



Early warning model for credit default

Executive presentation





Funding Credit team

Analysts:

Kevin M. Figueroa

September 19th, 2022

24 lines (17 sloc) | 1.12 KB

<>  Raw Blame   

Stride

Early warning model for credit default

Repository

Funding Credit team

Analysts:

Kevin M. Figueroa

September 19th 2022

Documents:

- [Executive report notebook](#)
- [Methodologic report notebook](#)
- [Library of functions created](#)
- [Final predictions](#)
- [Case dashboard](#)

About me

- BA in Economics and finance
- MA in Economics and data science
- 5 years in economic and financial consulting
- Fintech consultant: Risk Analysis
- Today's role: Stride Credit Risk Analyst



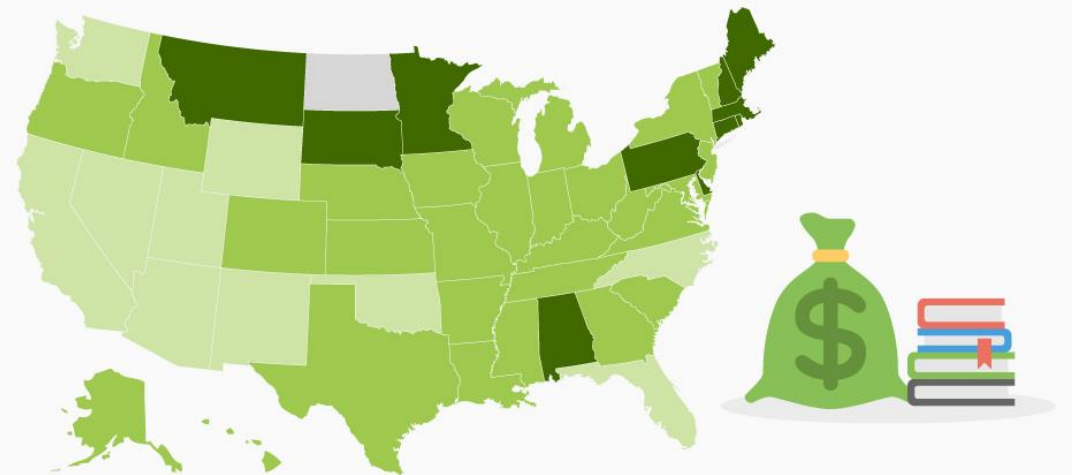
Fintechs' challenge in the growing student loan market

- Fintech: Reduce the cost of credit and increase financial inclusion
- Market size: 44 million borrowers who collectively owed USD 1.5 trillion in student loan debt
- Average student in the Class of 2016: \$37,000
- Student loan challenge!
 - Longer terms than BL
 - 1-4 yrs to first payment
 - Drastical conditions change for a graduate

U.S. Student Debt Is a National Problem

Average student loan debt by state in U.S. dollars

● >\$31,000 ● \$26,000 - \$30,000 ● <\$26,000 ● Insufficient data



Data published in September 2018. Data only include the 2017 undergraduate class of non-transfer students pursuing a bachelor's degree, loans made to students while enrolled as an undergrad, and co-signed loans.

CC BY ND
@StatistaCharts

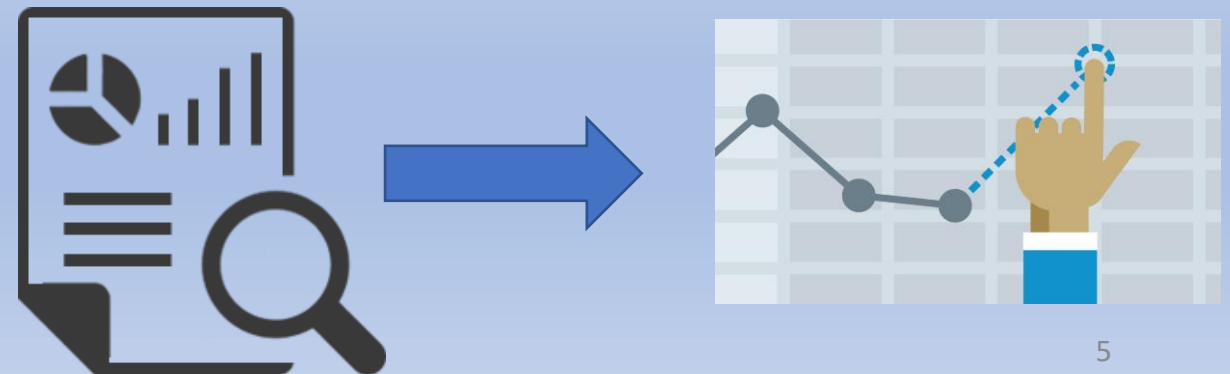
Source: The Institute for College Access & Success via CNBC

statista



Purpose

- Leverage all data available
- Database found but documentation is missing
- Our goal: obtain as much insight as possible that could be useful to improve Stride's operations
- Specific objectives:
 - Perform a deep exploratory data analysis process to understand as much of the data as possible.
 - Clean the data.
 - Produce a model that can help identify loans at risk of default.
 - Predict defaults in current portfolio loans.
 - Propose a business strategy that can take advantage of the predictive power of the model.



Exploratory data analysis s results

- Description of the data
- Score mistery
- Number of dependants issue
- Age issue
- Scales to standarize and Weak correlations

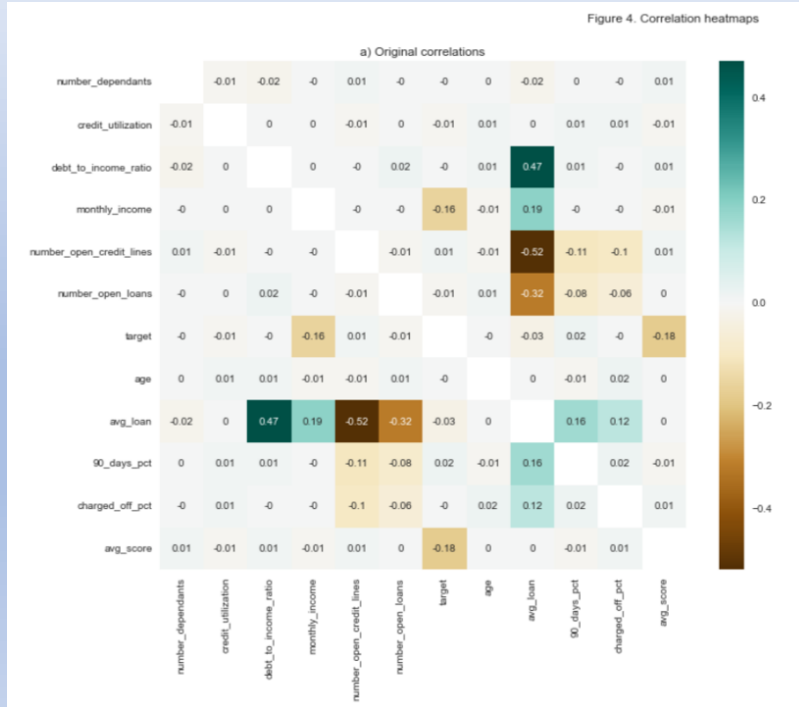
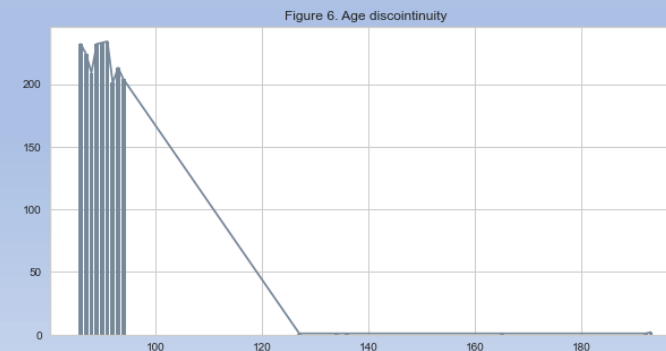


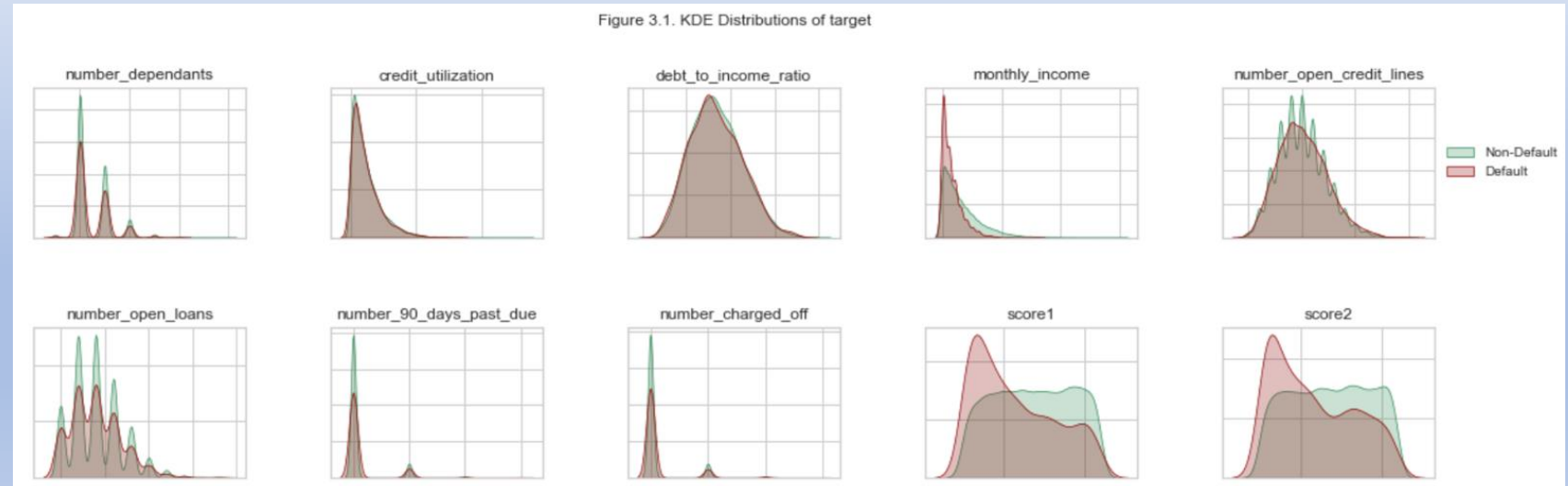
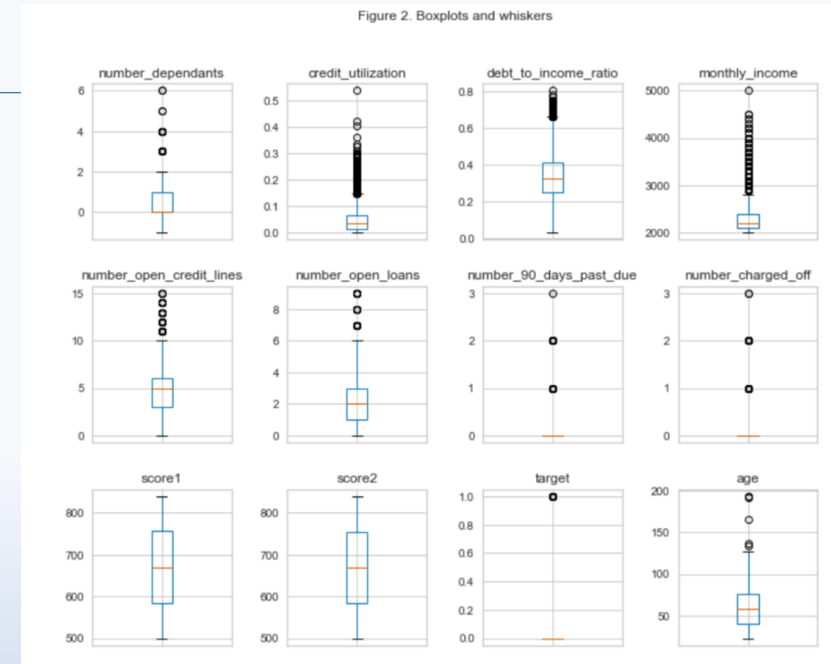
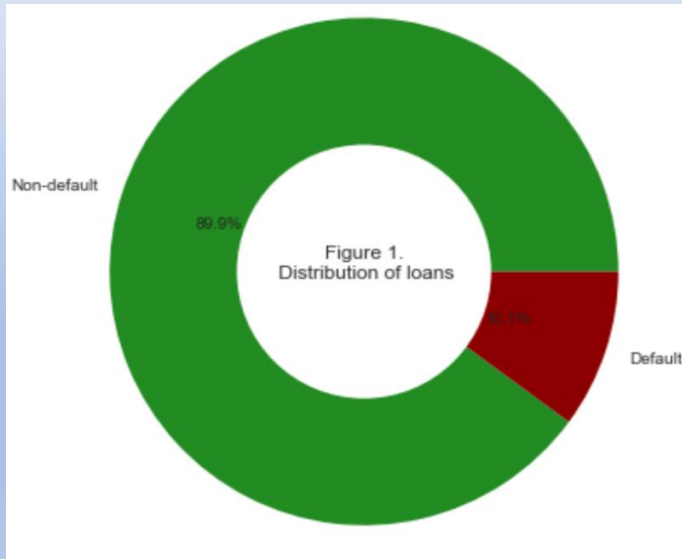
Table 1. Summary of variables training set

| | count | mean | std | min | 25% | 50% | 75% | max | Missing vals | Variable type |
|--------------------------|---------|---------|--------|---------|---------|---------|---------|---------|--------------|---------------|
| number_dependants | 16671.0 | 0.49 | 0.72 | -1.00 | 0.00 | 0.00 | 1.00 | 6.00 | False | int64 |
| credit_utilization | 16671.0 | 0.05 | 0.05 | 0.00 | 0.01 | 0.03 | 0.07 | 0.54 | False | float64 |
| debt_to_income_ratio | 16671.0 | 0.33 | 0.12 | 0.03 | 0.25 | 0.33 | 0.41 | 0.80 | False | float64 |
| monthly_income | 16671.0 | 2289.90 | 319.47 | 2000.00 | 2100.00 | 2200.00 | 2400.00 | 5000.00 | False | int64 |
| number_open_credit_lines | 16671.0 | 5.00 | 2.25 | 0.00 | 3.00 | 5.00 | 6.00 | 15.00 | False | int64 |
| number_open_loans | 16671.0 | 2.03 | 1.43 | 0.00 | 1.00 | 2.00 | 3.00 | 9.00 | False | int64 |
| number_90_days_past_due | 16671.0 | 0.10 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 3.00 | False | int64 |
| number_charged_off | 16671.0 | 0.10 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 3.00 | False | int64 |
| score1 | 16671.0 | 669.90 | 98.01 | 500.00 | 585.00 | 669.00 | 756.00 | 839.00 | False | int64 |
| score2 | 16671.0 | 669.49 | 98.61 | 500.00 | 583.00 | 670.00 | 754.00 | 839.00 | False | int64 |
| target | 16671.0 | 0.10 | 0.30 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | False | int64 |
| age | 16671.0 | 58.44 | 20.58 | 23.00 | 41.00 | 58.00 | 76.00 | 193.00 | False | int32 |
| total_debt | 16671.0 | 765.01 | 291.69 | 62.30 | 556.40 | 734.38 | 941.70 | 2663.30 | False | float64 |
| number_accounts | 16671.0 | 7.03 | 2.66 | 0.00 | 5.00 | 7.00 | 9.00 | 19.00 | False | int64 |
| avg_loan | 16671.0 | 131.26 | 97.10 | 0.00 | 74.74 | 107.93 | 157.12 | 1895.99 | False | float64 |
| 90_days_pct | 16671.0 | 0.02 | 0.07 | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | False | float64 |
| charged_off_pct | 16671.0 | 0.02 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | False | float64 |
| avg_score | 16671.0 | 669.70 | 69.95 | 502.00 | 619.00 | 670.00 | 720.00 | 839.00 | False | float64 |
| score_change | 16671.0 | -0.40 | 138.15 | -337.00 | -99.00 | 0.00 | 97.00 | 338.00 | False | int64 |



Exploratory data analysis s results

- Description of the data
- Unbalanced Target distribution
- Features Distributions
- No clear segmentation → Weak correlations



Evaluating if both samples were drawn from the same population

- Kolmogorov-Smirnov tests for 2 samples
 - H_0 : Both samples share the same distribution
 - Reject if $p\text{-val} \geq 0.05$
- Testing and objective data come from the same population
- All variables for both sets were drawn from the same distribution → Same populations

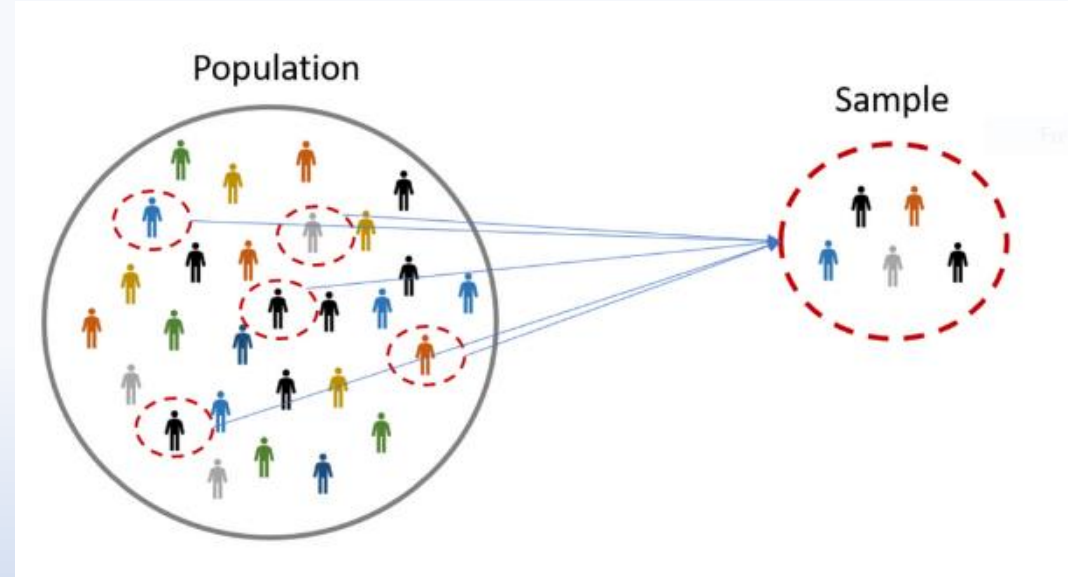
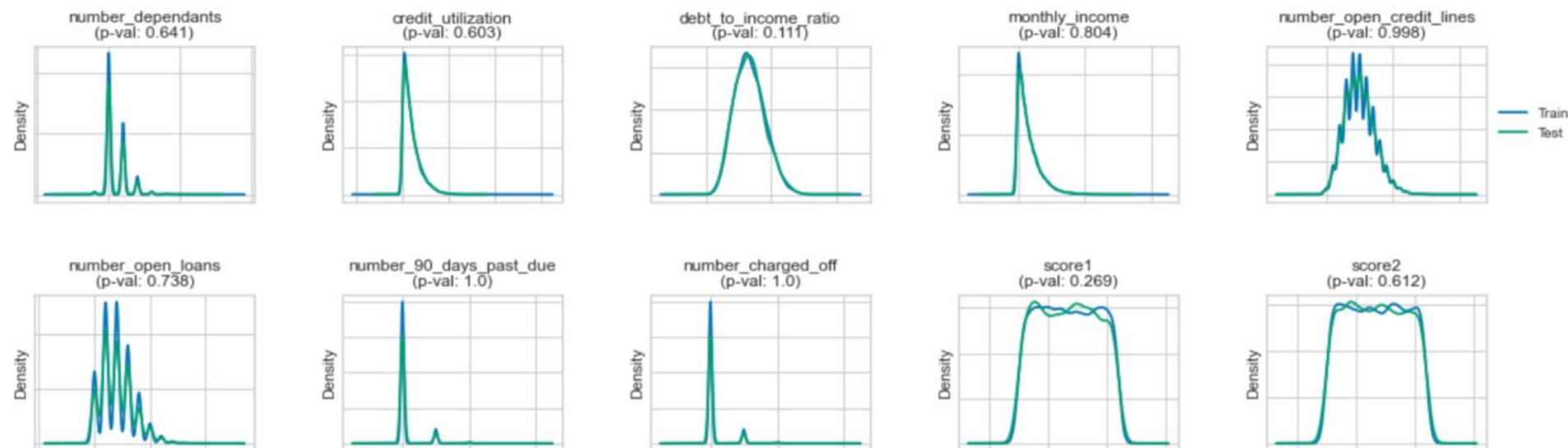
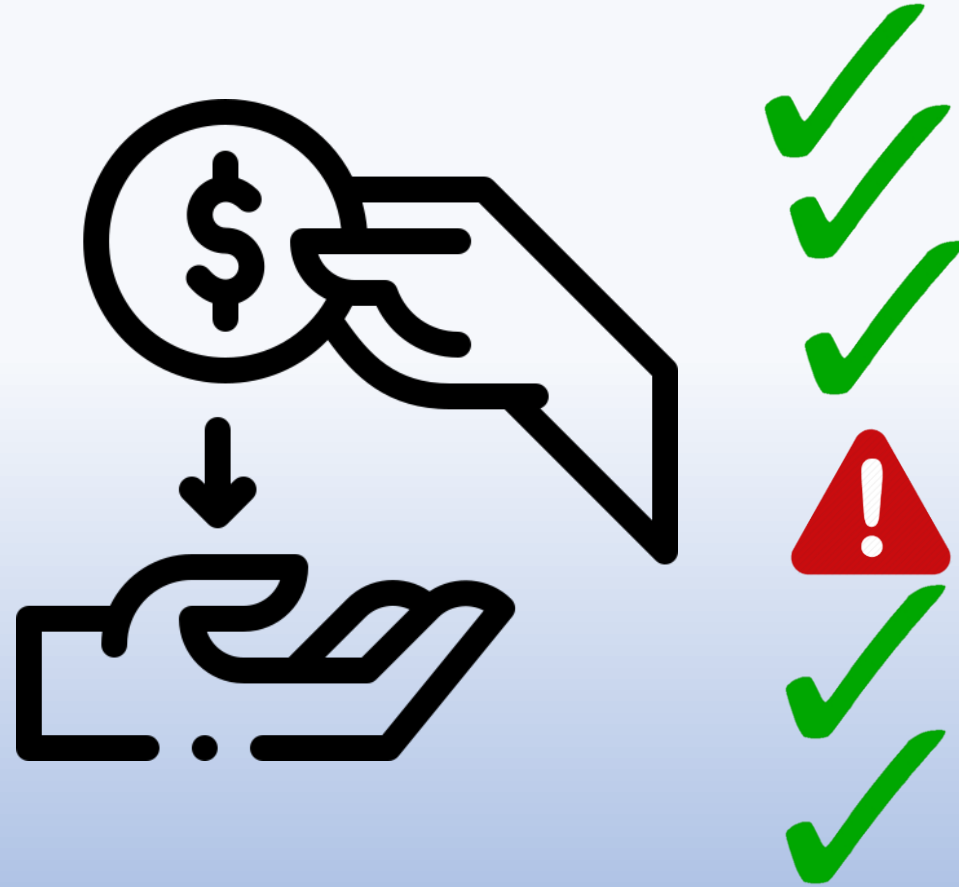


Figure 5. Train vs Test Distribution tests



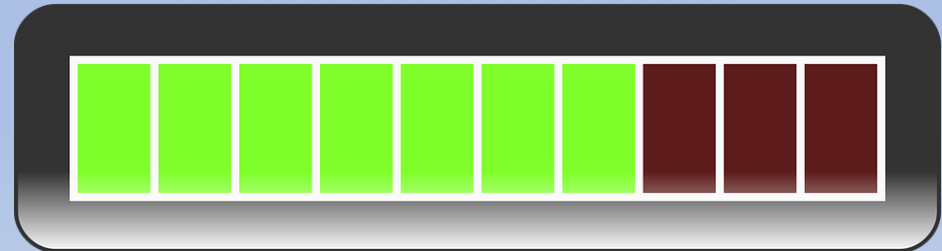
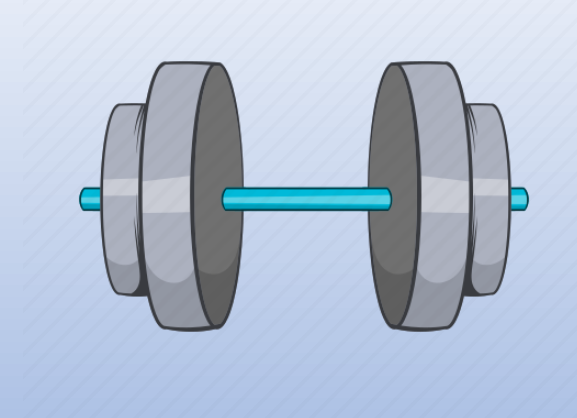
What can be done with this dataset? Early Warnings for Collections (EWC) tool

- Timing mystery: only 25% of ages below 41 years
- Not useful data the evaluate loan applications
- Useful to track loans and predict if an already existing loan is likely to default any time soon



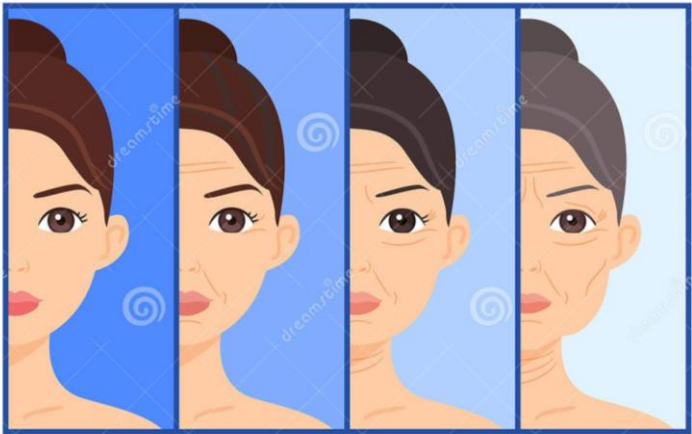
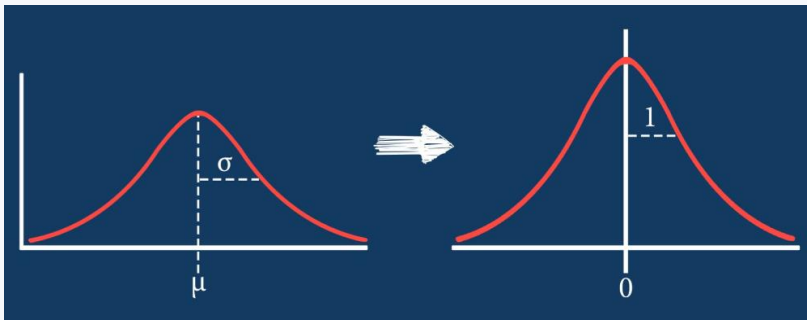
Additional variables to consider

- Number of loans with stride (internal data)
- Loan terms (internal data)
- Initial date (internal data)
- Information about the program to attend (require acceptance letter during the application.
- GPA (require transcripts of previous academic levels and follow-up after graduation
- Gym membership (ask during application)
- Loan percentage repaid (internal data):



Data cleaning

- Filling missing values (-1)
- Fixing ages
- Standarization



←20—35—55—75→

Table 3: Summary after corrections

| | count | mean | std | min | 25% | 50% | 75% | max | Missing vals | Variable type |
|--------------------------|---------|---------|--------|---------|---------|---------|---------|---------|--------------|---------------|
| number_dependants | 16671.0 | 0.50 | 0.71 | 0.00 | 0.00 | 0.00 | 1.00 | 6.00 | False | int64 |
| credit_utilization | 16671.0 | 0.05 | 0.05 | 0.00 | 0.01 | 0.03 | 0.07 | 0.54 | False | float64 |
| debt_to_income_ratio | 16671.0 | 0.33 | 0.12 | 0.03 | 0.25 | 0.33 | 0.41 | 0.80 | False | float64 |
| monthly_income | 16671.0 | 2289.90 | 319.47 | 2000.00 | 2100.00 | 2200.00 | 2400.00 | 5000.00 | False | int64 |
| number_open_credit_lines | 16671.0 | 5.00 | 2.25 | 0.00 | 3.00 | 5.00 | 6.00 | 15.00 | False | int64 |
| number_open_loans | 16671.0 | 2.03 | 1.43 | 0.00 | 1.00 | 2.00 | 3.00 | 9.00 | False | int64 |
| target | 16671.0 | 0.10 | 0.30 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | False | int64 |
| age | 16671.0 | 58.40 | 20.47 | 23.00 | 41.00 | 58.00 | 76.00 | 94.00 | False | int64 |
| avg_loan | 16671.0 | 131.26 | 97.10 | 0.00 | 74.74 | 107.93 | 157.12 | 1895.99 | False | float64 |
| 90_days_pct | 16671.0 | 0.02 | 0.07 | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | False | float64 |
| charged_off_pct | 16671.0 | 0.02 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 | 2.00 | False | float64 |
| avg_score | 16671.0 | 669.70 | 69.95 | 502.00 | 619.00 | 670.00 | 720.00 | 839.00 | False | float64 |

Model creation : contextualizing concepts

- Positives observation: Defaulted loans (Contain the target we want to predict)
- Negatives observations: Non-defaulted loans (Doesn't contain the target we want to predict)

Evaluation metrics

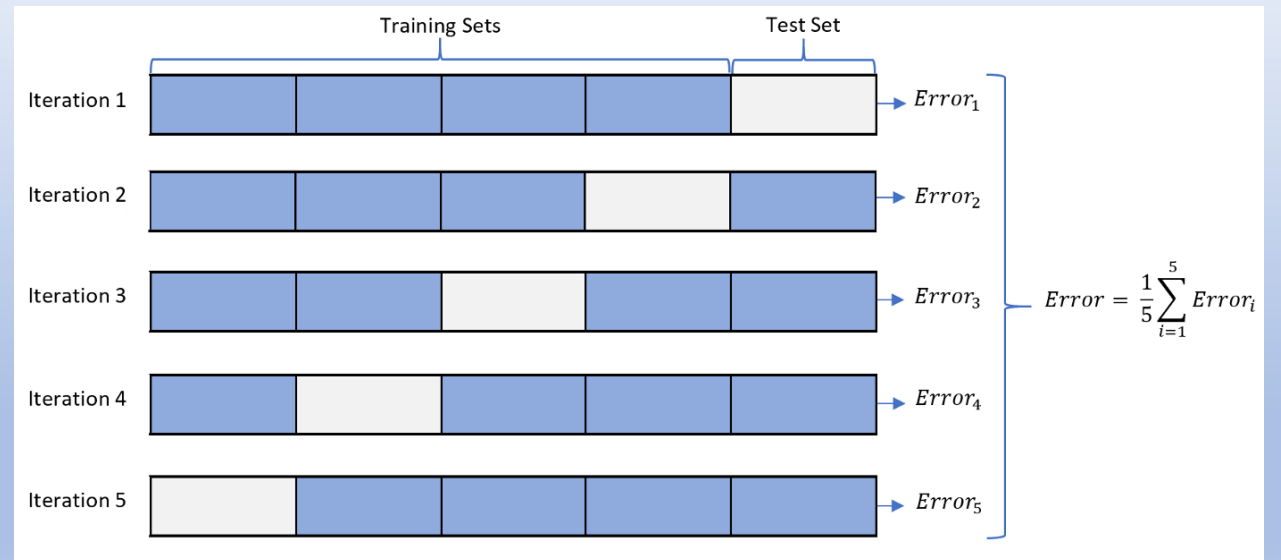
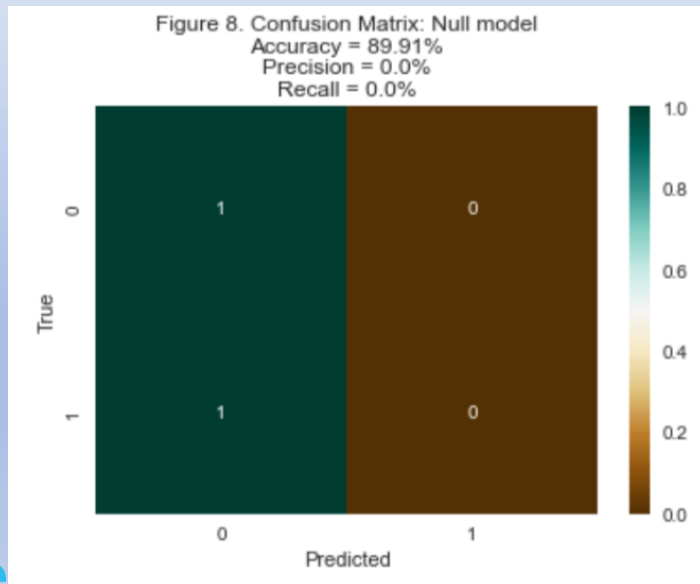
- Accuracy: How is the model predicting both defaulters and non-defaulters?
- Precision: What percentage of the default cases predicted were actual defaulters?
- Recall: What percentage of the actual defaulters was the model able to predict?
- **Best evaluation metric → Recall**

Table 4. Possible outcomes

| | Predicted Non-default | Predicted Default |
|-------------|-----------------------|-------------------|
| Non-default | True negative | False positive |
| Default | False negative | True positive |

Model creation: Splitting data and null model

- 80% of the observations (13,336) will be used for training purposes.
- 20% (3,335) of the observations for testing purposes.
- Stratifying: keeping default proportions
- Null model no defaulters



Model creation: PyCaret



- PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows.

- Best model → Decision Tree

•

```
DecisionTreeClassifier(
```

```
    ccp_alpha=0.0,|
    class_weight=None,
    criterion='gini',
    max_depth=None,
    max_features=None,
    max_leaf_nodes=None,
    min_impurity_decrease=0.0,
    min_impurity_split=None,
    min_samples_leaf=1,
    min_samples_split=2,
    min_weight_fraction_leaf=0.0,
    presort='deprecated',
    random_state=0,
    splitter='best'
```

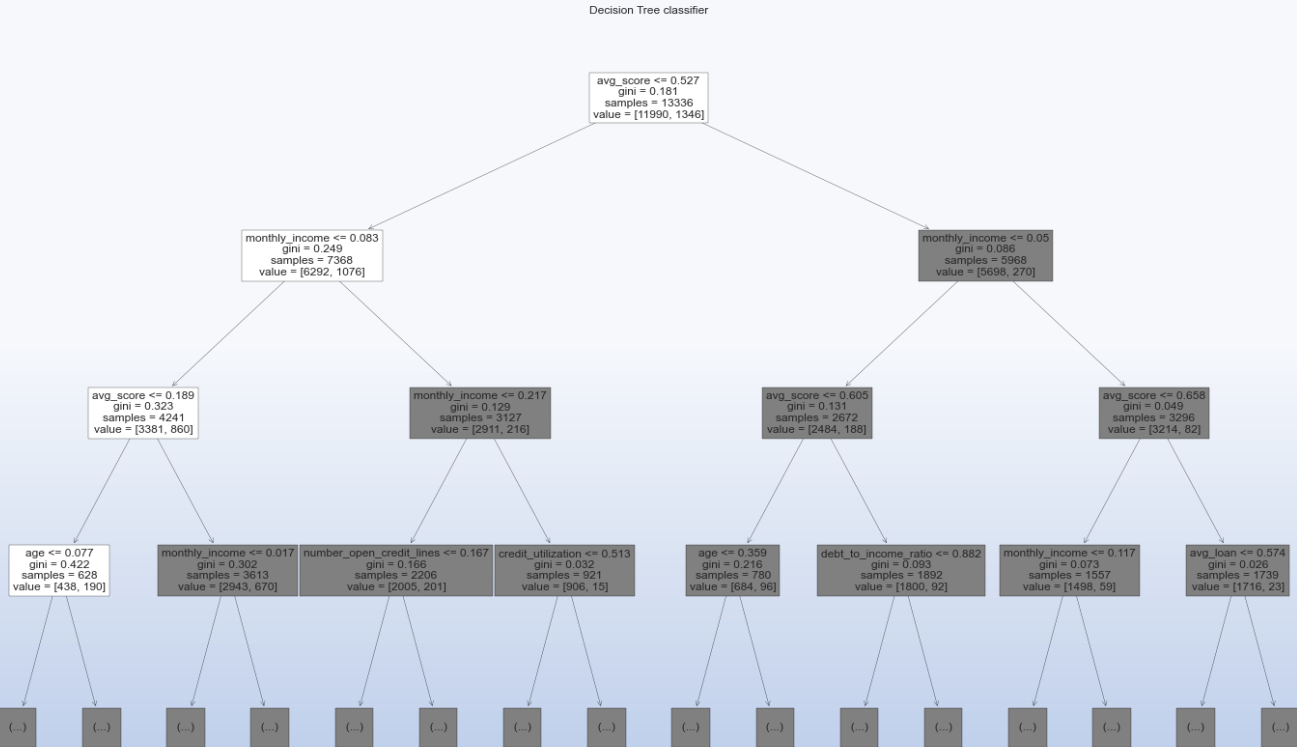
```
)
```

Table 6. Summary of models

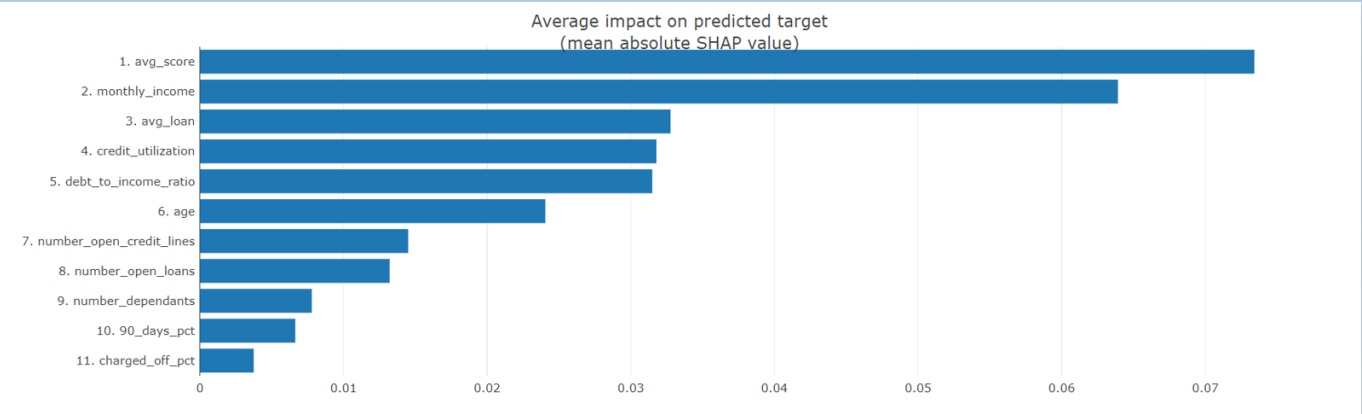
| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|----------|---------------------------------|----------|--------|--------|--------|--------|---------|---------|----------|
| dt | Decision Tree Classifier | 0.8134 | 0.5228 | 0.1579 | 0.1373 | 0.1464 | 0.0425 | 0.0428 | 0.0210 |
| nb | Naive Bayes | 0.8860 | 0.7197 | 0.0358 | 0.2121 | 0.0605 | 0.0289 | 0.0446 | 0.0070 |
| knn | K Neighbors Classifier | 0.8885 | 0.5852 | 0.0316 | 0.1945 | 0.0540 | 0.0270 | 0.0399 | 0.0730 |
| qda | Quadratic Discriminant Analysis | 0.8881 | 0.7190 | 0.0253 | 0.1935 | 0.0440 | 0.0181 | 0.0315 | 0.0080 |
| xgboost | Extreme Gradient Boosting | 0.8884 | 0.7054 | 0.0253 | 0.1750 | 0.0439 | 0.0184 | 0.0289 | 0.2990 |
| lightgbm | Light Gradient Boosting Machine | 0.8959 | 0.7181 | 0.0168 | 0.3655 | 0.0318 | 0.0212 | 0.0553 | 0.1280 |
| catboost | CatBoost Classifier | 0.8969 | 0.7248 | 0.0095 | 0.3683 | 0.0183 | 0.0122 | 0.0415 | 1.6990 |
| rf | Random Forest Classifier | 0.8975 | 0.6958 | 0.0063 | 0.4200 | 0.0124 | 0.0084 | 0.0381 | 0.2710 |
| gbc | Gradient Boosting Classifier | 0.8975 | 0.7365 | 0.0042 | 0.2000 | 0.0082 | 0.0051 | 0.0163 | 0.3570 |
| et | Extra Trees Classifier | 0.8964 | 0.6888 | 0.0042 | 0.2083 | 0.0082 | 0.0030 | 0.0131 | 0.1980 |
| lr | Logistic Regression | 0.8982 | 0.7418 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.6440 |
| svm | SVM - Linear Kernel | 0.8982 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0080 |
| ridge | Ridge Classifier | 0.8982 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0050 |
| ada | Ada Boost Classifier | 0.8977 | 0.7286 | 0.0000 | 0.0000 | 0.0000 | -0.0011 | -0.0049 | 0.1150 |
| lda | Linear Discriminant Analysis | 0.8981 | 0.7380 | 0.0000 | 0.0000 | 0.0000 | -0.0002 | -0.0011 | 0.0080 |
| dummy | Dummy Classifier | 0.8982 | 0.5000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0040 |

Creating and understanding the model

Table 7: Summary of models



| | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|------|----------|--------|--------|--------|--------|--------|--------|
| Fold | | | | | | | |
| 0 | 0.8298 | 0.5552 | 0.2105 | 0.1923 | 0.2010 | 0.1060 | 0.1061 |
| 1 | 0.8148 | 0.5049 | 0.1158 | 0.1100 | 0.1128 | 0.0095 | 0.0095 |
| 2 | 0.8030 | 0.5263 | 0.1789 | 0.1382 | 0.1560 | 0.0465 | 0.0470 |
| 3 | 0.8126 | 0.5130 | 0.1368 | 0.1226 | 0.1294 | 0.0247 | 0.0248 |
| 4 | 0.8041 | 0.5362 | 0.2000 | 0.1508 | 0.1719 | 0.0633 | 0.0641 |
| 5 | 0.8199 | 0.5124 | 0.1263 | 0.1237 | 0.1250 | 0.0247 | 0.0247 |
| 6 | 0.7899 | 0.5144 | 0.1684 | 0.1203 | 0.1404 | 0.0245 | 0.0249 |
| 7 | 0.8178 | 0.5299 | 0.1684 | 0.1495 | 0.1584 | 0.0567 | 0.0568 |
| 8 | 0.8264 | 0.5160 | 0.1263 | 0.1319 | 0.1290 | 0.0327 | 0.0327 |
| 9 | 0.8156 | 0.5194 | 0.1474 | 0.1333 | 0.1400 | 0.0370 | 0.0371 |
| Mean | 0.8134 | 0.5228 | 0.1579 | 0.1373 | 0.1464 | 0.0425 | 0.0428 |
| Std | 0.0112 | 0.0140 | 0.0309 | 0.0220 | 0.0247 | 0.0262 | 0.0263 |



Model Explainer

Positive class:

1

Download

- Feature Importances
- Classification Stats
- Individual Predictions
- What if...
- Feature Dependence
- Feature Interactions

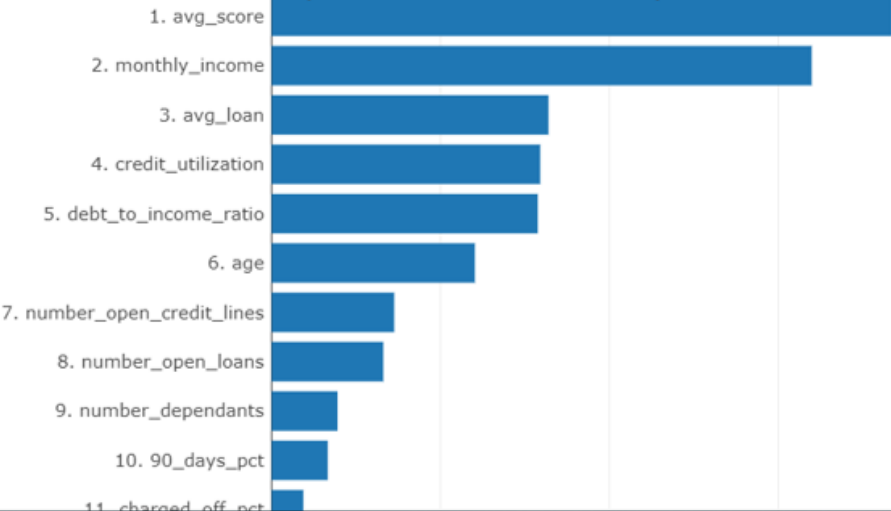
Shap Summary

Ordering features by shap value

Depth: Summary Type:

Aggregate

Average impact on predicted target
(mean absolute SHAP value)



Shap Dependence

Relationship between feature value and SHAP value

Feature:

avg_score

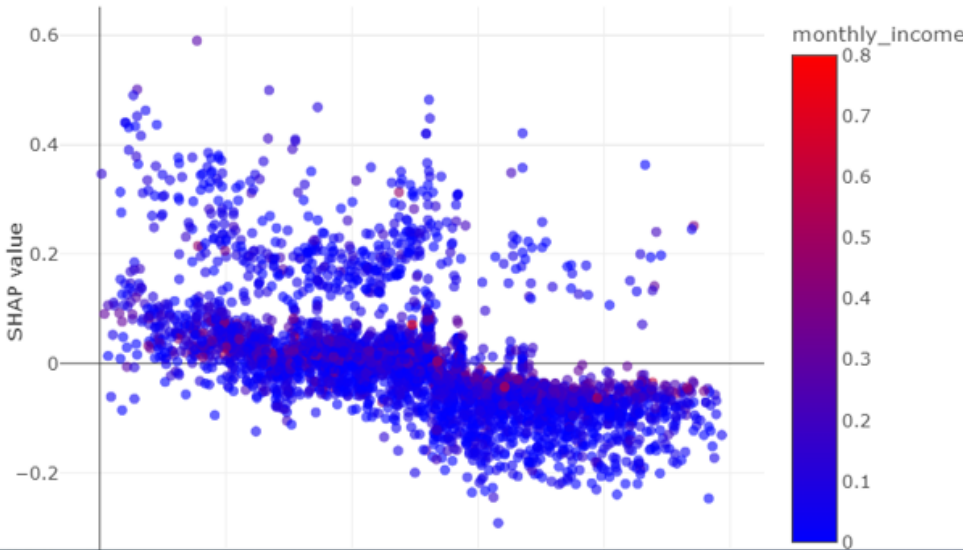
 Color feature:

monthly_income

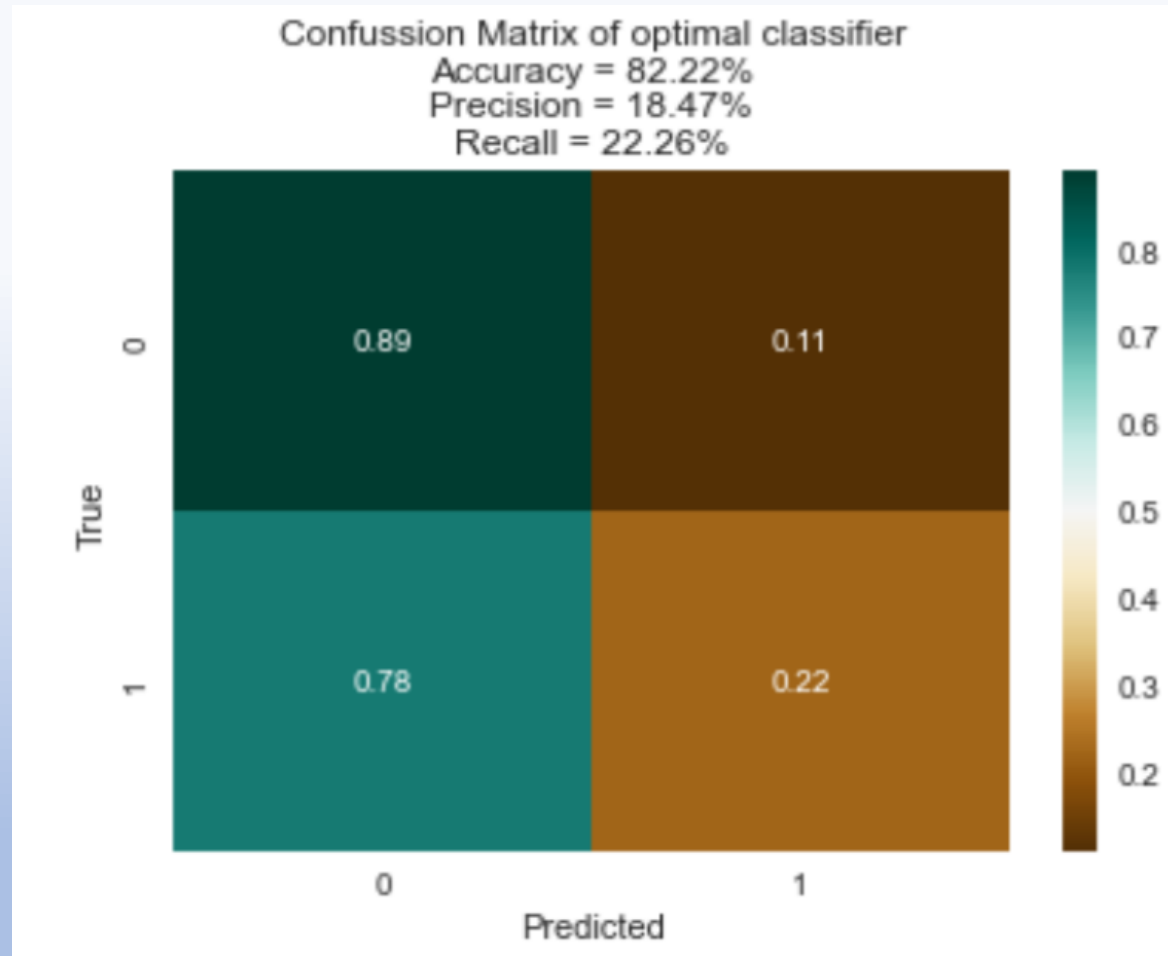
 Index:

Select...

Dependence plot for avg_score



Testing the model



10. EWC to alert current portfolio defaults

Based on our overall analysis we can expect a similar outcome when evaluating the target data provided of the current portfolio. The predictions can be found in the README file of the main repository, or clicking [here](#)

Projecting current portfolio

The current portfolio is composed of 4,168 open loans. The model is alerting about 489 potential defaults (11.73% of the total portfolio). However, these alerts have a 20% precision, meaning that around 391 are false alerts and 98 are real defaults. Additionally, since the model is able to correctly predict 20% of the real potential defaulters, we could approximate the total number of defaulters that the current portfolio has, which is around 489 ($98/0.2$), around 11% of the total portfolio. Coincidentally, because the precision and recall values are very close, the model will be able to predict correctly the number of potential defaulters.

On the other hand, the model will not send alerts for 3,679 loans (88.27% of the total portfolio), out of which 391 ($489 - 98$) are potential defaulters (11% of the non-alerts).

Case scenario

The proposal is simple, with the results described in the last section, the collections team can better allocate their resources in the following way.

The model will divide the total portfolio in 2 groups:

- Low maintenance portfolio (3,679 loans): These loans have a probability of default of 11% and can be treated with standard prevention.
- High maintenance portfolio (489 loans); These loans have a probability of default of 20%, which is almost twice that of the low maintenance group. This loans require intensive follow-up and attention.

Assumptions:

- Assume a scenario in which a collections firm provider offers intensive follow-ups, and Stride has budget to get up to 500 of this specialized follow-ups.
- Traditional standard prevention has a recovery rate of 20%, while intensive follow-up has a recovery rate of 50%.
- The cost of a standard prevention is USD 100 per loan, and intensive follow-up costs USD300.
- Using the data presented at the beginning about the loan market in the U.S. we can estimate that a portfolio with 4,168 student loans to be around USD92.5 million (4,168 37,000 0.6) (total loans *avg student loan* outstanding proportion)

Projections (See [Case scenario of collections dashboard](#))

Using all information available we get:

- Total portfolio: 4,168
- Model will produce 489 alerts
- Out of the 489 alerts, 98 will be true
- The portfolio contains 489 real defaulters, hence it also contains 3,679 good loans
- The probability that a randomly picked loan in the non-alert group is a default is 11%
- The probability that a randomly picked loan in the non-alert group is a good loan is 89%
- The probability that a randomly picked loan in the alert group is a default is 20%
- The probability that a randomly picked loan in the alert group is a good loan is 80%

Case summary figure shows the percentage of charged off % that the portfolio would have depending on what % of the 500 intense follow ups are allocated to the alert group.

EWC to alert current portfolio defaults : See report

Case Summary

| Scenario | | Random | | | Even | | | Optimal | |
|---------------|-------|--------|-------|-------|-------|-------|-------|---------|-------|
| Default resou | 0% | 10% | 20% | 40% | 50% | 60% | 80% | 90% | 100% |
| Charged-off % | 8.99% | 8.96% | 8.93% | 8.86% | 8.83% | 8.80% | 8.73% | 8.70% | 8.67% |

| Outcome | Reduction poin | Imrpovement |
|-----------------|----------------|-------------|
| 8.673% | -3.175 | -3.175% |
| Avg loan | \$37,000 | |
| Outstanding | 60% | |
| Portfolio level | \$92,529,600 | |
| Portfolio saved | \$2,937,601 | |

Optimization intense follow-ups

