The City College of New York

Credit Risk

DSE I2100 - Applied Machine Learning

Laura Estaire, Artjola Meli, Shradha Godse, Wayne Lam

Project Final Presentation

Outline

- Project Statement
- The Dataset
  - Summary
  - Tables Granularity
  - Data Model Schema
- Data Preprocessing
- EDA
- Baseline Model
  - Models
  - Metrics
- Improvements to Model
- Final Model
- Conclusion

Problem Statement

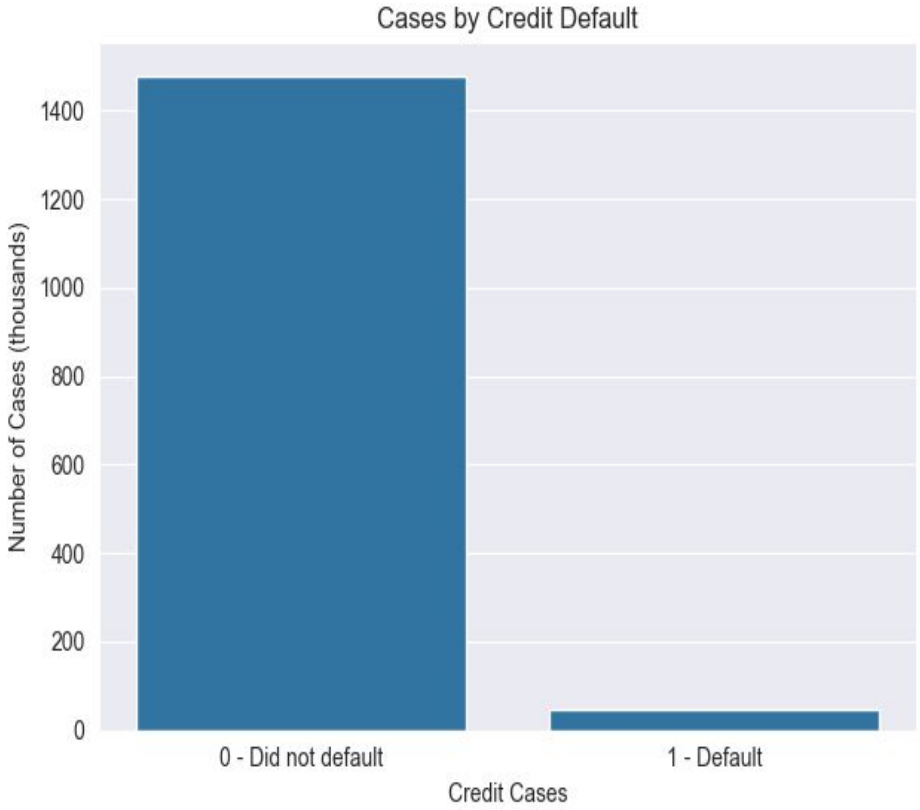
Credit Risk Model Stability

- Competition:
  - Kaggle and Home Credit
- Credit risk:
  - The possibility of a loss resulting from a borrower's failure to repay a loan or meet contractual obligations
- Goal:
  - Accurately determine which clients can repay a loan using financial history, current financial status, and socio-economic factors
- Stakeholders:
  - Lenders, Borrowers, Regulators, Investors

The Dataset

Summary

- **Binary Target:**
  - Default: positive class, 1
  - No default: negative class, 0
- **Cases: >1.5M**
  - Negative class: ~1.45M
  - Positive class: ~50K
- **Imbalance:**
  - Ratio:
  - Missing values: 92%
- **Features: >400**
  - Mix of data types
  - Examples: industry of employment, number of clients who have used the same mobile number
- **Sources: 8**
  - Number of tables: 17
  - Data depth: Static and transactional



The Dataset

Granularity: Depth 0, Depth 1, and Depth 2

- Indices:
  - *case_id* - specific loan case
  - *num_group1* - applicant or associate
    - 0 means applicant
    - All else means non-applicant
  - *num_group2* - additional entries
- Depth 0:
  - Data directly tied to a single case
- Depth 1:
  - Each case tied to historical record
  - Includes associated parties
  - Additional *num_group1* index
- Depth 2:
  - Each case tied to historical record
  - Includes associated parties
  - Additional *num_group1* and *num_group2* indices

D0

Case ID	App. Count	Credit Type
1	5	“CAL”
2	3	“REL”

D1

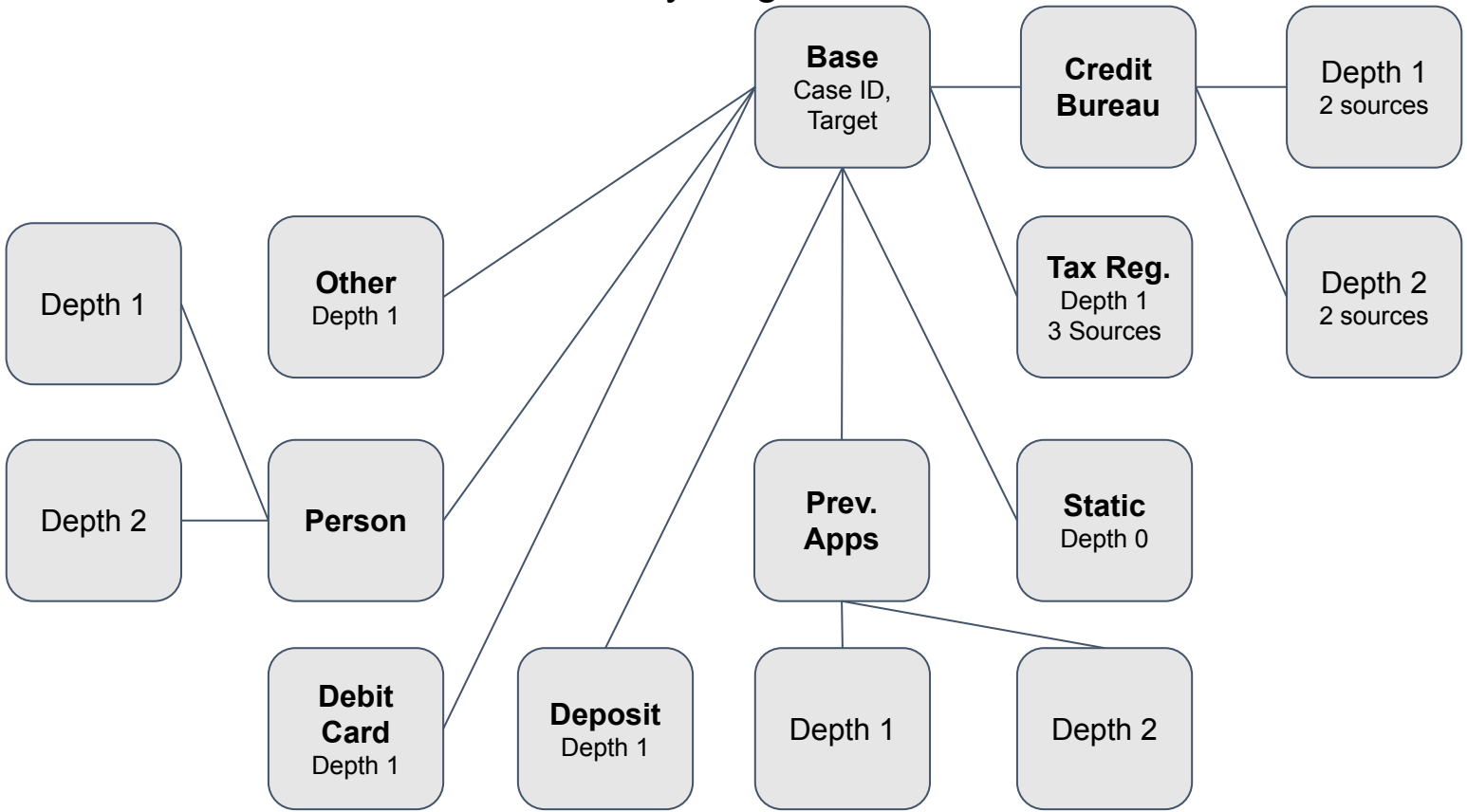
Case ID	N.G. 1	Cred. St.	Cred. End
1	0	2020-8-1	2021-9-6
1	1	2016-1-5	2018-8-7

D2

Case ID	N.G. 1	N.G. 2	Ph. Ty.	Status
1	0	0	“mobile”	“open”
1	0	1	“home”	“open”

The Dataset

Data Model Schema: Case ID ties everything



Data Pre-Processing

Aggregate the dataset to the case ID level

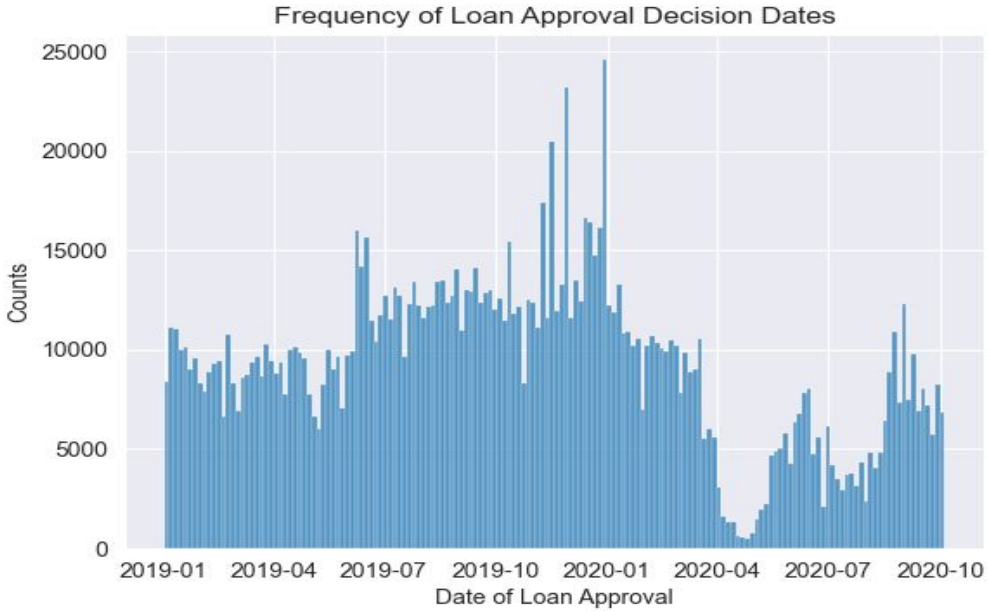
Aggregations

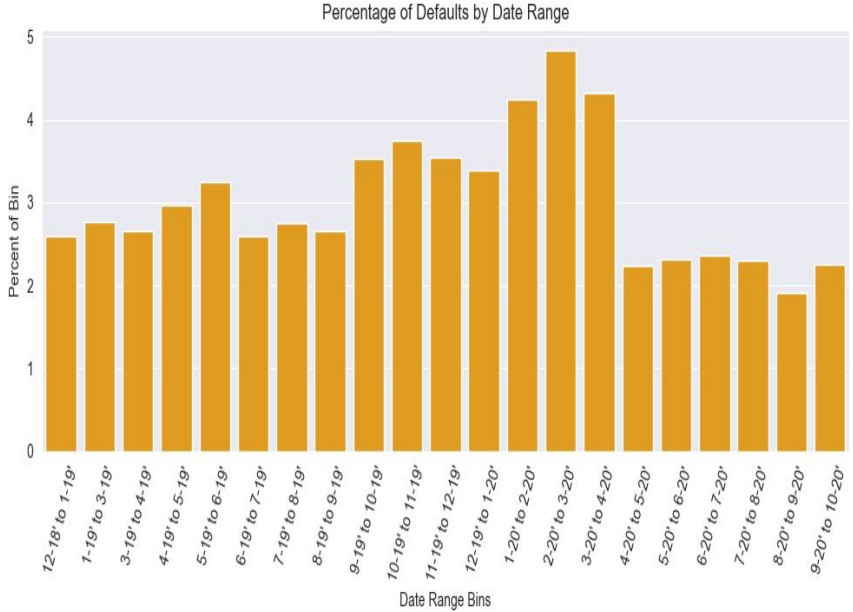
- By *case_id* and *num_group_1*
  - *num_group1* = 0 by itself
  - *num_group1* != 0 together
- Numerical
  - Mean, min, max, sum, median
- Dates
  - Min, max, distinct
- Categorical
  - Mode
  - Encoded with frequency and binary encoding

Case ID	N.G. 1	N.G. 2	Num_1
1	0	0	5
1	0	1	10
1	1	0	4

Case ID	N.G. 1	Num_1_min	Num_1_max
1	0	5	10
1	1	4	4

EDA





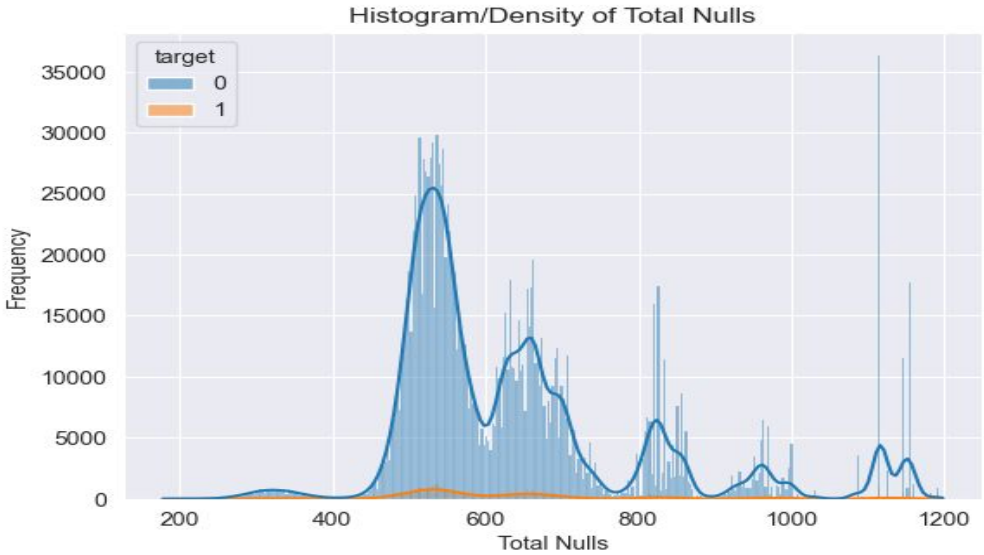
Date of cases

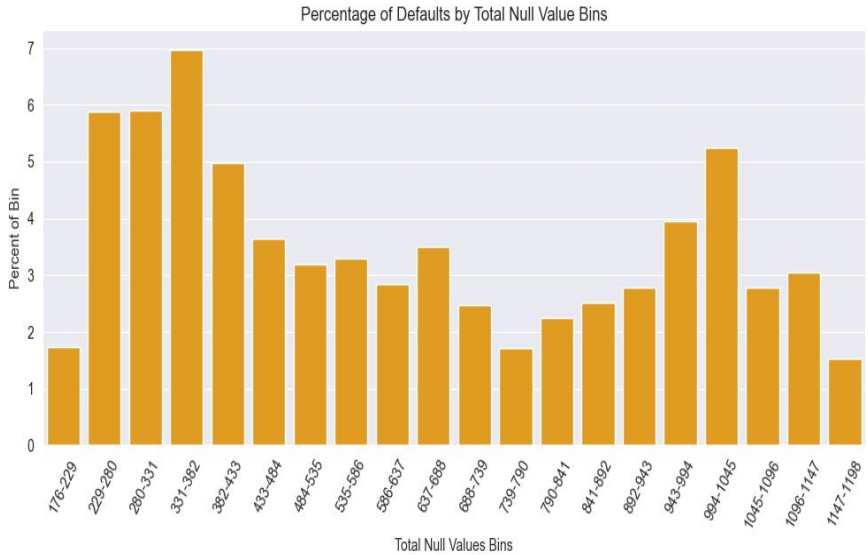
- There is a significant decline in April 2020, COVID-19

Defaults by date

- Slight increase in percentage of case defaults in the late-2019 to early-2020

EDA





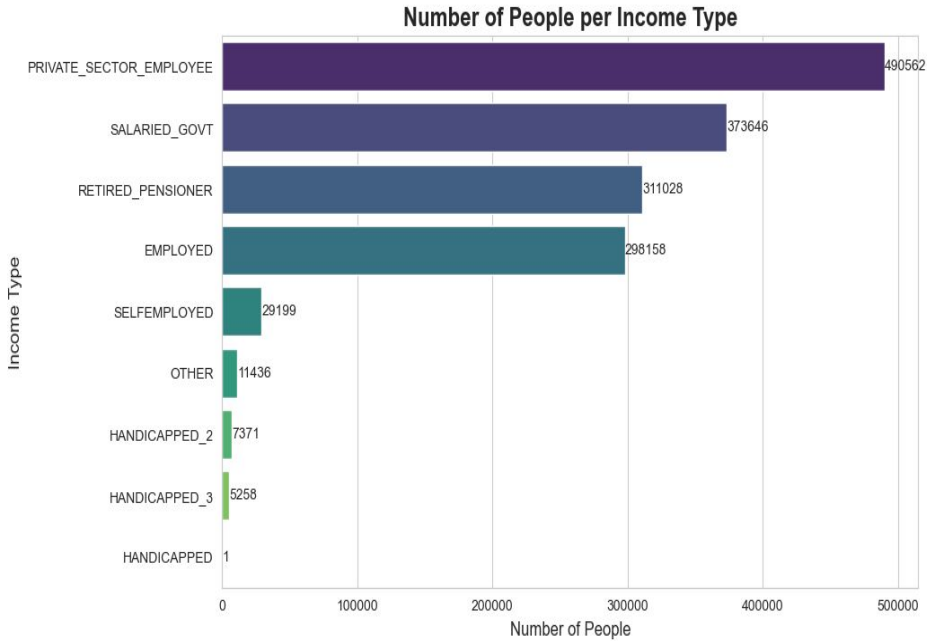
Total nulls by *case_id*

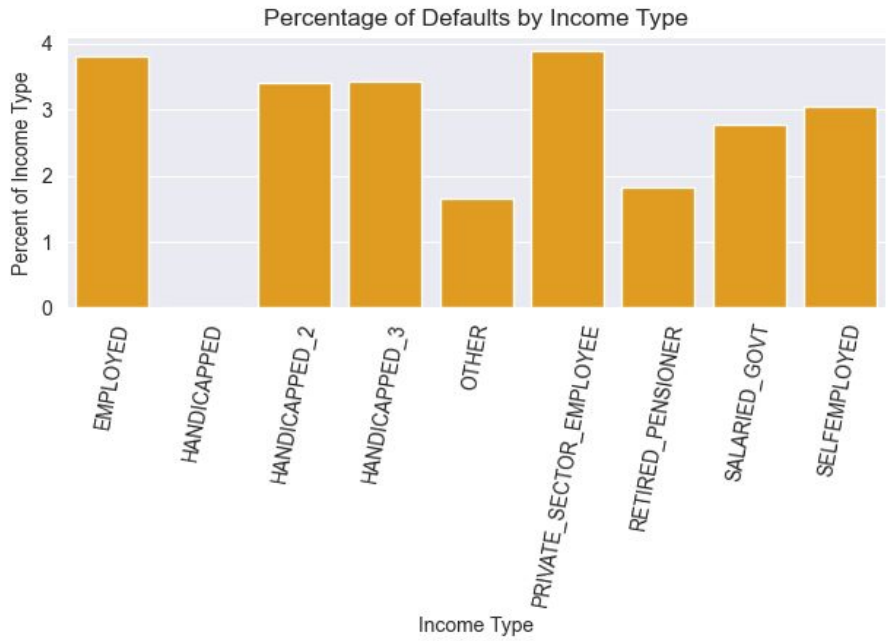
- Appears to be several clusters

Defaults by total null values

- Slight pattern on likelihood of defaults based on total null values

EDA

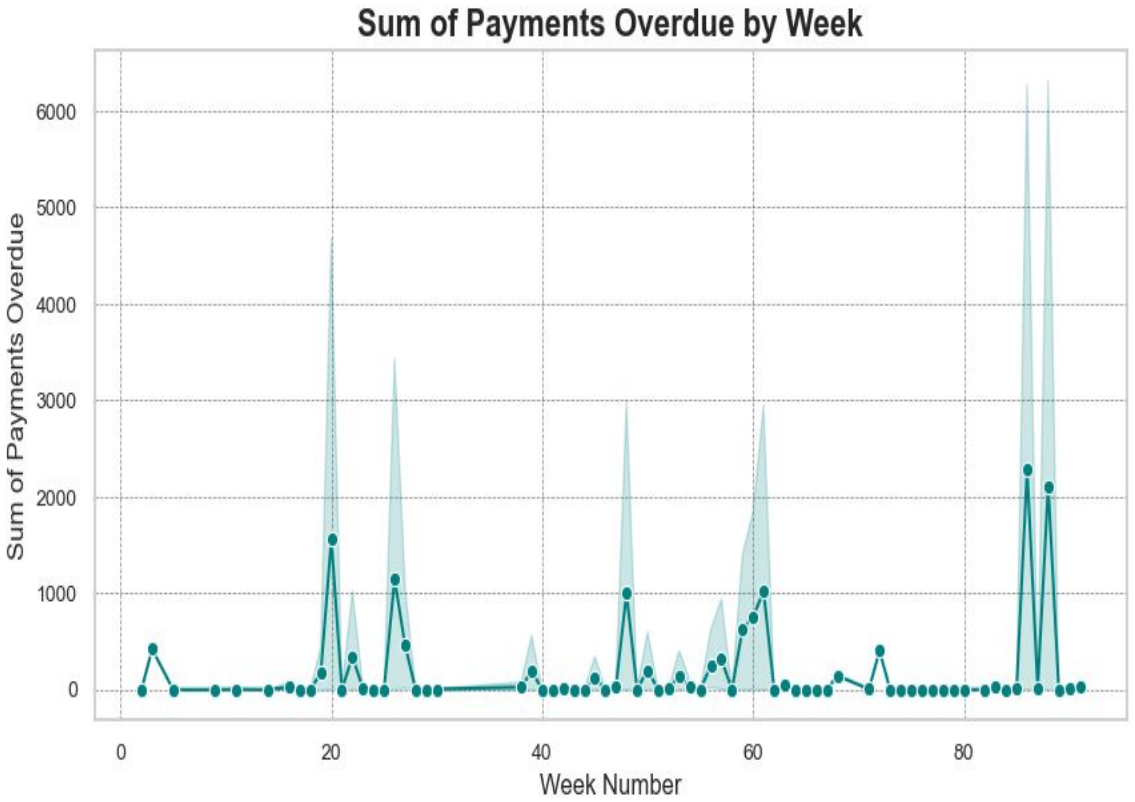




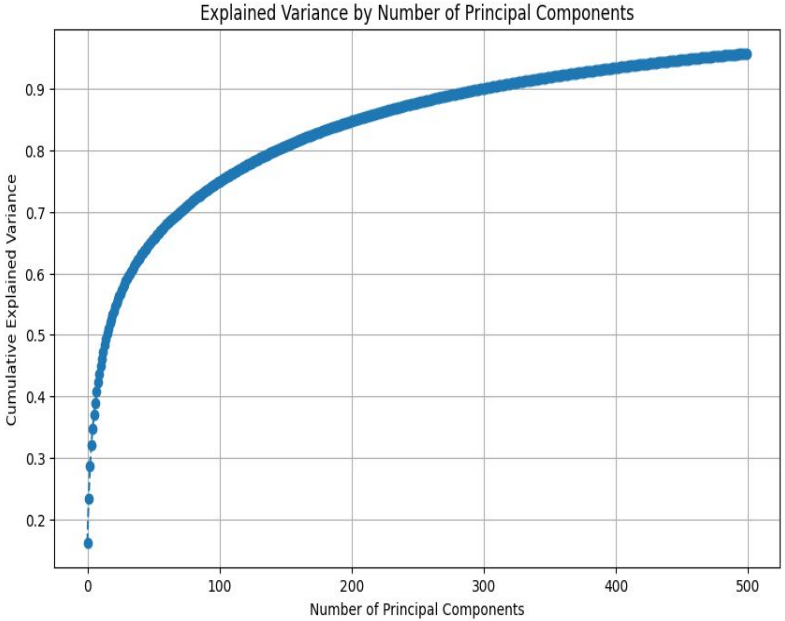
- Income types
  - Differences in default rates in income types
  - Private sector employees have highest number and higher default rate amongst all.

EDA

- General Observations
  - Sporadic Peaks
    - Significant spikes in overdue payments at weeks 20, 50, and 80
  - General Low Level of Overdue Payments
  - Significant Increase Towards the End

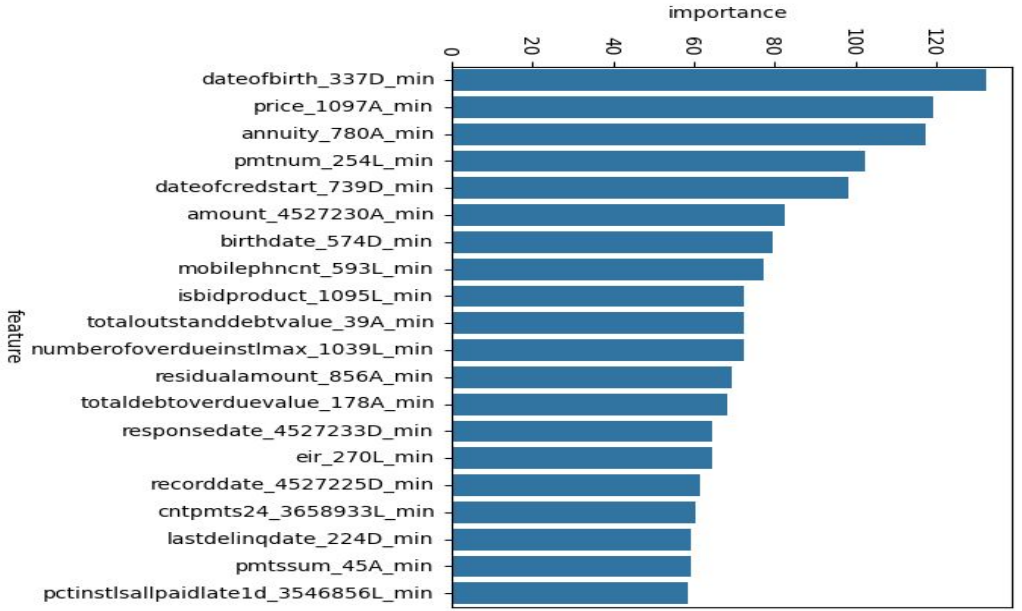


EDA



High complexity: PCA: **>400**

- components required to explain 95% variance
- “Elbow” point around 50-100 components → principal components capture most of variance before diminishing returns



Feature importance from LightGBM

- Aggregated data held a great deal of importance
- Mins dominated top 20 most important features

Main Barriers in the project

- **Domain Knowledge**
  - Selection of relevant features
  - Imputing missing values with meaningful substitutes
- **Missing Values**
  - 92% missing values
- **Class Imbalance**
  - Highly imbalanced target data
    - 97% negative class (clients who repaid the loan)
    - 3% positive class (clients who defaulted on their loan)
- **Class overlap**
  - No major patterns during the initial EDA that make both classes distinct or separable

Baseline Models

Establish Baseline Performance Metrics for future comparisons

- **Random Weighted Model**
  - predict the target label based solely on the average percentage (3% positive) distribution of classes in the entire dataset
- **Project Baseline**
  - pre-processed data and missing values mean imputation
  - The best of Logistic Regression, Random Forest or LightGBM on default parameters

Baseline Models - Metrics

Establish Baseline Performance Metrics for future comparisons

Metrics	Metrics Meaning	Random Weighted		Project Baseline	
AUC	Model's ability to distinguish between classes	0.50	No better than random guessing	0.79	56% improvement in AUC. Influenced class imbalance
Accuracy	Proportion of all correct predictions (both true positives and true negatives)	0.74	Influenced by the class distribution in the dataset. Predicting majority class.	0.78	High accuracy, though influenced by class imbalance.
Precision	Proportion of true positives among all positive predictions	0.24	Low. Does not have any mechanism to prioritize true positives over false positives	0.56	Better predicting positives cases with the trade-off of increase in FP.
Recall	Proportion of actual positives that are correctly identified by the model	0.03	Almost unable to identify true positives	0.47	Still major challenges in capturing all true positives. Proportional to the class distribution.
F1	Harmonic mean of precision and recall	0.05	Low trade-off between precision and recall	0.51	Better trade-off.

Initial Goal: Build a robust machine learning model capable of effectively predicting true positives in a highly imbalanced dataset with significant missing values.
—> Final Goal: **Improve the Project Baseline Model performance.**

Improvements to Boost Model Performance

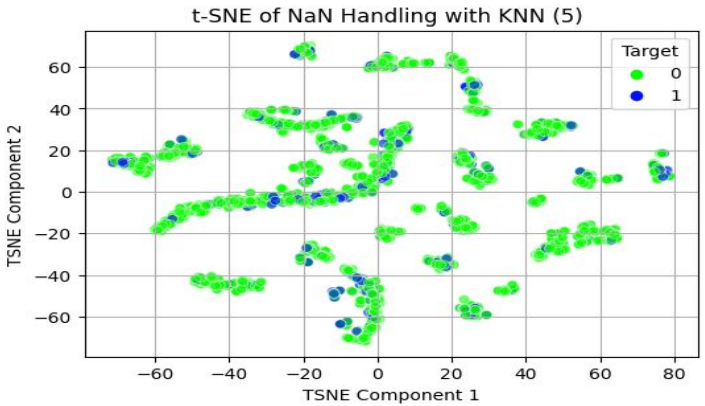
Handling Missing Values (MV) through Imputation

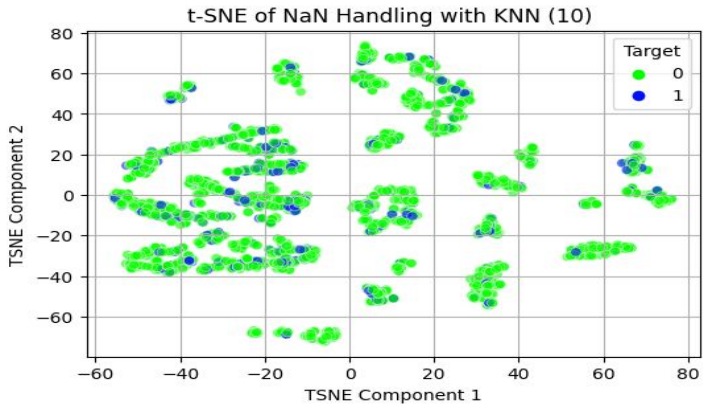
Methods:

- Mean Imputation (baseline model)
- LightGBM's Native Handling
- KNN Based Imputation (5 and 10 neighbours)
- Mean Imputation and Binary Flag for MV
- Median Imputation and Binary Flag for MV

Exploring Patterns Post-Imputation visually with t-SNE: Classes in separate clusters = better!

- KNN Based Imputation (5 and 10 neighbours)

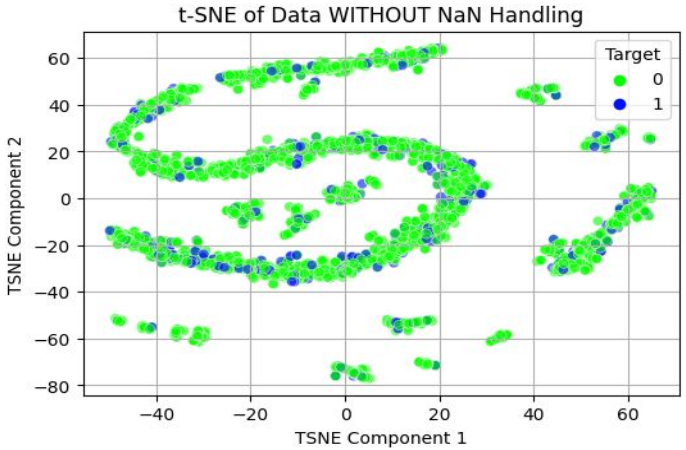


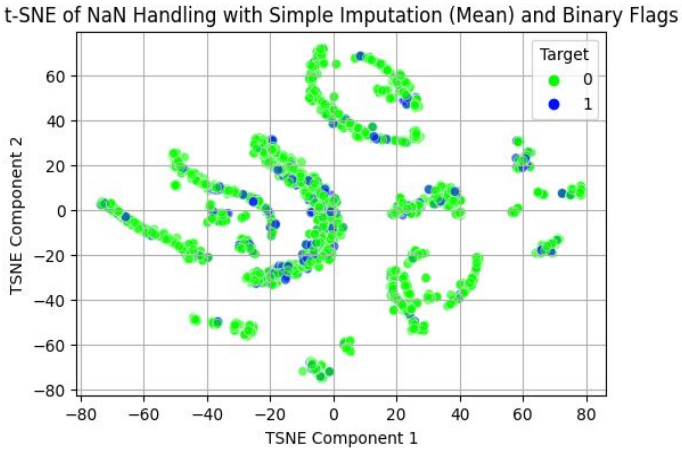


Improvements to Boost Model Performance

Handling Missing Values (MV) through Imputation

Exploring Patterns Post-Imputation visually with t-SNE





Improvements to Boost Model Performance

Handling Missing Values (MV) through Imputation vs Model Baseline

Methods:

- Mean Imputation (Same as Model Baseline)
- LightGBM Native Handling ≈ Accuracy; ≈ F1
- KNN Based Imputation (5 and 10 neighbours) ≈Accuracy; ↓↓(30%) F1
- Median Imputation and Binary Flag for MV ≈ Accuracy; ↓(25%) F1
- Mean Imputation and Binary Flag for MV ≈Accuracy; ↑(19%) F1

- Including the Binary Flag provides more information, capturing total null values by case.
- Mean imputation yields better results than median.
- Median is more robust to extreme values, which are useful for predictions.

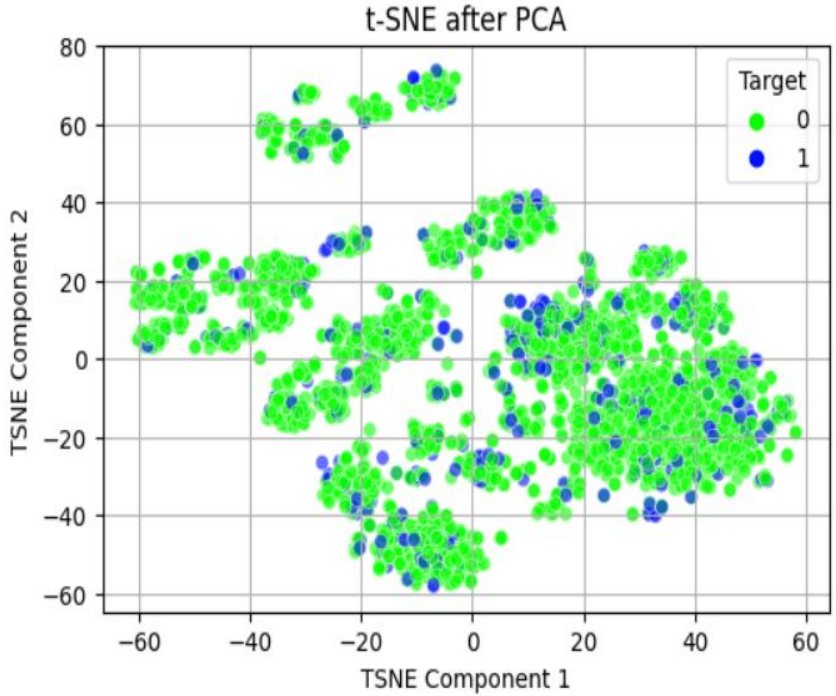
Improvements to Boost Model Performance

Dimensionality Reduction

PCA

Observation:

- Several distinct clusters are scattered throughout the space.
- No clear separation between the two classes, indicating limited improvement in predictive power.



Improvements to Boost Model Performance

Dimensionality Reduction

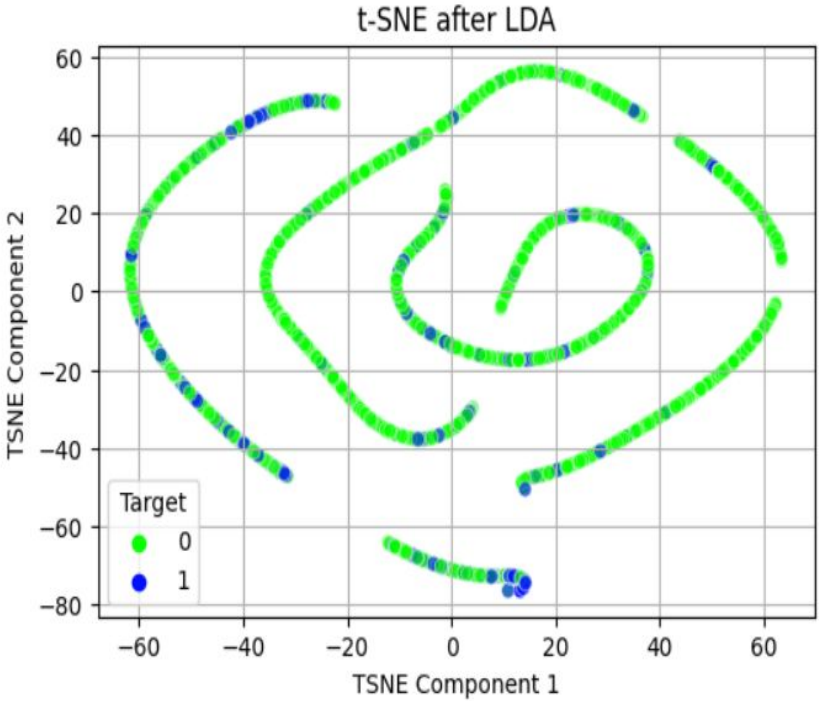
LDA

Observation:

- Classes are more distinctly grouped.
- Clearer distinction improves recall and precision.

Comparison to Model Baseline:

- ≈Accuracy ; ↑(52%) F1



Improvements to Boost Model Performance

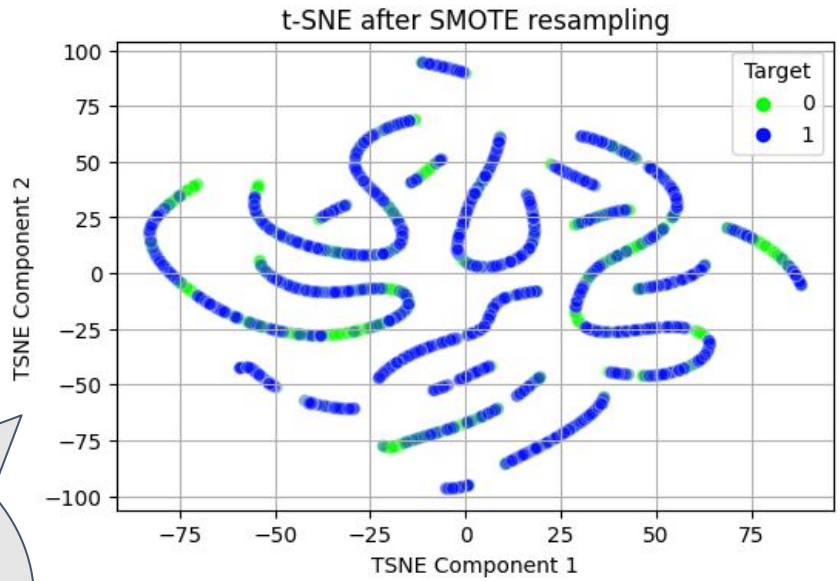
Addressing Class Imbalance with SMOTE

SMOTE

- Generated synthetic samples for minority class

Comparison to Model Baseline:

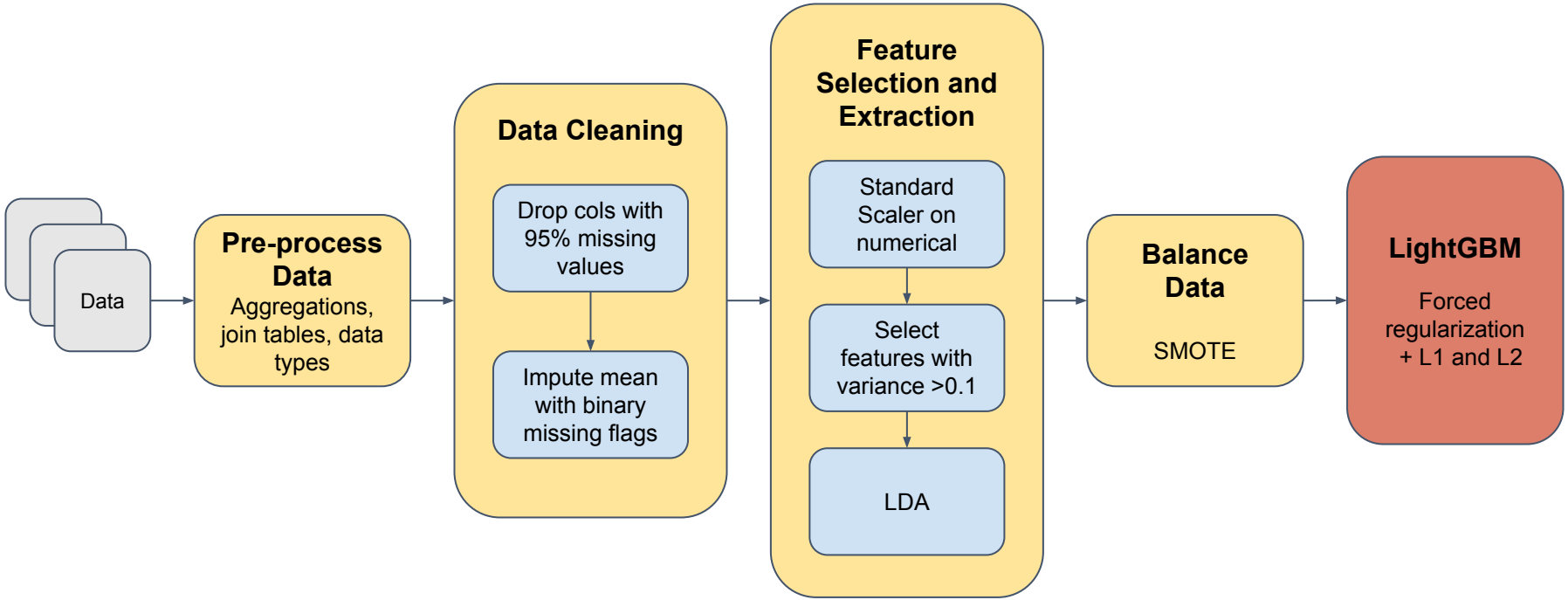
- ↓↓(25%) Accuracy; ↑↑↑(20%) F1



Enhances recall and precision by providing more minority class examples.

Final Model

Combining the previous steps



Final Model

Metrics Comparison to Baseline Project Model Performance and Interpretation - Test Set

Metrics	Project Baseline	Final Model	
AUC	0.75	0.75	Roughly **equivalent**.
Accuracy	0.93	0.56	Has **decreased**, which is expected in **imbalanced datasets** when improving recall.
Precision	0.09	0.04	The drop in precision indicates an increase in false positives, which can be attributed to the model's sensitivity to the synthetic minority samples generated by SMOTE, potentially leading to overfitting to the minority class characteristics.
Recall	0.25	0.79	**More effective at identifying actual positive cases.**
F1	0.13	0.07	**Worse overall balance between precision and recall**.

Conclusion and Project Lessons

Project Modeling Conclusions

- **Imputation:** Binary flags for missing data added valuable information.
- **Dimensionality Reduction:** LDA improved recall and precision by enhancing class separability.
- **Re-Sampling:** SMOTE balanced the dataset and reduced model bias.
- **Regularization:** Controlled overfitting with model parameters (reg_alpha, reg_lambda).
- **Accuracy:** Still low, not ready for production due to high false negatives.

Project Lessons:

- **Data Quality:** Real-world data is often messy and poorly documented.
- **Modeling Goal:** Achieving marginal gains can be more realistic than aiming for perfection.

Contributions

- Project Diary (link)