CREDIT RISK MODELLING:

Credit Default Models

Brandon

TABLE OF CONTENTS

01

BUSINESS PROBLEM

BUSINESS PROBLEM

- Predict risk of default based on the profile of borrower.
  - Default/non-default Loans

- Business value created
  - Lender can factor the risk of default when deciding to disburse a loan

FRAMING ML SOLUTION TO THE PROBLEM

- Using historical loans to predict how *future loans* will perform
- Features (**x-input**) will be the metadata and profile of borrower (i.e. age, education, sex)
- Target output (**y-output**) will be default/non-default
- **Supervised, Binary Classification problem**

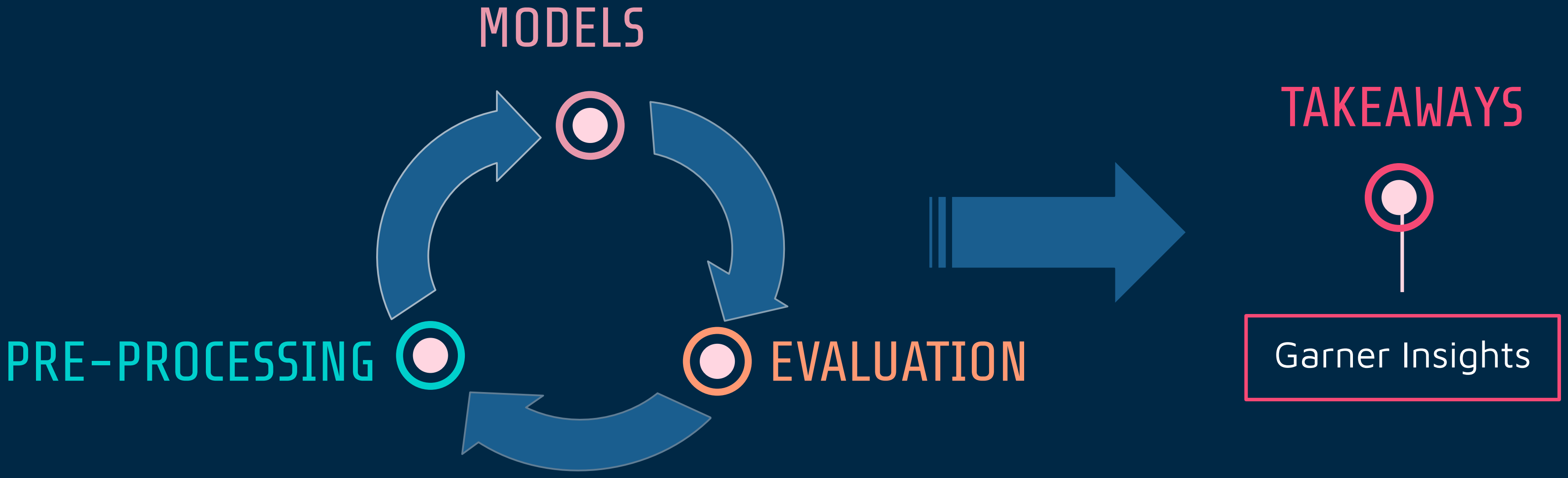
DATASET

- Dataset: Loan repayment schedule from **Indonesian Lender**
- 271K rows, 58 columns
- 80:10:10 (training-dev-test split)
  - Dev is used for finding best models (9 fold CV)
  - Test is used for final evaluation

02

DATA PRE-PROCESSING

OVERVIEW OF THE DEVELOPMENT PROCESS



Exploratory Data Analysis

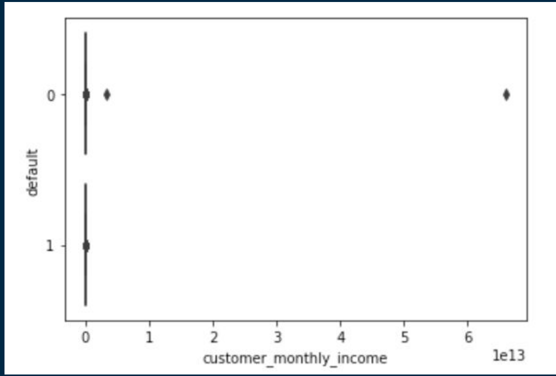
- Only **<u>14 columns</u>** selected → loan pre-approval stage
- Dropped rows with null values - 7% of total rows

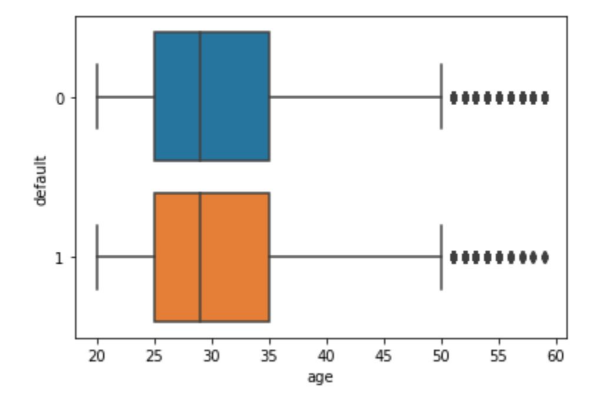
- New binary target variable column created using output column
  - Number of days past loan repayment **(DPD)** >= 90 → **Default (1)**
  - DPD < 90 → Non-Default (0)

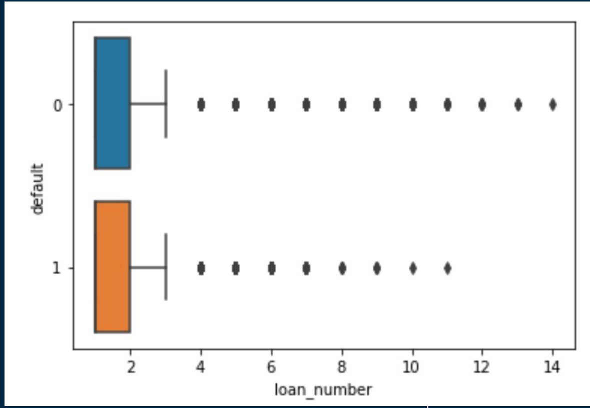
- Dataset is highly imbalanced
  - only **11% of values** were Default

Exploratory Data Analysis - Insights

- Numerical features don't discriminate default/non-default very well



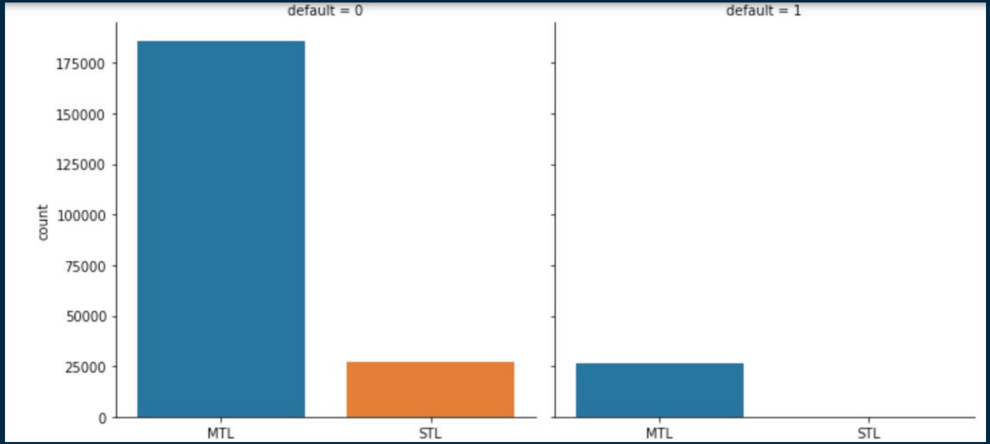




Exploratory Data Analysis - Insights

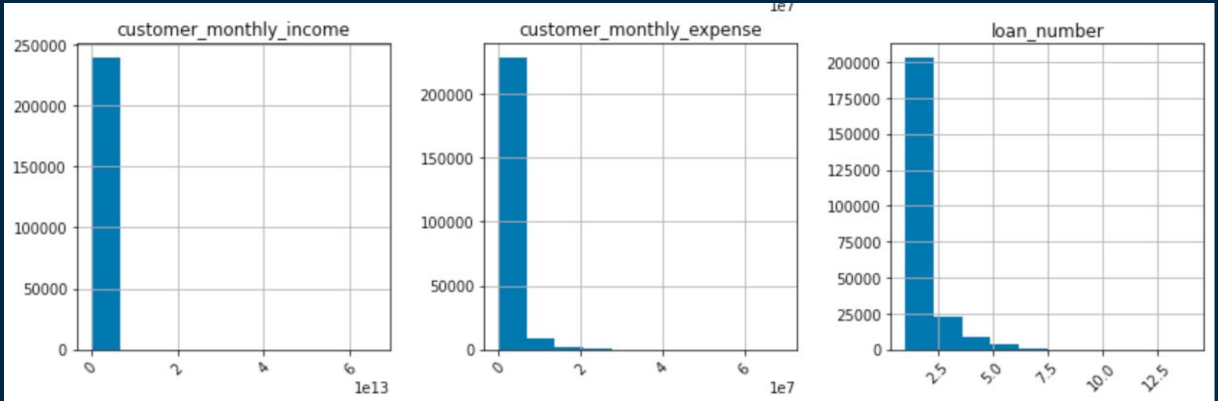
Pre-Processing

- Categorical variable (Product_Type) potentially strong variable



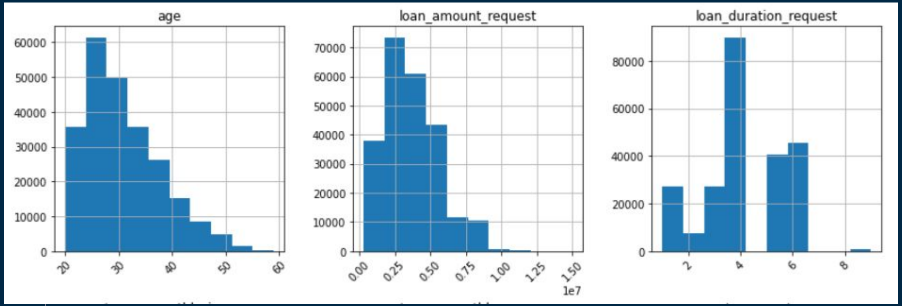
Exploratory Data Analysis - Insights

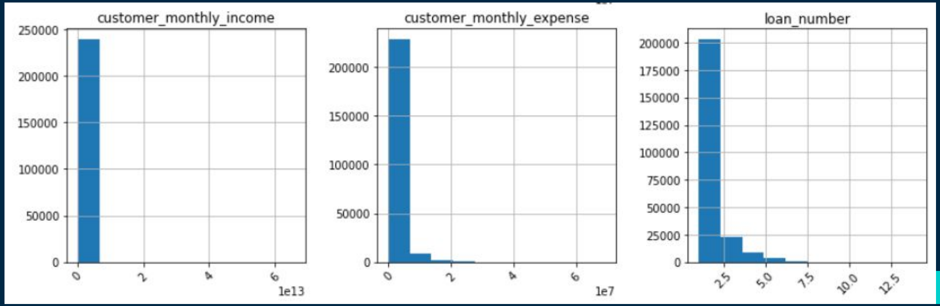
- Numerical features with skewed distributions



Scaling Data

- Tried Scaled data using StandardScaler
  - Numerical columns were normalized such that all standard deviations = 1
  - Speed up training
- Eventually not adopted because performance depreciated
  - Magnitudes are significant





Feature Selection

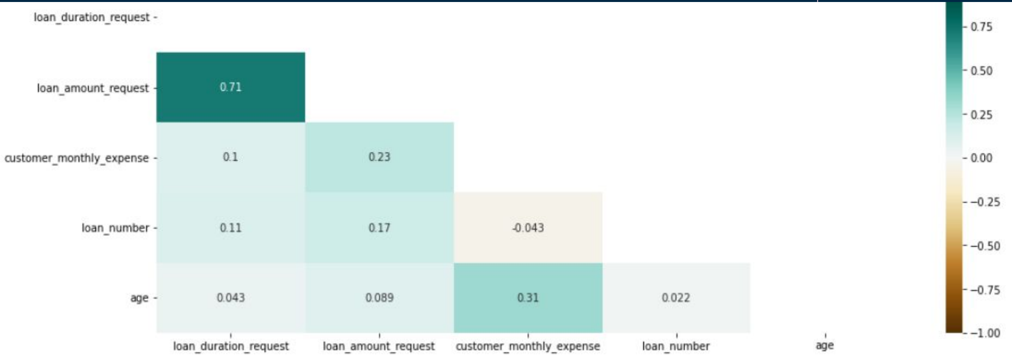
- Chi-Squared Test of Independence
  - Find dependence between the input categorical and target variables
  - Pick the **top 4 categorical variables** with most significant p-values
  - Dummy variables created for selected categorical variables

	Feature	p-value
0	product	0.000000
1	job_type	0.000000
2	job_industry	0.000000
3	last_education	0.000000

- ANNOVA F-Statistic
  - Find dependence between the input numerical and target variables
  - Pick the **top 5 numerical variables** with most significant p-values

	Numerical_Feature	F-Score	p values
0	loan_duration_request	802.760809	0.000000e+00
1	loan_amount_request	645.337406	0.000000e+00
2	customer_monthly_expense	370.051878	0.000000e+00
3	loan_number	197.147871	0.000000e+00
4	age	26.614304	2.486000e-07

- Pairwise Correlation
  - Used to find multicollinearity between input variables
  - 71% correlation between loan duration and loan amount, but did not remove the columns based on judgement



Oversampling

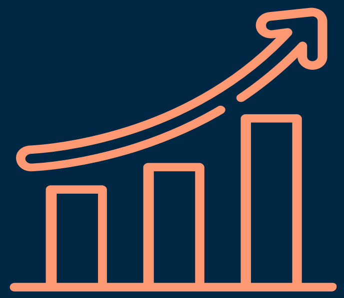
- SVMSMOTE to address imbalanced dataset
  - Creates synthetic observations of the minority classes

03

MODEL SELECTION

MODELS TESTED

1. Logistic Regression
2. XGBoost
3. Stacking (XGB + Logistic Regression)
4. Multi-Layer Perceptron



Base Models

- Before hyperparameter tuning, base model with standard parameters used on **oversampled samples**
- Evaluation based on average **F2 score** for 9-Fold CV

Model	Mean	Std Deviation
Logistic Regression	0.680	0.002
XGBoost	0.893	0.001
Stacking XGBoost and Log Reg	0.901	0.002
Multi-Layer Perceptron	0.610	0.001

MODEL 1: Logistic Regression

Rationale	Conventional model used in credit risk modelling
Strengths	Coefficient size and direction
Weaknesses	Assumption of linearity, expensive to obtain complex relationships
Params Tuned	Solver, penalty, C, class_weight
Config	{'solver': 'liblinear', 'penalty': 'l1', class_weight=None, 'C': 0.001,}

MODEL 2: XGBoost

Rationale	Modern solution, popular for classification
Strengths	Speed and efficiency, combines weak learners, reduces overfitting
Weaknesses	Sensitivity to outliers
Random Search Params	scale_pos_weight , gamma , subsample
Config	{'subsample': 0.5, 'scale_pos_weight': 8, 'gamma': 0.2}

MODEL 3: Stacking (XGB + Logistic Regression)

Rationale	Harnesses the strengths of multiple models
Strengths	Less likely to overfit
Weaknesses	Dependent on performance of base models
No CV	Limitations to computation for nested CV

MODEL 4: Multi-Layer Perceptron

Rationale	Powerful and effective for classification task
Strengths	Works well with large input Data Provides quick predictions after training
Weaknesses	Blackbox Potential overfitting
Random Search Params	solver, learning_rate, hidden_layer_sizes, alpha, activation
Config	{'solver': 'sgd', 'learning_rate': 'constant', 'hidden_layer_sizes': (10,), 'alpha': 0.05, 'activation': 'logistic'}

MODEL EVALUATION

1. F2-Score (Threshold-centric)
    a. Higher score desired
2. AU-Precision-Recall Curve (Discriminatonatory Power)
    a. Higher score desired
3. Brier Score (Probabilistic measure)
    a. Lower score desired

Final Evaluation:
Rank on each metric, select model with best ranks

EVALUATION TABLE

Model Evaluation	F2-Score	AUPR	Brier Score
Logistic Regression	0.32285	0.154736	0.186212
XGBoost	0.451269	0.200745	0.280801
Stacking	0.395933	0.200745	0.167932
Multi-Layer Perceptron	0.382572	0.555130	0.889739

COMPARISON

Model	F2-Score	AUPR	Brier Score
Logistic Regression	4	3	2
XGBoost	1	2	3
Stacking	2	2	1
Multi-Layer Perceptron	3	1	4

Final Evaluation: **Stacking (XGB -> Log Reg)** chosen

Comparable Analysis

- Dataset of consumer loans from Kenya

Model	Precision Score	Recall Score	F1_score	Accuracy	PR AUC
Logistic Regression	0.7778	0.0052	0.0104	0.7814	0.4006
Random Forest	0.4386	0.2058	0.2802	0.7680	0.4932
SVM	0.6373	0.0921	0.1609	0.7893	0.4643
Gradient Boosting	0.6012	0.1422	0.2300	0.7911	0.4658
MLP	0.5579	0.1369	0.2200	0.7869	0.4421

Table 4.1: Summary of the performance evaluation results of the models

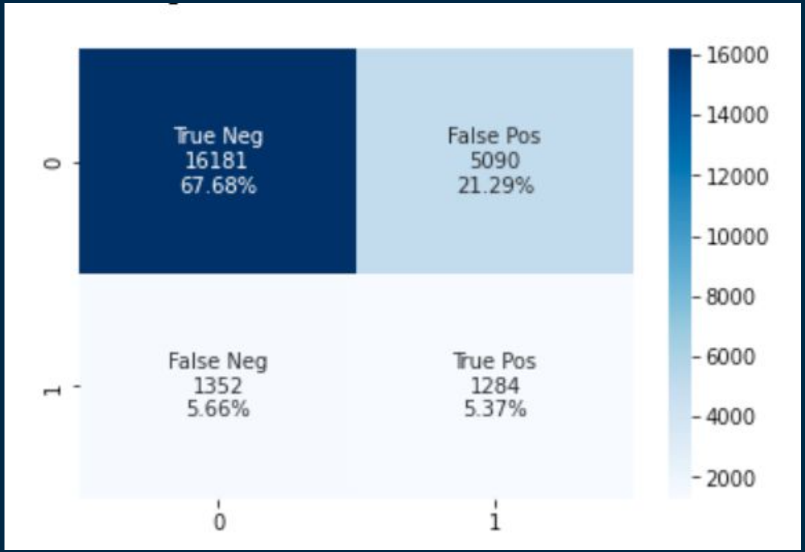
APPLICATION – Model as First Filter

Baseline: Predict 11% of "bad credit" - dataset makeup

With ML: Predict 20% "bad credit"

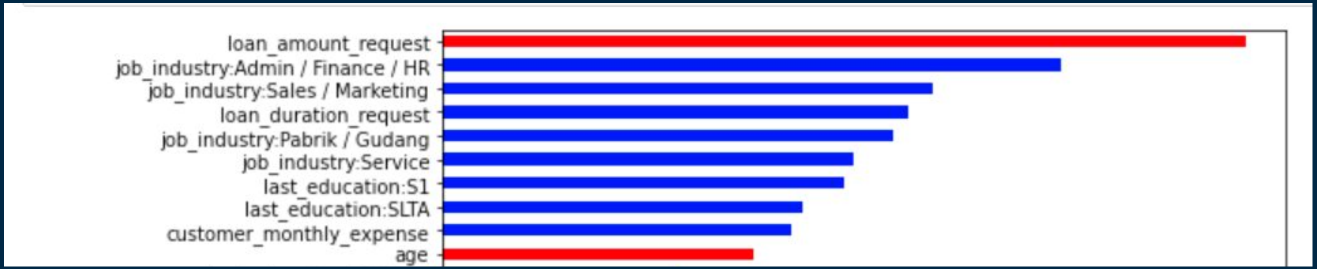
At selected threshold=0.52,

1. Precision: 0.20
   a. True Positive / Predicted Positive
2. Recall: 0.50
   a. True Positive / (True Positive + False Negative)



MOST IMPORTANT FEATURES

- Explaining a prediction using **SHAP**
  - **Red bars** indicate a positive correlation with bad credit
  - **Blue bars** indicate a negative correlation



FEATURE ATTRIBUTION

- Model Output: **Bad Credit Probability**
  - **Red bars** indicate features driving UP probability
  - **Blue bars** indicate features driving DOWN probability

```
True: 1 --> Pred: 1 | Prob: 0.6501620246019498
```

04

CONCLUSION

KEY CHALLENGES

1. Appropriate performance measure to optimize for
    a. Chose F2 because of focus on recall, costs higher for a false negative
    b. Final evaluation on THREE independent criteria

2. Choosing appropriate steps to improve performance
    a. Drop columns? Feature engineering?
    b. What is most relevant to business context

3. Model selection
    a. Which ensemble method?
    b. Which artificial neural network model?

KEY TAKEAWAYS

Garbage in, garbage out

- Quality of data could be due to geographical context
- Analysis done on a cluster-basis

Cluster-level analysis might be necessary

- Features don't discriminate well at global level
- Anomalies and skewed data affect performance

Complex models don't always perform the best

- Tendency to overfit despite regularisation
- More hyperparameters to decide
- Sacrifices on explainability

Recommendations - Future Work

Improving quality of data

- Review data collection processes
- Consider collecting new features

Custom Credit Rating Scoring

- Once satisfied with model, can use weights to build a custom credit score similar to FICO score

Anomaly detection

- Build an anomaly detection algorithm used as a second filter
- OneSVM

THANK YOU!