Summer camp 2025

Loaners data predictions – Final presentation

Data Science Lab

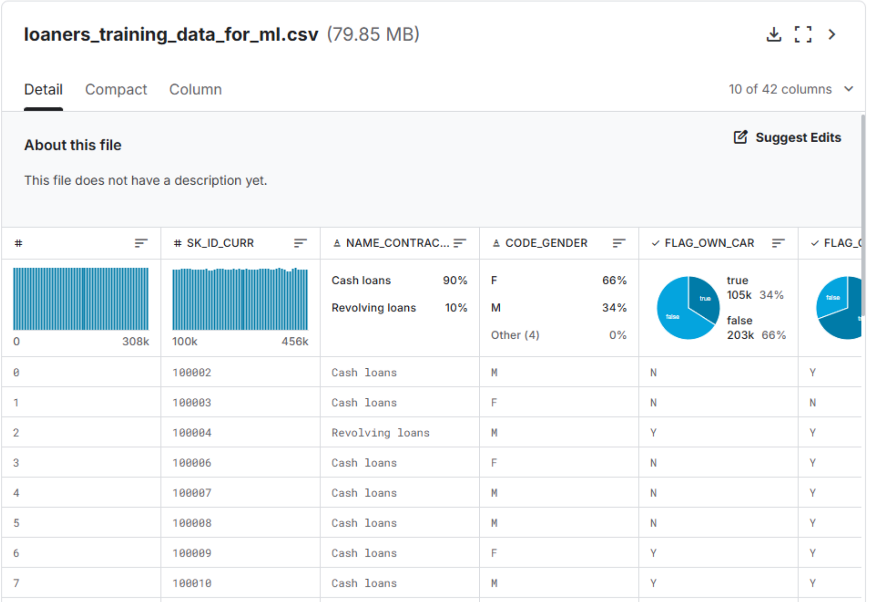
Lenny PORCHER
Lucas PUPAT
Cambyse MOUSSAVI AZARBAYEJANI

Table of contents

I. Problem definition

Objective: predict whether a person is eligible for a loan

42 columns
307510 Inputs


loaners_training_data_for_ml.csv (79.85 MB)

Detail Compact Column 10 of 42 columns

About this file Suggest Edits

This file does not have a description yet.

#	SK_ID_CURR	NAME_CONTRAC...	CODE_GENDER	FLAG_OWN_CAR	FLAG_(
0 – 308k	100k – 456k	Cash loans 90% Revolving loans 10%	F 66% M 34% Other (4) 0%	true 105k 34% false 203k 66%	
0	100002	Cash loans	M	N	Y
1	100003	Cash loans	F	N	N
2	100004	Revolving loans	M	Y	Y
3	100006	Cash loans	F	N	Y
4	100007	Cash loans	M	N	Y
5	100008	Cash loans	M	N	Y
6	100009	Cash loans	F	Y	Y
7	100010	Cash loans	M	Y	Y

kaggle

https://www.kaggle.com/datasets/benjamincornurota/bcr-loaners-data-for-solvency-prediction/data

II. Dataset description & preprocessing

a. Data type identification

Datas

II. Dataset description & preprocessing

a. Data type identification

Qualitative

Datas

Quantitative

II. Dataset description & preprocessing

a. Data type identification

II. Dataset description & preprocessing

b. Managing undefined values, median imputation and mode imputation

> Handle undefined values with Low Impact – Delete rows

- AMT_ANNUITY (N/A: 12)
- CODE_GENDER
- TARGET
- CNT_FAM_MEMBERS

```
df.dropna(subset=['TARGET', 'AMT_ANNUITY', 'CODE_GENDER', 'CNT_FAM_MEMBERS'], inplace=True)
```

> Handle missing values with median imputation

- AMT_GOODS_PRICE (N/A: 278)

```
df['AMT_GOODS_PRICE ].fillna(df['AMT_GOODS_PRICE'].median(), inplace=True)
```

> Handle missing values with mode imputation

- NAME_TYPE_SUITE

```
df[‘NAME_TYPE_SUITE’].fillna(df[‘NAME_TYPE_SUITE’].mode()[0], inplace=True)
```

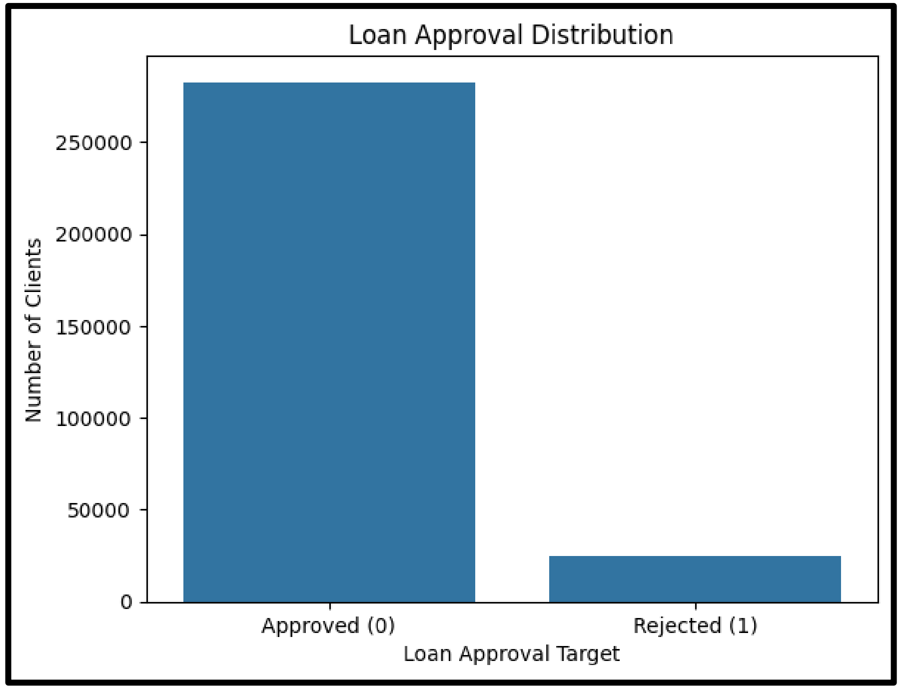
III. Dataset visualization

a. Loan Approval Distribution

Dataset balance analysis

<u>Influential data</u>

```
sns.countplot(x='TARGET', data=df)
```



Loan Approval Distribution

- Highly imbalanced – 91.93% approved

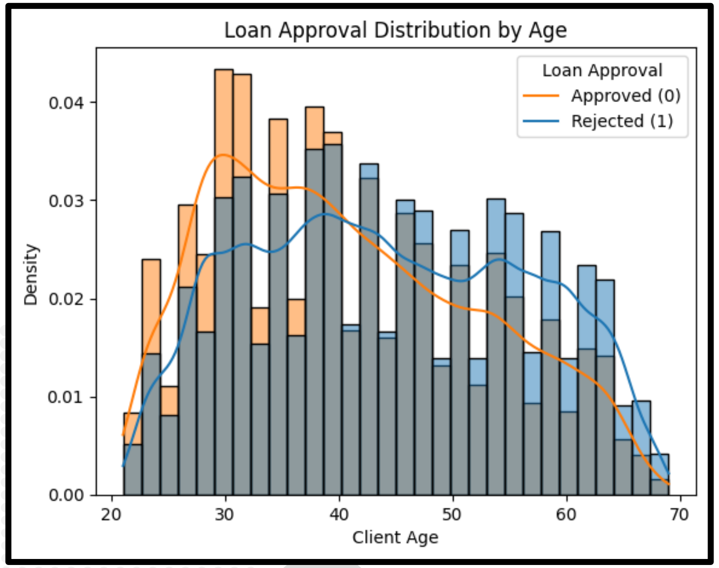
III. Dataset visualization

b. Borrower Demographics

Individual, career and relationship analysis (Examples)

Influential data

```
sns.histplot(data=df, x='LOANER_AGE', hue='TARGET', …
```

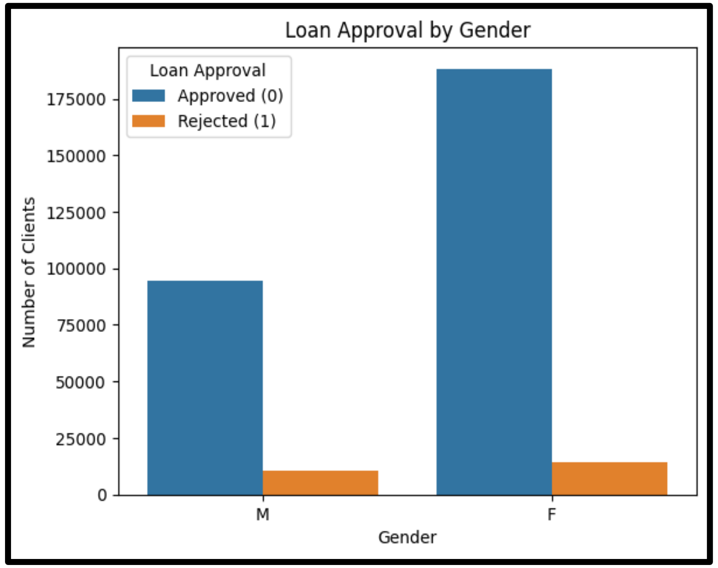


Age vs Loan Approval

- < 40 are more likely approved

Moderately influential data

```
sns.countplot(x='CODE_GENDER', hue='TARGET', data=df)
```



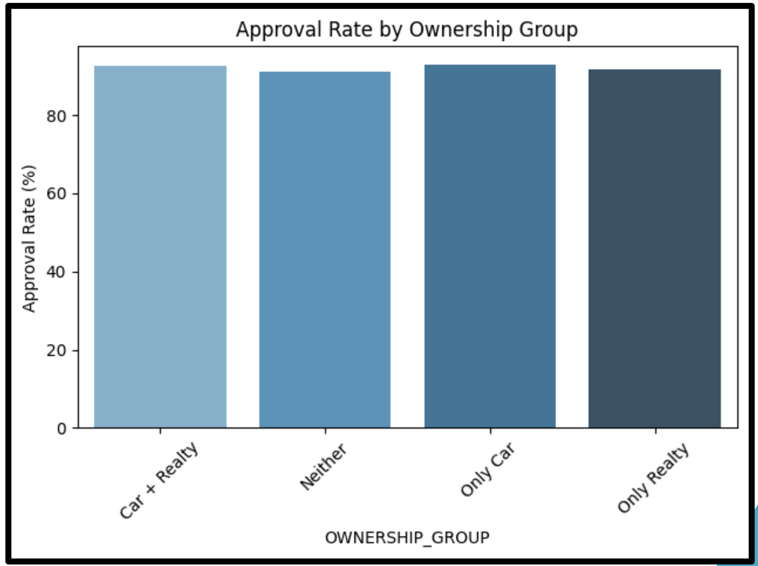
Gender vs Loan Approval

- Male approval rate: 89.86 %
- Female approval rate: 93.00%

A \ B ≈ 3%

Data with little or no influence

```
plt.bar(grouped['OWNERSHIP_GROUP'], …
```



Asset Ownership and Loan Approval

- Difference is marginal (~1%)

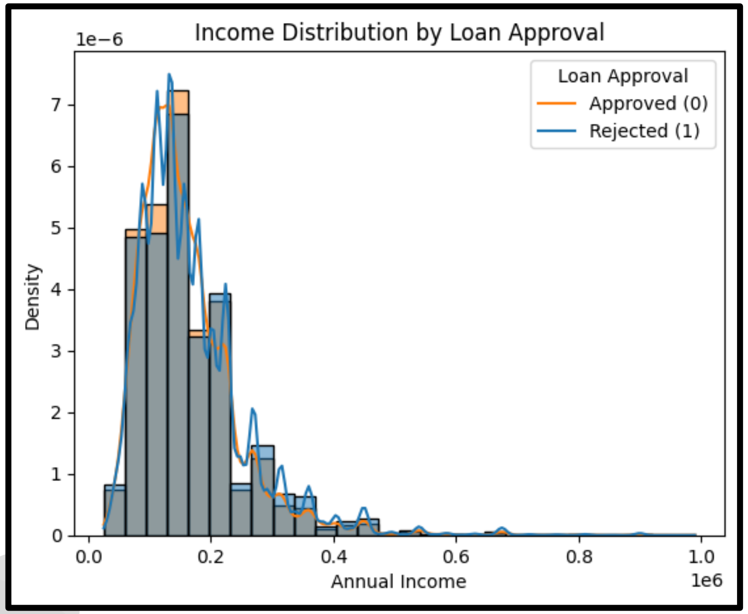
III. Dataset visualization

c. Financial situation

Analysis of individual's economic situation (Examples)

Influential data

```
sns.histplot(data=df, x='AMT_INCOME_TOTAL', hue='TARGET', …
```

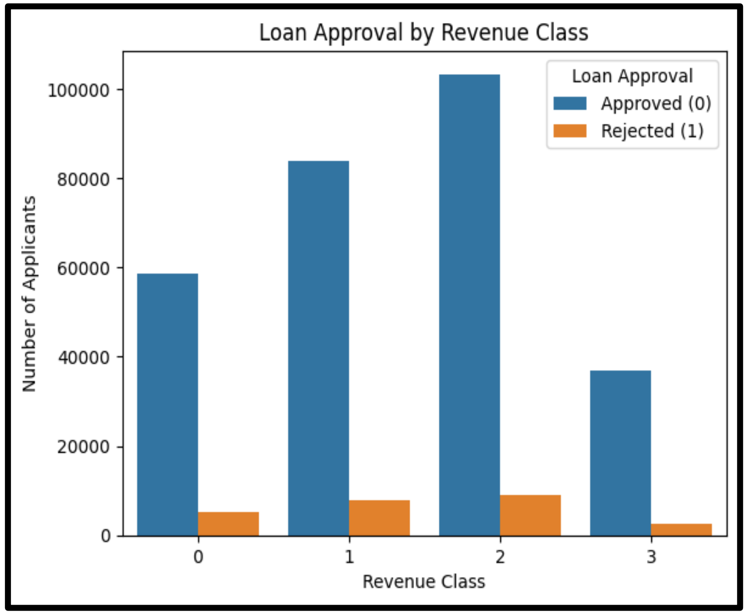


Income Distribution

- Approval highest between 50K–200K income

Data with little or no influence

```
sns.boxplot(data=df, x='REVENUE_CLASS', y='AMT_INCOME_TOTAL')
```



Revenue Class

- Difference is marginal (~1%)

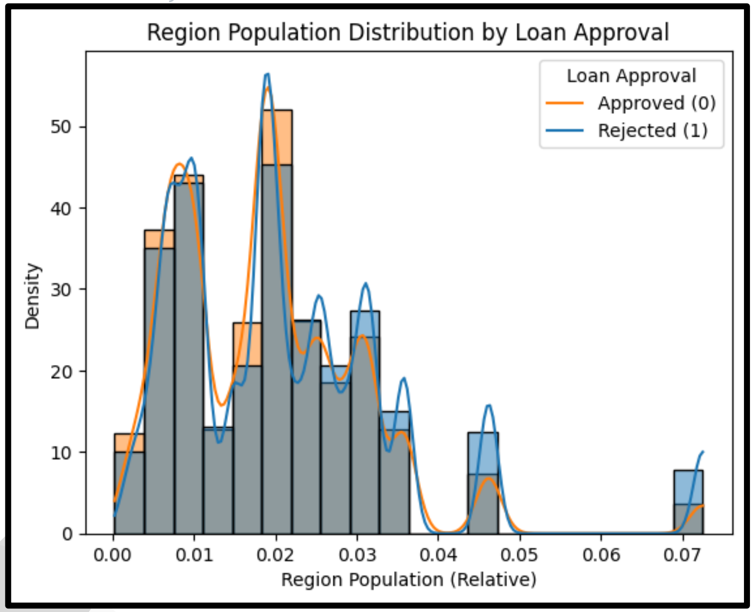
III. Dataset visualization

d. Regional data and environment

Analysis of individuals' geographical location (Examples)

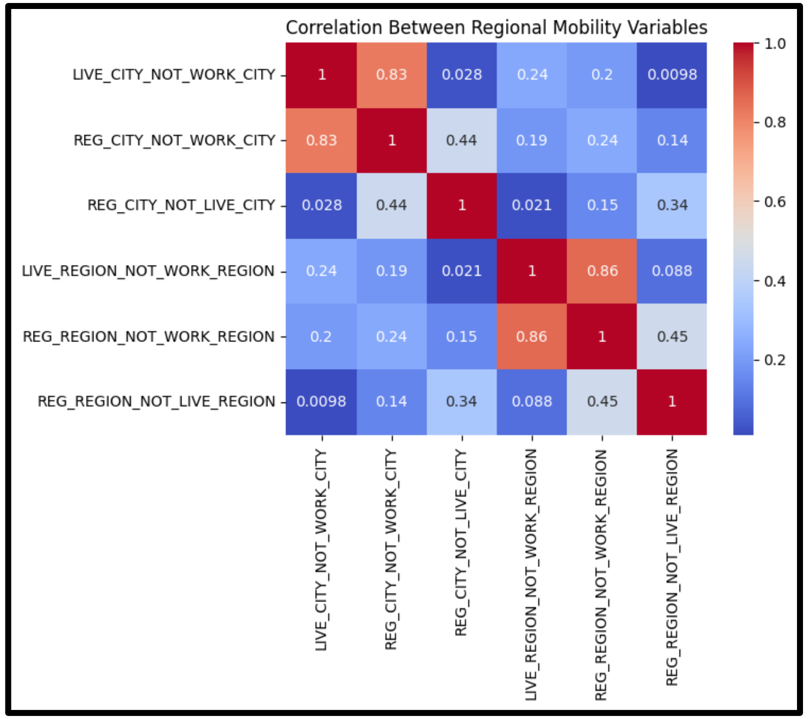
<u>Influential data</u>

```
sns.histplot(data=df, x='REGION_POPULATION_RELATIVE',
hue='TARGET', …
```



Region Population and Loan Approval

- Low region population (< 0.28) → higher approval



Regional Mobility Correlation

- High mobility → possible lower approval

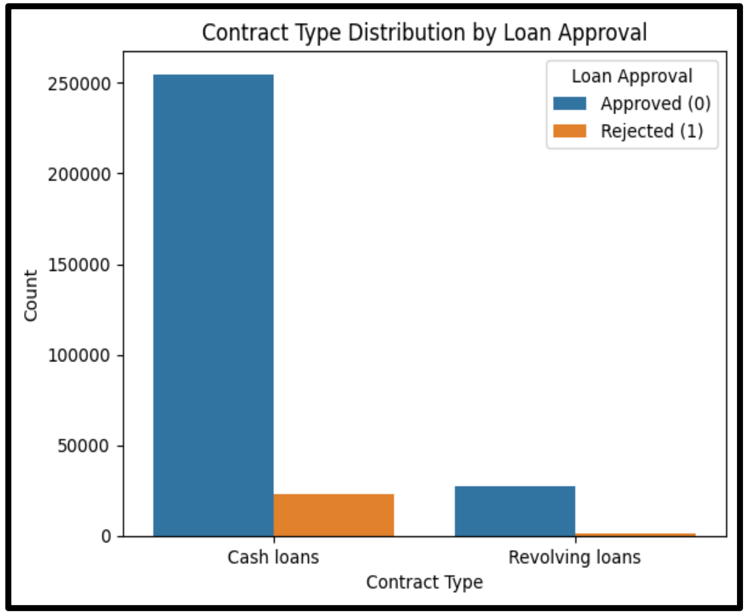
III. Dataset visualization

e. Loan application form

Individual loan analysis

Influential data

```
sns.countplot(x='NAME_CONTRACT_TYPE', hue='TARGET', data=df)
```



Contract type and Loan Approval

- Revolving loans → higher approval (94.5%)

IV. Dataset formatting

Managing qualitative & quantitative values

Qualitative values

- Convert nominal variables to 0 and 1

```
df['CODE_GENDER'] = df['CODE_GENDER'].map({'F': 0, 'M': 1})
```

- One-hot encode categorical variable

```
df = pd.get_dummies(df, columns=[
'NAME_CONTRACT_TYPE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'NAME_TYPE_SUITE',
'OWNERSHIP_GROUP'])
```

Quantitative values

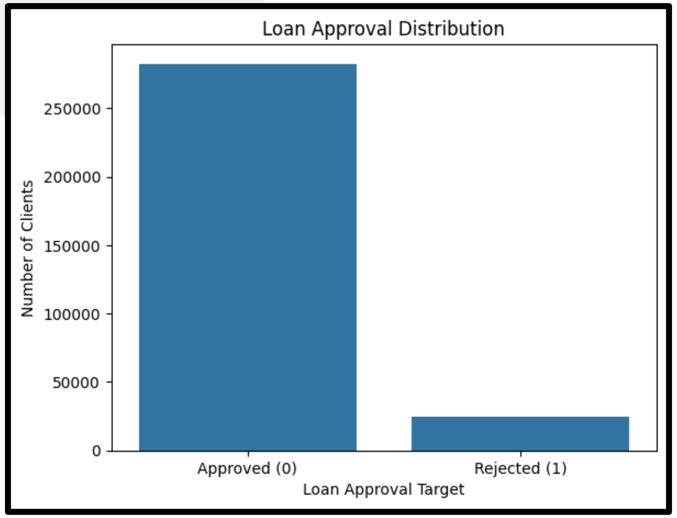
1

- Use a logarithmic transformation to reduce the impact of extreme values and skewed distributions

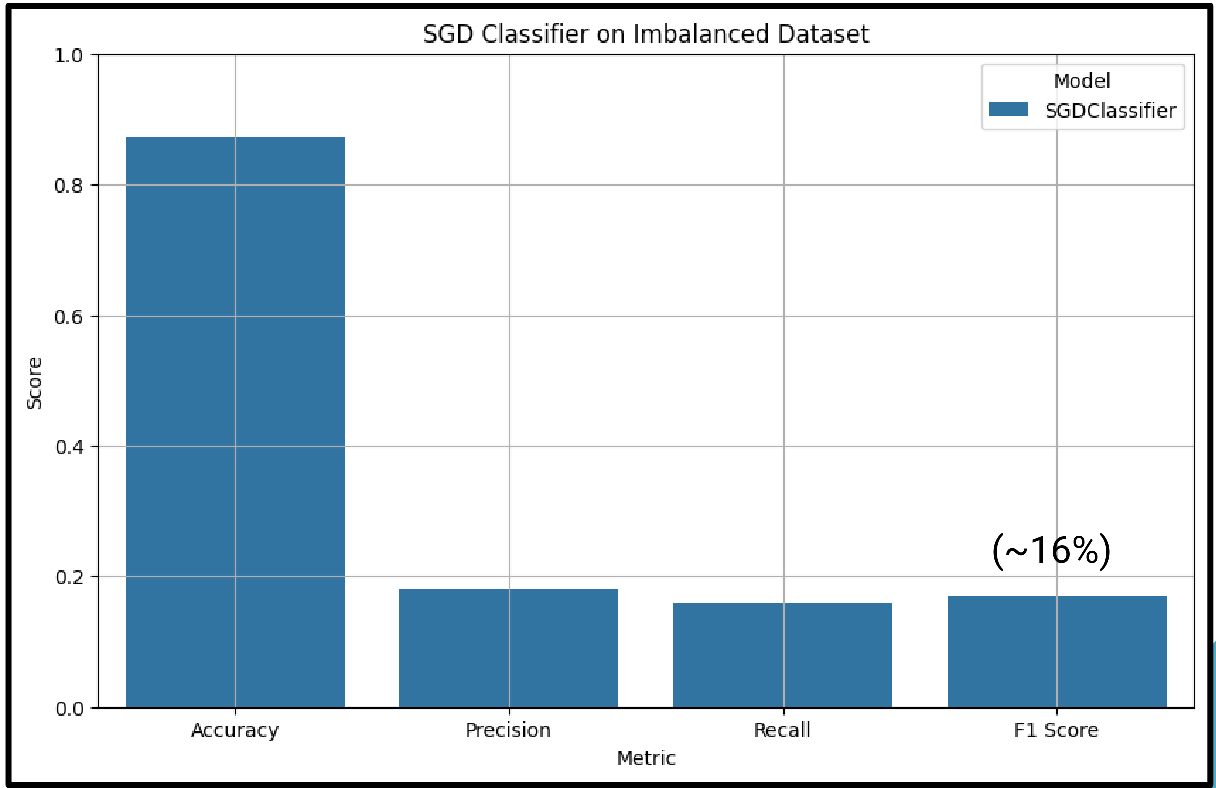
```
num_cols = [
'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_GOODS_PRICE',
'AMT_ANNUITY',
'CREDIT_SHARE', 'NUM_ANNUITY']

df[num_cols] = np.log1p(df[num_cols])
```

V. Machine Learning Model Training & Evaluation Metrics

a. Initial Model on Imbalanced Dataset



Big influence



Distribution : {0: 282430, 1: 24812}

SGDClassifier

- Accuracy 0.872463
- Precision 0.181777
- Recall 0.159475
- F1 Score 0.169897

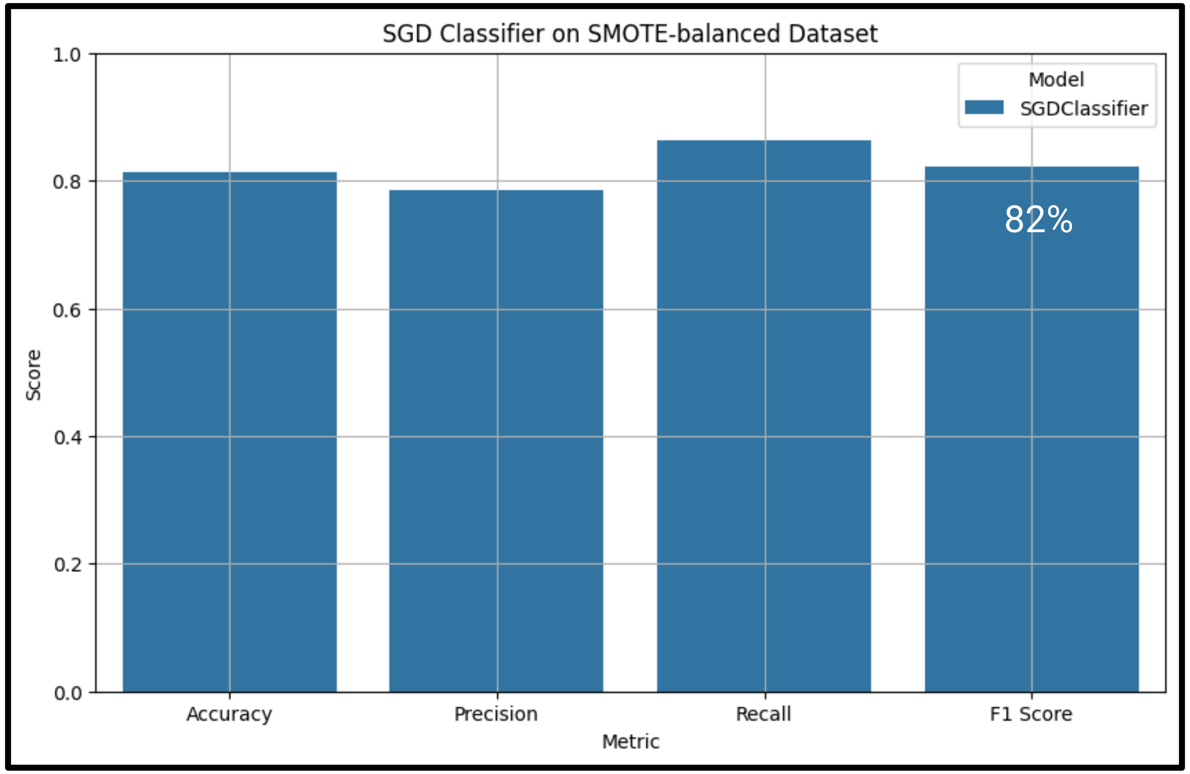
V. Machine Learning Model Training & Evaluation Metrics

b. Strategy 1: Oversampling (SMOTE)

Distribution :
{0: 282430, 1: 282430}

SGDClassifier

- Accuracy 0.812538
- Precision 0.784532
- Recall 0.862382
- F1 Score 0.821617

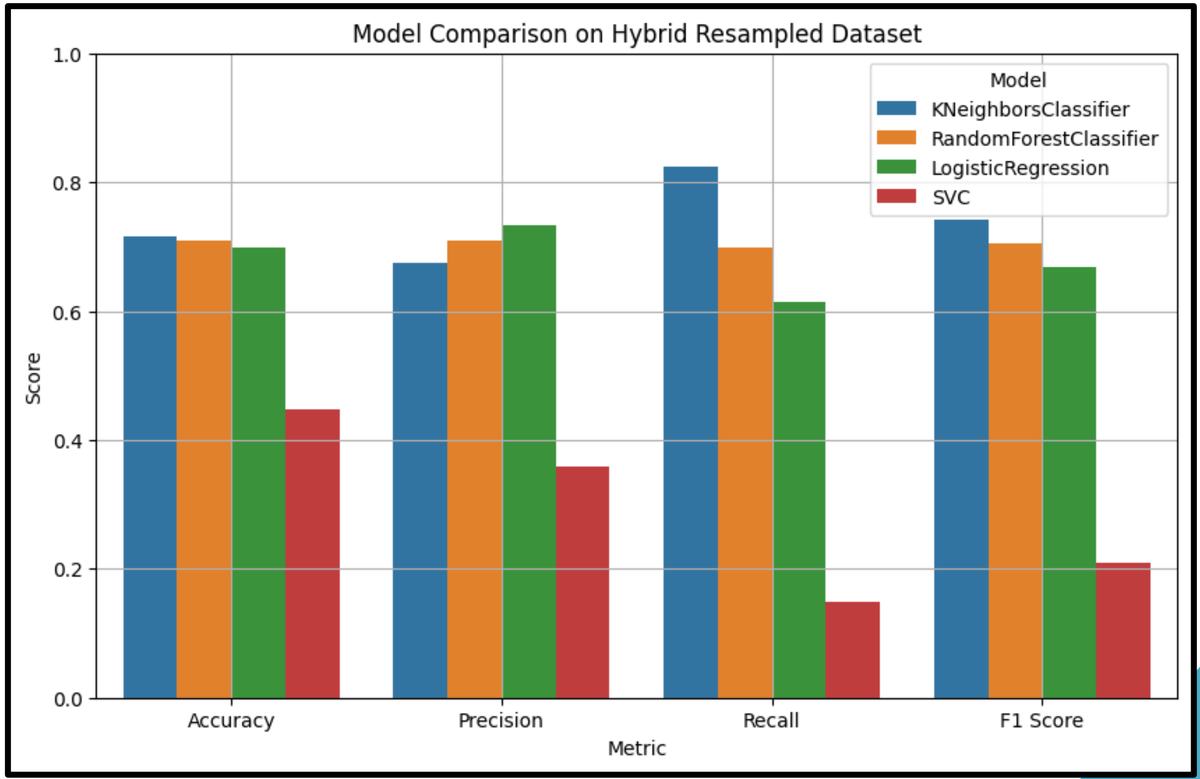


V. Machine Learning Model Training & Evaluation Metrics

c. Strategy 2: Hybrid Resampling (Under + Over Sampling)

Distribution after under sampling :
{0.0: 49624, 1.0: 24812}

Distribution after over sampling :
{0.0: 49624, 1.0: 49624}



V. Machine Learning Model Training & Evaluation Metrics

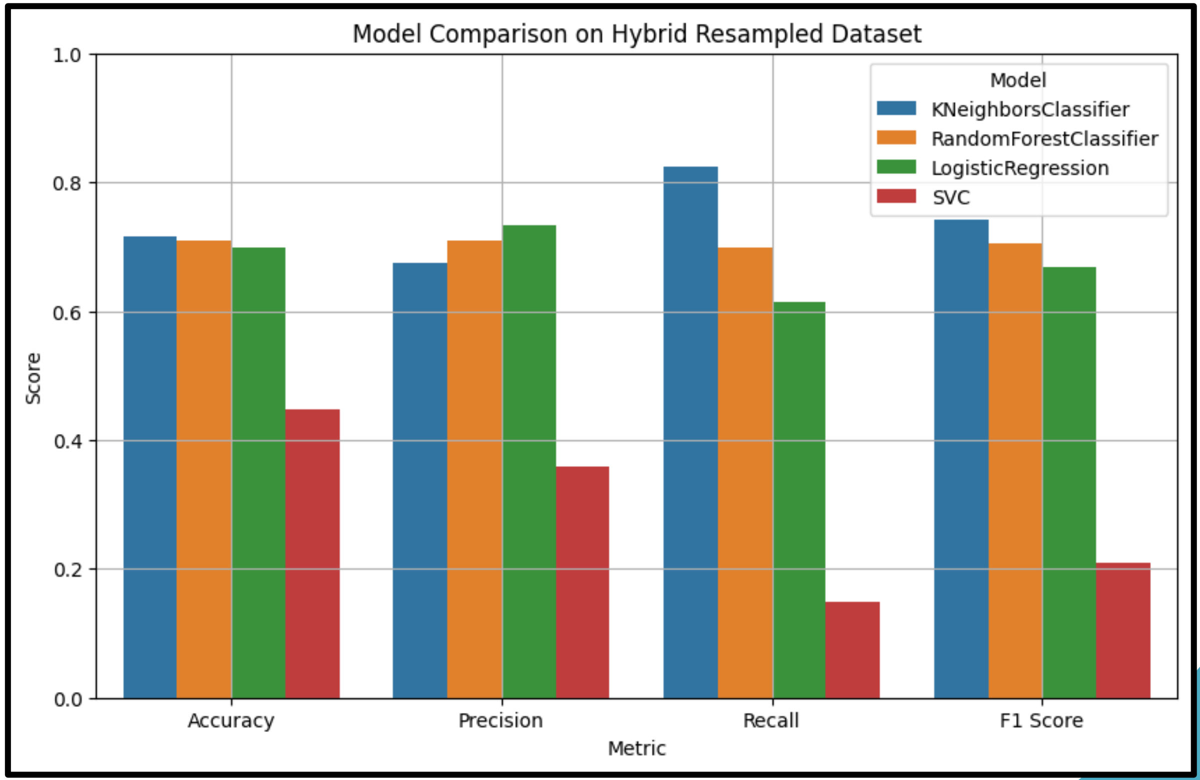
c. Strategy 2: Hybrid Resampling (Under + Over Sampling)

KNeighbors Classifier	Accuracy: 0.716423
	Precision: 0.674990
	Recall: 0.823116
	F1 Score: 0.741730

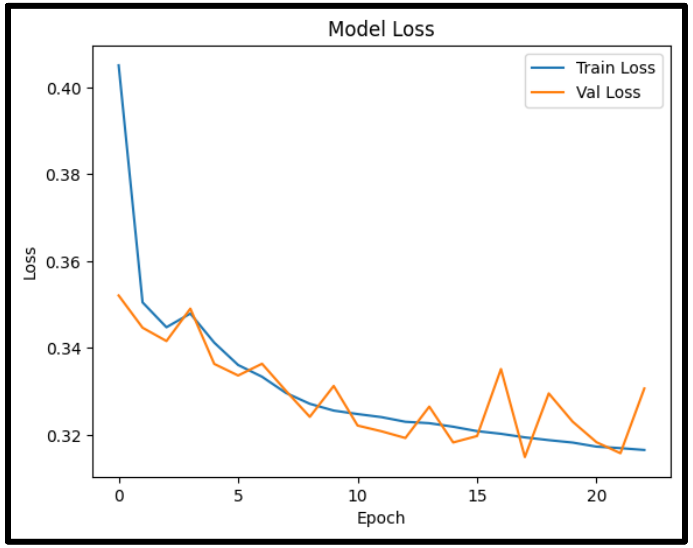
Random Forest Classifier	Accuracy: 0.709572
	Precision: 0.709344
	Recall: 0.699593
	F1 Score: 0.704435

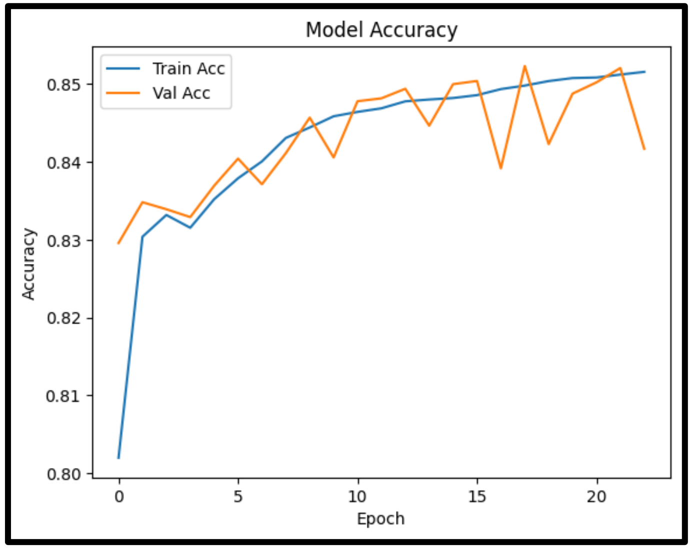
Logistic Regression	Accuracy: 0.698287
	Precision: 0.732210
	Recall: 0.615071
	F1 Score: 0.668548

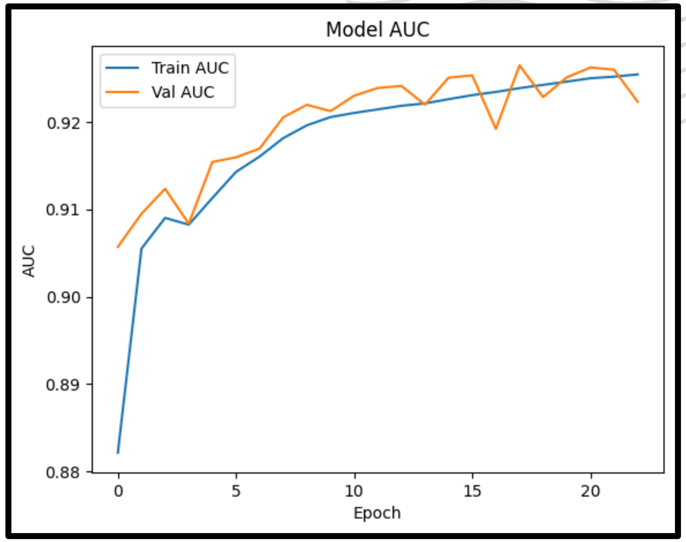
SVC	Accuracy: 0.447254
	Precision: 0.358546
	Recall: 0.148676
	F1 Score: 0.210193



VI. Deep Learning Model Training & Evaluation Metrics



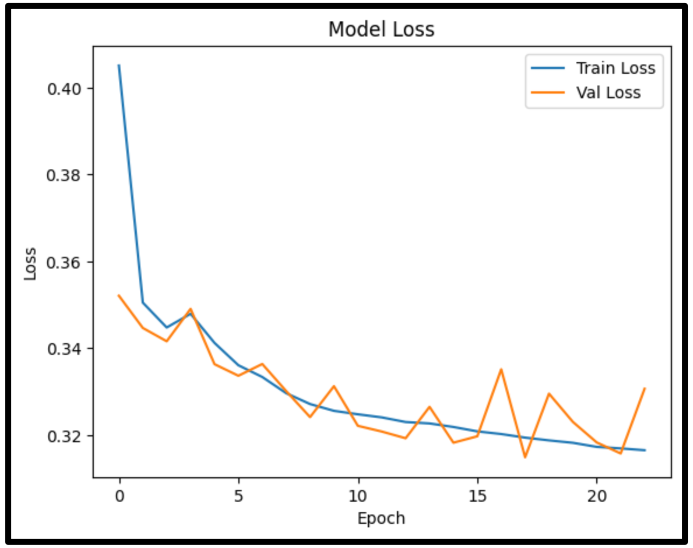


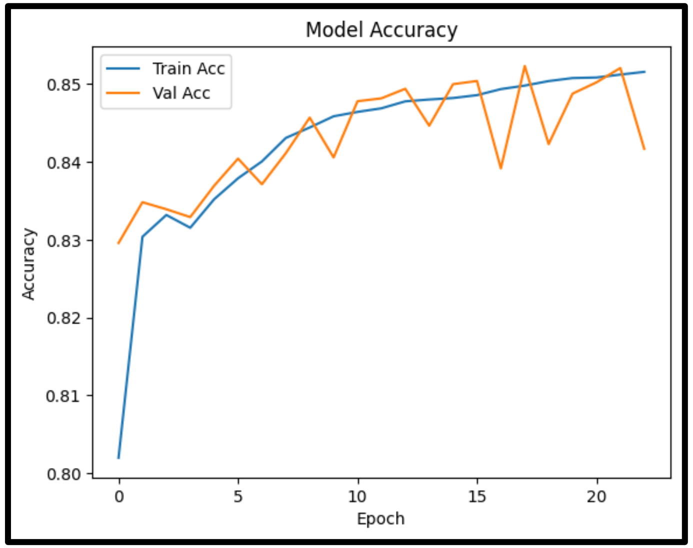


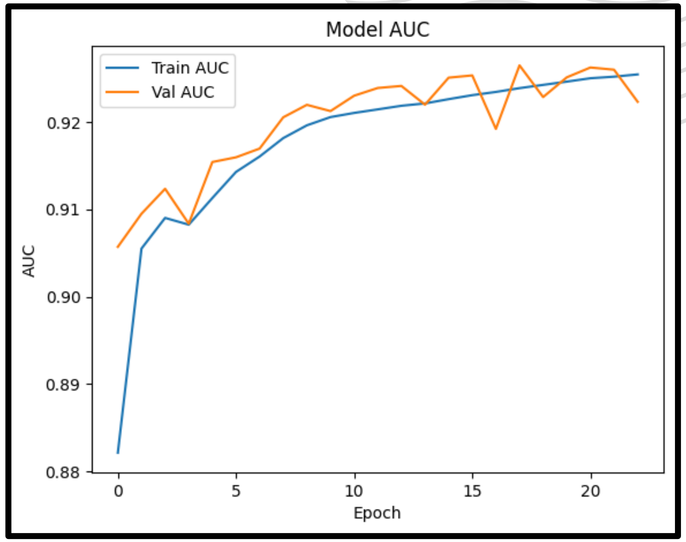
Deep Learning

- Accuracy 0.8523
- Precision 0.9201
- Recall 0.7719
- F1 Score 0.8395

VI. Deep Learning Model Training & Evaluation Metrics



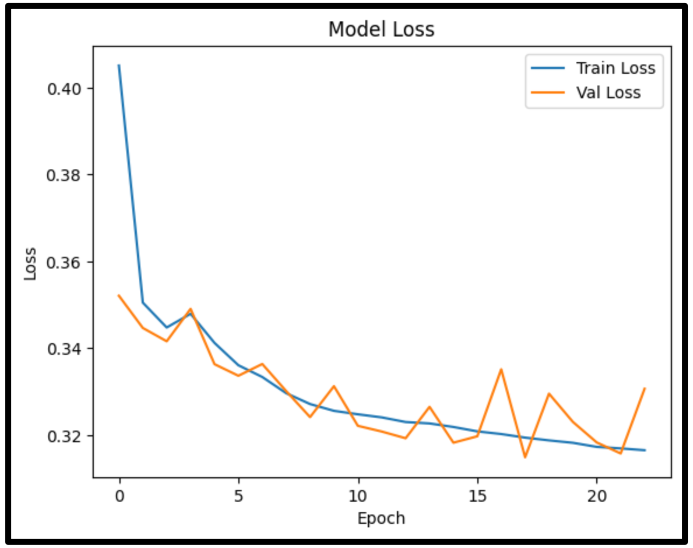


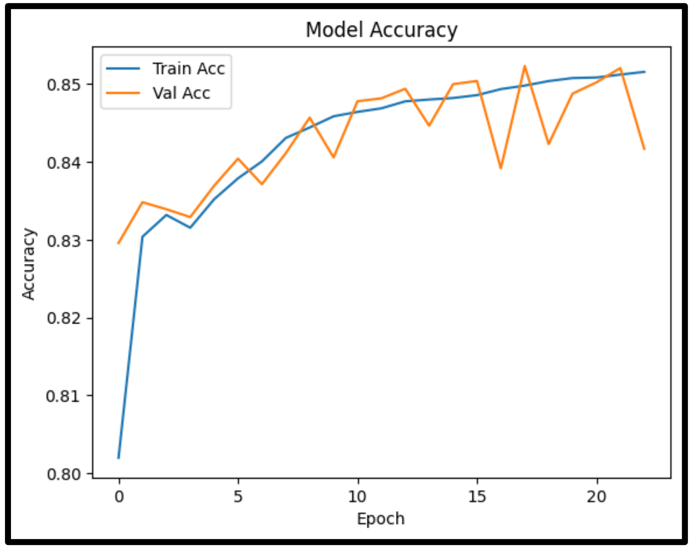


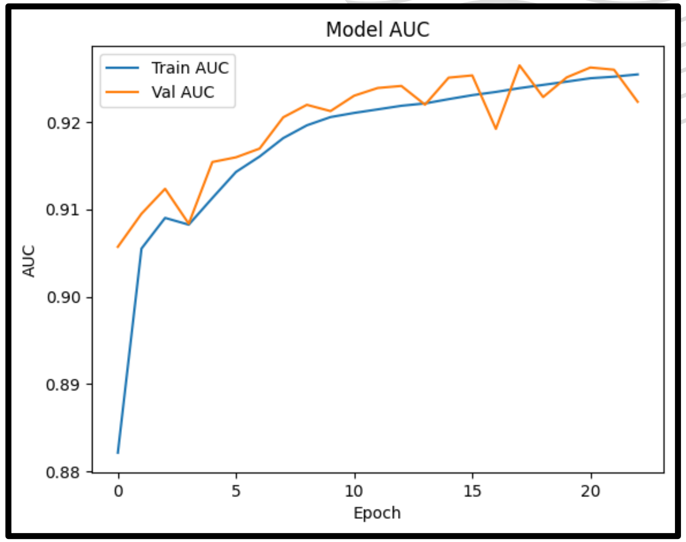
Deep Learning

- Accuracy 0.8523
- Precision **0.9201** +
- Recall **0.7719** -
- F1 Score 0.8395

VI. Deep Learning Model Training & Evaluation Metrics







Deep Learning

- Accuracy 0.8523
- Precision **0.9201** +
- Recall **0.7719** -
- F1 Score 0.8395

SGDClassifier

- Accuracy 0.812538
- Precision **0.784532**
- Recall **0.862382**
- F1 Score 0.821617

VII. Responsible AI Practices

Bias in data

Watch for historical and representation biases.

Data limitations

Model can't reflect individual or internal bank factors.

Fairness evaluation

Use multiple metrics, not just F1-score (demographic parity, equal opportunity, …

Continuous monitoring

Model can't reflect individual or internal bank factors.

Data privacy

Update regularly to stay fair and accurate.

Conclusion

SGDClassifier:
Achieving an F1-score of 80%.

Areas for Improvement:

- Include minimum class representation data
- Build a weighted scoring system for bank evaluation

Summer camp 2025

Loaners data predictions – Final presentation

Thank you for listening

Lenny PORCHER
Lucas PUPAT
Cambyse MOUSSAVI AZARBAYEJANI