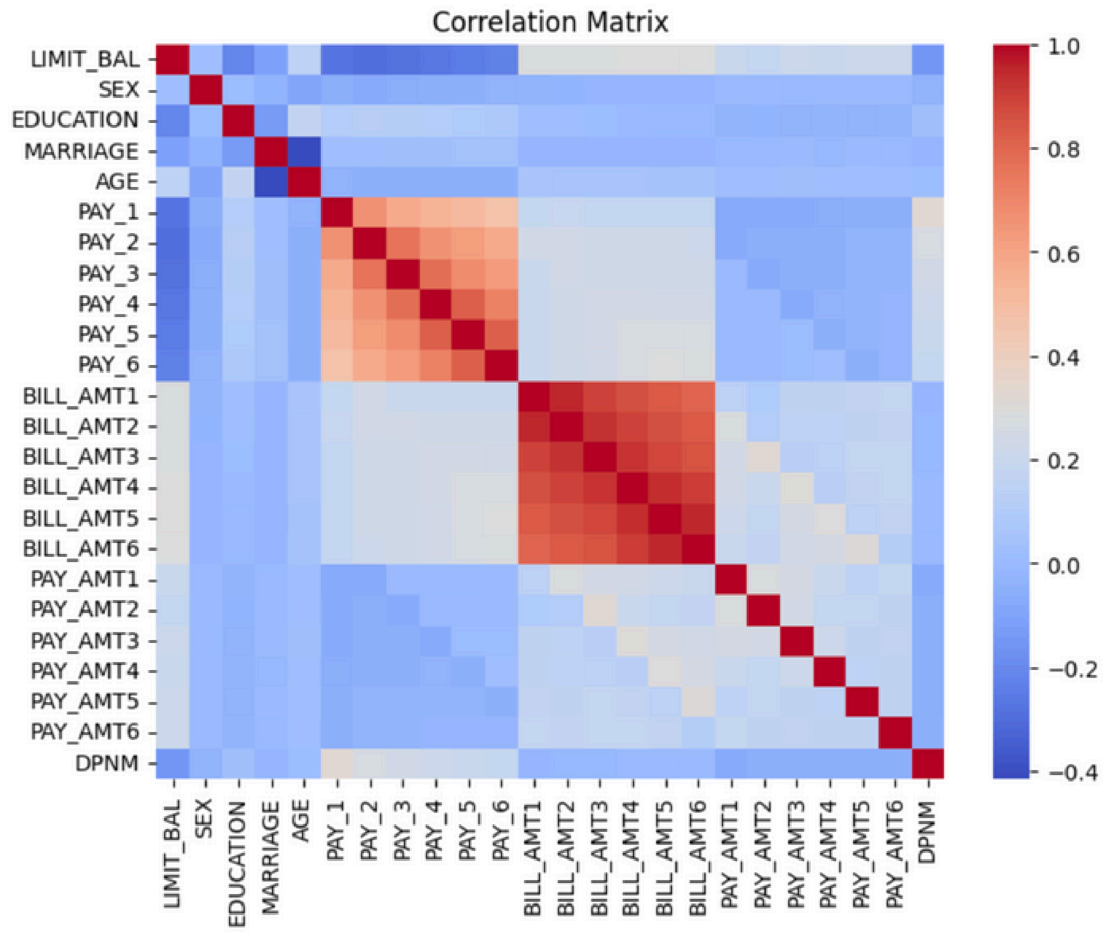


Figure 3. Correlation matrix

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4
0	-1.136720	2	2	1	24	2	2	-1	-1	-2	...	-0.672497
1	-0.365981	2	2	2	26	-1	2	0	0	0	...	-0.621636
2	-0.751350	1	2	2	30	1	2	2	0	0	...	0.365590
3	-1.136720	1	1	2	24	0	0	2	2	2	...	-0.387444
4	-0.365981	2	2	1	39	-1	-1	-1	-1	-1	...	-0.672497

Figure 3. Head of the processed dataset

Data Preprocessing

Feature Selection

- Besides previously discussed methods, we illustrated dependency of several parameters.

1 = married; 2 = single; 3 = divorce; 0=others)

```
[22]:
g = sns.FacetGrid(df, row='DPNM', col='MARRIAGE')
g = g.map(plt.hist, 'AGE')
plt.show()
```

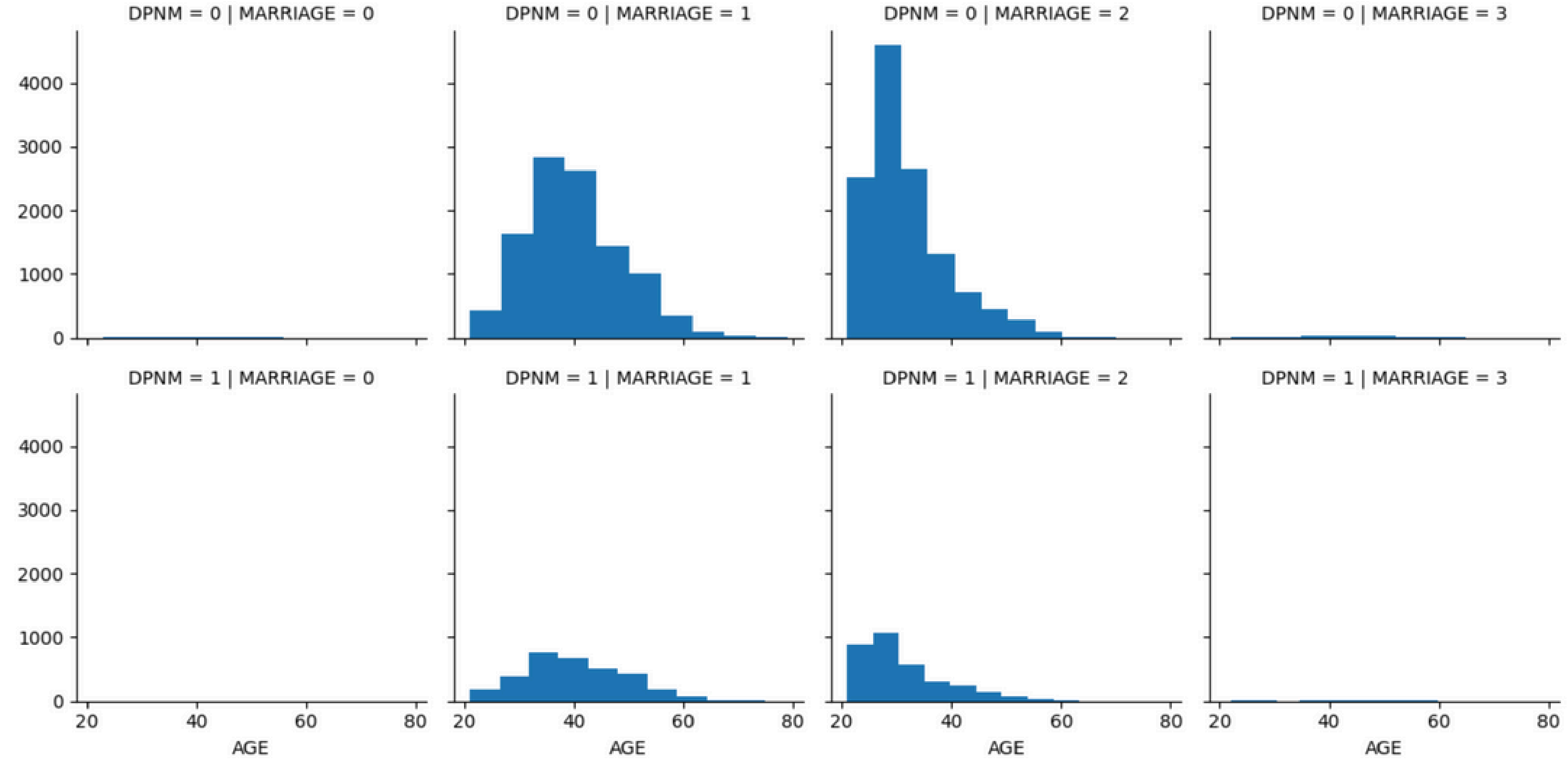


Figure 4. Marriage status and Age dependency

SEX: Gender (1=male, 2=female)

```
[23]:
g = sns.FacetGrid(df, row='DPNM', col='SEX')
g = g.map(plt.hist, 'AGE')
```

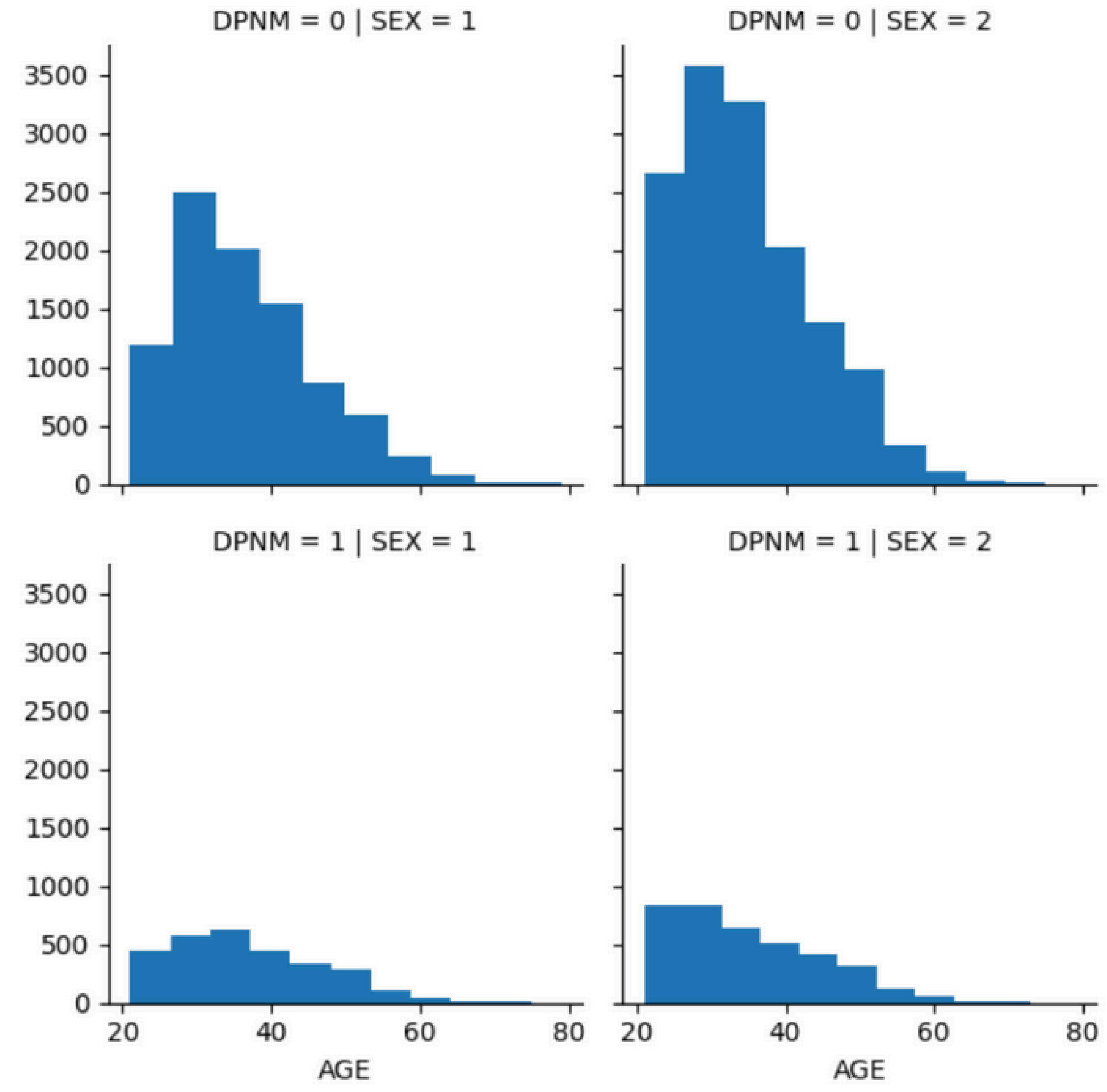


Figure 5. Gender and Age dependency

Data Preprocessing

The overall probability of default

- raw data - **23364 to 6636 = 78:22**
- under-sampling to reach **70:30**

```
Class distribution before under-sampling: Counter({1: 23364, 0: 6636})
Class distribution after under-sampling: Counter({1: 22120, 0: 6636})
```

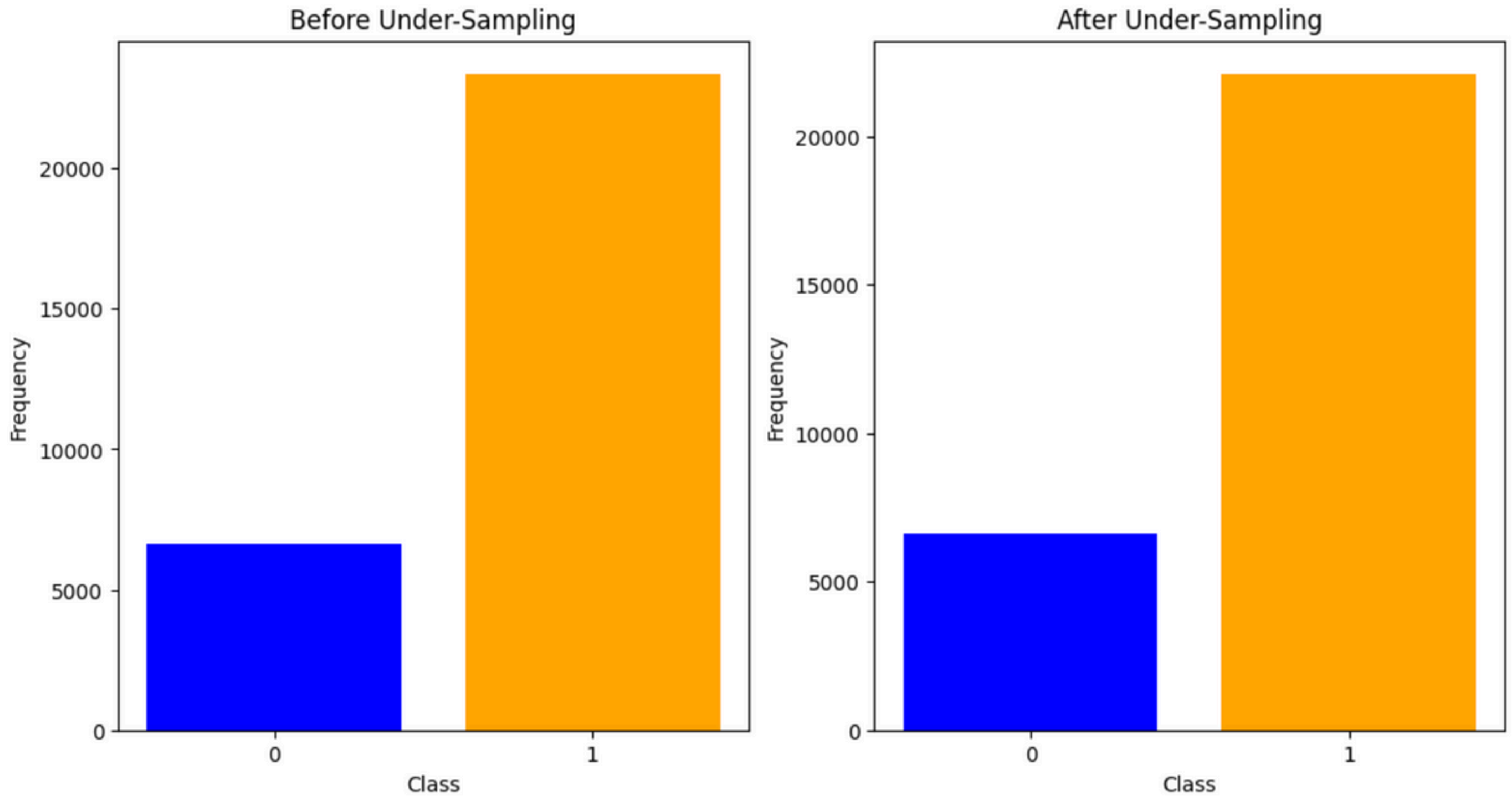


Figure 6. Overall probability of default

```
[18]: df.isnull().sum()
```

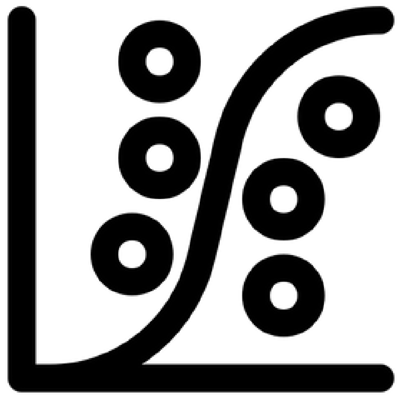
```
[18…  LIMIT_BAL                     0
      SEX                           0
      EDUCATION                     0
      MARRIAGE                      0
      AGE                           0
      PAY_1                         0
      PAY_2                         0
      PAY_3                         0
      PAY_4                         0
      PAY_5                         0
      PAY_6                         0
      BILL_AMT1                     0
      BILL_AMT2                     0
      BILL_AMT3                     0
      BILL_AMT4                     0
      BILL_AMT5                     0
      BILL_AMT6                     0
      PAY_AMT1                      0
      PAY_AMT2                      0
      PAY_AMT3                      0
      PAY_AMT4                      0
      PAY_AMT5                      0
      PAY_AMT6                      0
      default.payment.next.month    0
      dtype: int64
```

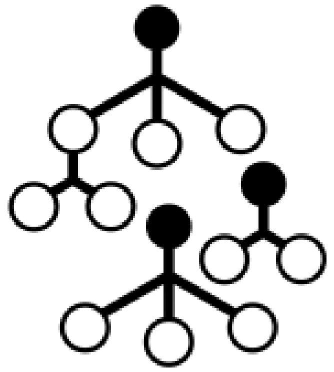
Figure 7. Not NULL check

Methods:

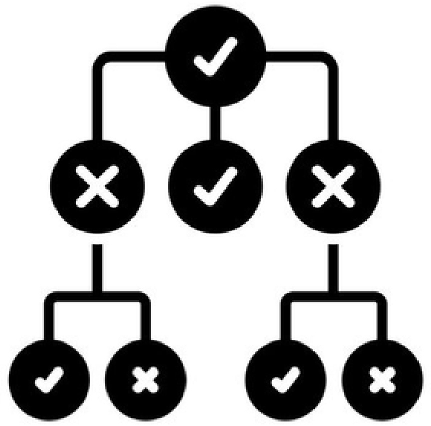
Train 80 - 20 Test

KNN → Logistic regression → Random Forest





SVC → Decision tree



Metrics

	KNN (euclidean, 7, uniform)	SVC	Logistic regression	Random Forest	Decision tree
Accuracy	0.7926	0.7916	0.8067	0.8103	0.7314
ROC AUC	0.6364	0.6303	0.7231	0.7650	0.6303
Precision	0.8304	0.7937	0.8139	0.8364	0.8303
Recall	0.9204	0.9851	0.9706	0.9367	0.8181
F1-Score	0.8731	0.8791	0.8854	0.8837	0.8241

Table 1. Results

Best in amont metrics

Best in amont methods

Metrics examples

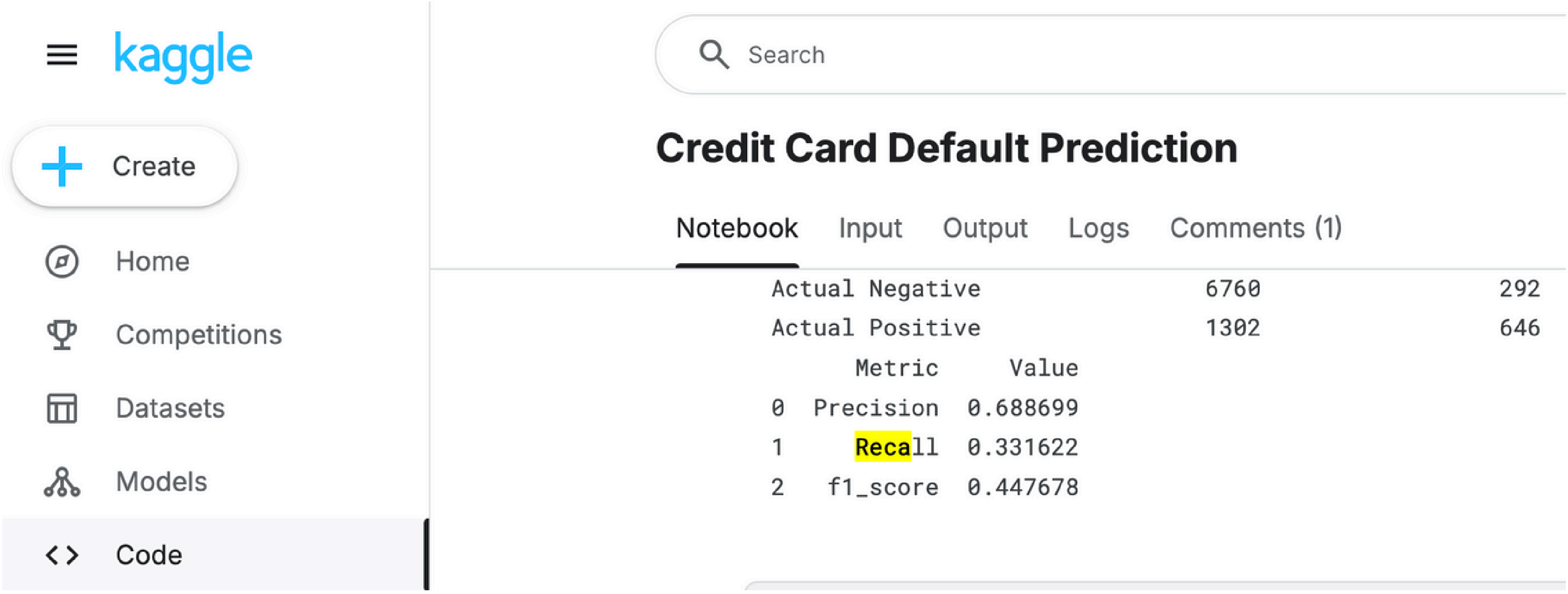


Figure 8. KNN Results from kaggle

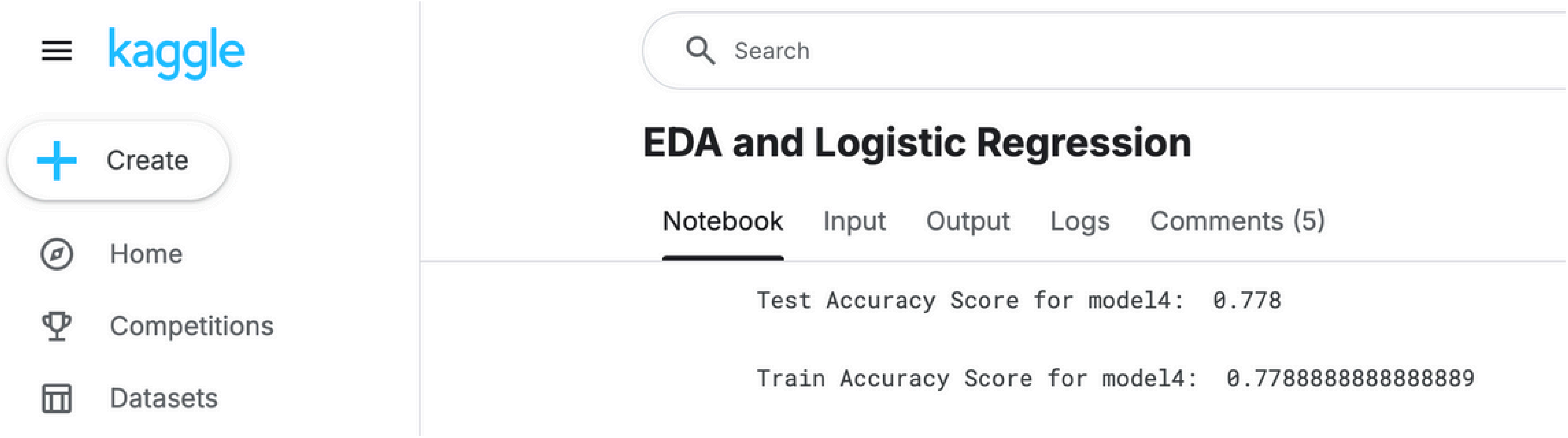


Figure 9. Random forest Results from kaggle

If we change the balance of the model, the results for accuracy will be higher, others lower

Insights and Business Implications

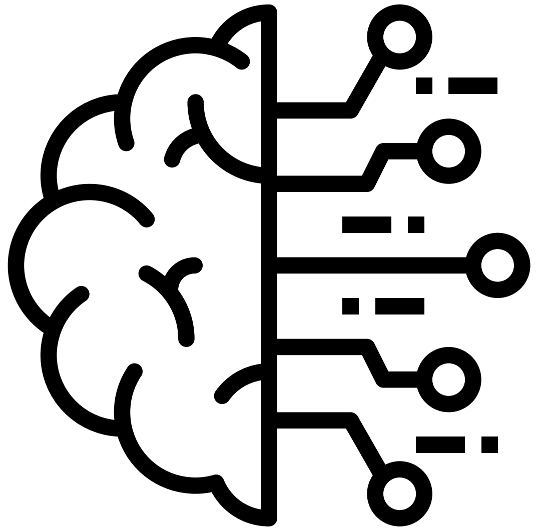
Key Insights:

- Customers with high credit limits or recent payment delays are high-risk.
- Can be integrated into automated risk management systems, improving operational efficiency and reducing defaults.

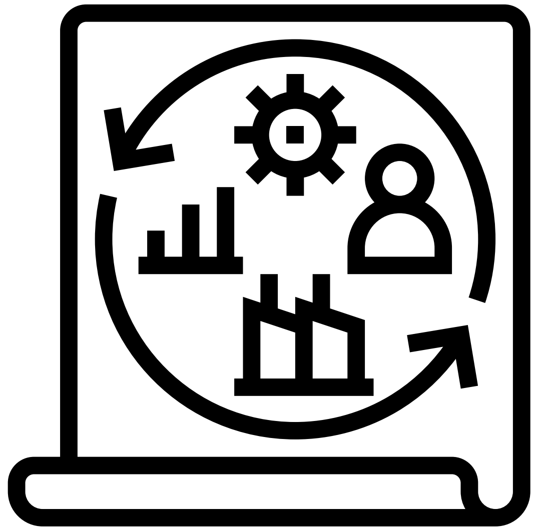
Actionable Recommendations:

- Adjust credit limits based on predicted risk and offer financial counseling.
- Prioritize follow-up on customers with high default probability

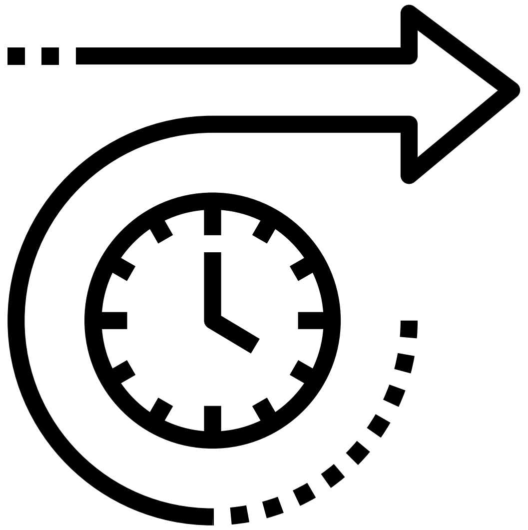
Conclusion



The analysis successfully predicted credit card default risks using machine learning models.



Best Performing Model:
- After evaluating several models (e.g., Decision tree, random forest), the **Logistic Regression** achieved the highest performance.



This analysis serves as a strong foundation for credit card default prediction. It showcases how data-driven decision-making can enhance credit risk management.

References

- https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset
- Amin Zollanvari (2024). Machine Learning with Python: Theory and Implementation.
- Yeh, I. C., & Lien, C. H. (2009). The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients. Expert Systems with Applications, 36(2), 2473–2480. DOI: 10.1016/j.eswa.2007.12.020.
- Liu, T., & Schumann, L. (2005). Data Mining and Knowledge Discovery for Credit Card Default Risk Analysis. Advances in Intelligent Data Analysis VI (Lecture Notes in Computer Science), 424–435. Springer. DOI: 10.1007/11552253_38.
- Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance, 23(4), 589-609.
- World Bank. (2022). World Development Report 2022: Finance for an Equitable Recovery.

Thank You!