# Case Study
# Classification Pipeline with SMOTE and Evaluation

Niyazi Sorkunlu

# Outline

| | |
|---|---|
| Data | 10000  Records 32 columns<br>Identify the model can be generated from the available data. |
| Model | Comparable models |
| Evaluation | Model performances |

# Understanding Data

Lending Club Data has 10K rows and 32 features.

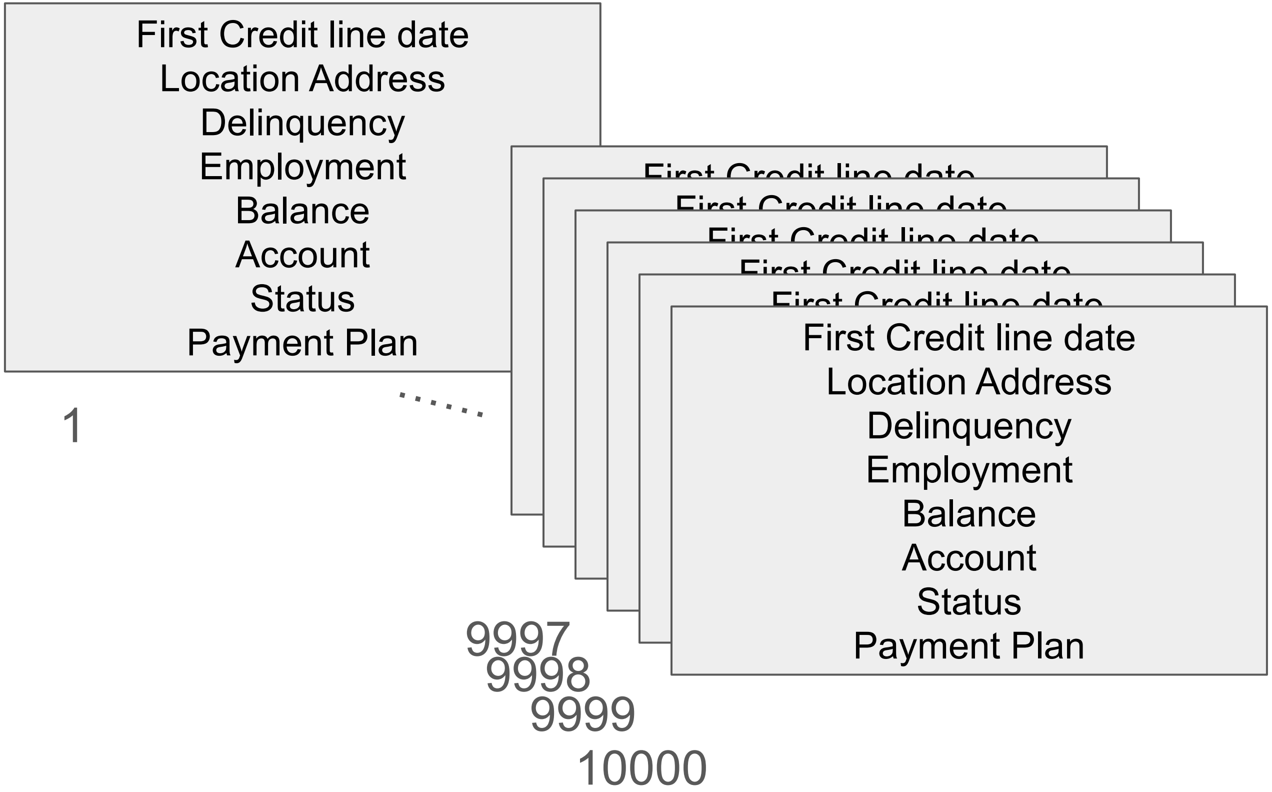
Financial information

Credit use history

Default risk data

Identify data points for credit worthiness.

10,000 Data points as such



# Peek into the Data

```
df1.head()
```

| | Id | is_bad | emp_title | emp_length | home_ownership | annual_inc | verification_status | pymnt_plan | Notes | purpose_cat | purpose | zip_code | add |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | Time Warner Cable | 10 | MORTGAGE | 50000.0 | not verified | n | NaN | medical | Medical | 766xx | |
| **1** | 2 | 0 | Ottawa University | 1 | RENT | 39216.0 | not verified | n | Borrower added on 04/14/11 > I will be using... | debt consolidation | My Debt Consolidation Loan | 660xx | |
| **2** | 3 | 0 | Kennedy Wilson | 4 | RENT | 65000.0 | not verified | n | NaN | credit card | AP Personal Loan | 916xx | |
| **3** | 4 | 0 | TOWN OF PLATTEKILL | 10 | MORTGAGE | 57500.0 | not verified | n | NaN | debt consolidation | Debt Consolidation Loan | 124xx | |
| **4** | 5 | 0 | Belmont Correctional | 10 | MORTGAGE | 50004.0 | VERIFIED - income | n | I want to consolidate my debt, pay for a vacat... | debt consolidation | consolidate | 439xx | |

# Missing Part of the data per column name

```
0 0 Id
1 0 is_bad
2 592 emp_title
3 0 emp_length
4 0 home_ownership
5 1 annual_inc
6 0 verification_status
7 0 pymnt_plan
8 3231 Notes
9 0 purpose_cat
10 4 purpose
11 0 zip_code
12 0 addr_state
13 0 debt_to_income
14 5 delinq_2yrs
15 5 earliest_cr_line
16 5 inq_last_6mths
17 6316 mths_since_last_delinq
18 9160 mths_since_last_record
19 5 open_acc
20 5 pub_rec
21 0 revol_bal
22 26 revol_util
23 5 total_acc
24 0 initial_list_status
25 32 collections_12_mths_ex_med
26 0 mths_since_last_major_derog
27 0 policy_code
28 5 earliest_cr_num_months
29 0 boolean_list_status
30 0 bool_pymnt_plan
31 0 bool_list_status
```

Out of 10000 data records some features are missing ⅓ of data, some others are missing ⅔ of it.

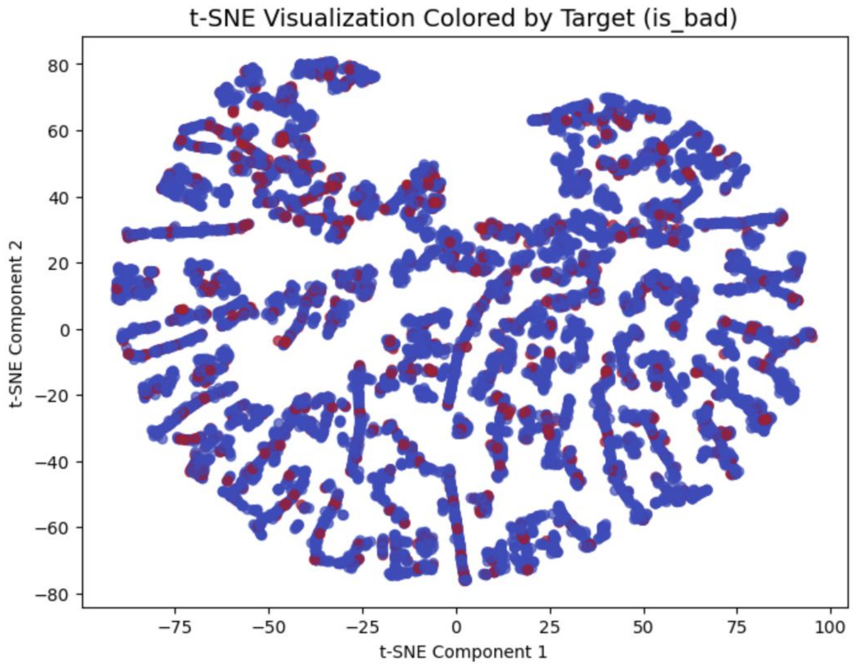
# Imbalanced Data

- Total of **10,000 rows** in dataset. .
- Out of these, **8,705 rows are ‘good’ (is_bad = 0)** and **1,295 rows are ‘bad’ (is_bad = 1)**.
- This means the dataset is **imbalanced**.
    - ‘good’ cases are **much more frequent** than the ‘bad’ cases **~ 7 times more.**
    - Training a model on this data **without adjustment**, it might **mostly predict ‘good’**.

# t_SNE Representation

Think of t-SNE like a **camera zooming out**:

- You start with a 54-dimensional world way too complex to see
- t-SNE “zooms out” and flattens it into **a 2D picture**, while **keeping the relationships intact**




Mapping 54 features to 2 helps visualize patterns, since we lose most of the information, accurate classification in just 2 dimensions is not an easy task for classifier.

# Feature: Months Since First Credit

Shows **how long it has been since the customer opened their first credit account.**
Calculated from `earliest_cr_line` and today's date.
Measured in **months** for each record.
Helps the model understand **credit history length.**

```
today = datetime.now()
df1['earliest_cr_num_months'] = 0
for i in range(df1.shape[0]):
    start_date = pd.to_datetime(df1.loc[i, 'earliest_cr_line'], format='%m/%d/%y')
    df1.loc[i, 'earliest_cr_num_months'] = (today.year - start_date.year) * 12 + (today.month - start_date.month)
```

| | earliest_cr_line | earliest_cr_num_months |
|---|---|---|
| **0** | 12/1/92 | 395.0 |
| **1** | 11/1/05 | 240.0 |
| **2** | 6/1/70 | 665.0 |
| **3** | 9/1/82 | 518.0 |
| **4** | 10/1/99 | 313.0 |
| **...** | ... | ... |
| **9995** | 9/1/01 | 290.0 |
| **9996** | 5/1/00 | 306.0 |
| **9997** | 12/1/89 | 431.0 |
| **9998** | 3/1/99 | 320.0 |
| **9999** | 9/1/00 | 302.0 |

# Categorical Variables

`home_ownership` Customer's housing situation

- Categories: **MORTGAGE, RENT, OWN, OTHER, NONE**

`verification_status` Income verification

- Categories: **not verified, VERIFIED - income, VERIFIED - income source**

`pymnt_plan` Whether the customer has an active payment plan

- Categories: **y / n**

`policy_code` → Internal policy identifier
Categories include:

- **Personal purposes:** medical, wedding, vacation, major purchase, home improvement, car, educational, house
- **Debt-related purposes:** debt consolidation, credit card
- **Small business variants:** e.g., `other small business`, `debt consolidation small business`, `credit card small business`, `home improvement small business`, etc.
- **Other:** renewable energy, moving

Total of **26 categories**, including small business-specific purposes.

# What is One-Hot Encoding

| home_ownership | MORTGAGE | RENT | OWN |
|---|---|---|---|
| MORTGAGE | 1 | 0 | 0 |
| RENT | 0 | 1 | 0 |
| OWN | 0 | 0 | 1 |

# One-Hot Encoding

| purpose_cat | debt_consolidation | credit_card | car | wedding | medical |
|---|---|---|---|---|---|
| debt_consolidation | 1 | 0 | 0 | 0 | 0 |
| credit_card | 0 | 1 | 0 | 0 | 0 |
| car | 0 | 0 | 1 | 0 | 0 |
| wedding | 0 | 0 | 0 | 1 | 0 |
| medical | 0 | 0 | 0 | 0 | 1 |

# One-Hot Encoding Applied

| Original Feature | Type | After One-Hot |
|---|---|---|
| home_ownership | Cat | 5 columns |
| purpose_cat | Cat | 26 columns |
| verification_status | Cat | 3 columns |
| policy_code | Cat | 5 columns |
| Numeric features | Num | unchanged |

# Handling Missing Data

Some records had **missing values** in multiple key features:

- `delinq_2yrs`, `total_acc`, `open_acc`, `pub_rec`, `inq_last_6mths`

Observation: **all missing values coincided in the same records**
Decision: **exclude these records** to clean the dataset
Purpose: Ensure **model has complete and reliable data**
**Drop rows** where all or most key features are missing
**Fill remaining missing values** with sensible defaults (e.g., 0 for counts)
The feature `emp_length` (years of employment) had **some missing values**
**Imputation strategy:** fill missing values with **1.0** (assuming minimum employment length)

# Excluding those 5 records

| | delinq_2yrs | total_acc | open_acc | pub_rec | inq_last_6mths | revol_util | a |
|---|---|---|---|---|---|---|---|
| **4319** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **4328** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **4678** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **6232** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **7592** | NaN | NaN | NaN | NaN | NaN | NaN | |

9995 Records left to consider

# Training  - Testing Split

Total Dataset
├── 70% Training Set → model training
└── 30% Test Set → model evaluation

# Feature Scaling & Handling Class Imbalance

**Scaling numbers:**

- Some features (like income or credit balances) have **big differences in size**
- We **standardize them** so all features are on the **same scale** → helps the model learn better

**Balancing the classes (SMOTE):**

- Most loans are “good” and fewer are “bad”  dataset is **imbalanced**
- SMOTE **creates artificial examples of bad loans** to balance the dataset
- Ensures the model **doesn’t just predict the majority class**

# Models

| Model | Type | Key Idea |
|---|---|---|
| Logistic Regression | Linear | Probabilities for class |
| Ridge Classifier | Linear | Linear + regularization |
| Random Forest | Tree Ensemble | Many trees vote together |
| Gradient Boosting | Tree Ensemble | Sequentially improves trees |
| AdaBoost | Ensemble | Focus on hard examples |
| SVM | Linear / Kernel | Maximize class separation |
| XGBoost | Boosting | Fast, accurate gradient boosting |
| LightGBM | Boosting | Efficient boosting for large data |

# Evaluation

**AUC (Area Under the Curve)**

- Measures level of separation.
- Think of it as ranking score:
  - 1.0 : perfect separation
  - 0.5 : random guessing

**F1 Score**

- How well it is identifying minority class
- Balances two things:
  - **Precision** : of all loans predicted as bad, how many really are bad
  - **Recall** : of all actual bad loans, how many did the model catch
- Higher F1 = model catches bad loans with avoiding false alarm.

# Model outputs

| Model | AUC | F1 |
|---|---|---|
| Logistic Regression | 0.698 | 0.267 |
| Ridge Classifier | 0.698 | 0.260 |
| Random Forest | 0.688 | 0.236 |
| Gradient Boosting | 0.700 | 0.297 |
| AdaBoost | 0.701 | 0.292 |
| SVM | 0.627 | 0.252 |
| XGBoost | 0.684 | 0.285 |
| LightGBM | 0.677 | 0.230 |

# Result

- **Gradient Boosting and AdaBoost** are the best performers for this data
- Some models like **SVM and LightGBM** struggled to detect bad loans
- F1 scores are generally **low** due to **imbalanced data**, even after SMOTE.
- AUC around 0.7 model **better than random guessing**, but not perfect

# Appendix - AdaBoost - Weak Learners

Iteration 1 → weak model 1 → some mistakes
Iteration 2 → weak model 2 → focuses on previous mistakes
Iteration 3 → weak model 3 → focuses on remaining mistakes
Final → combine all → strong prediction

# Appendix - Logistic regression

**Logistic Regression** is a **simple, widely used model** for classification
Predicts the **probability of an event happening** (e.g., loan being bad or good)
**Key idea:**

- Takes numeric input features (like income, credit history, debt ratio)
- Combines them **linearly**
- Uses a **sigmoid function** to convert the result into a **probability between 0 and 1**

Can **classify observations** based on a probability threshold (e.g., <0.5  bad loan)

# Appendix - LightGBM

Iteration 1 → tree predicts some loans correctly

Iteration 2 → tree focuses on mistakes

Iteration 3 → tree improves further

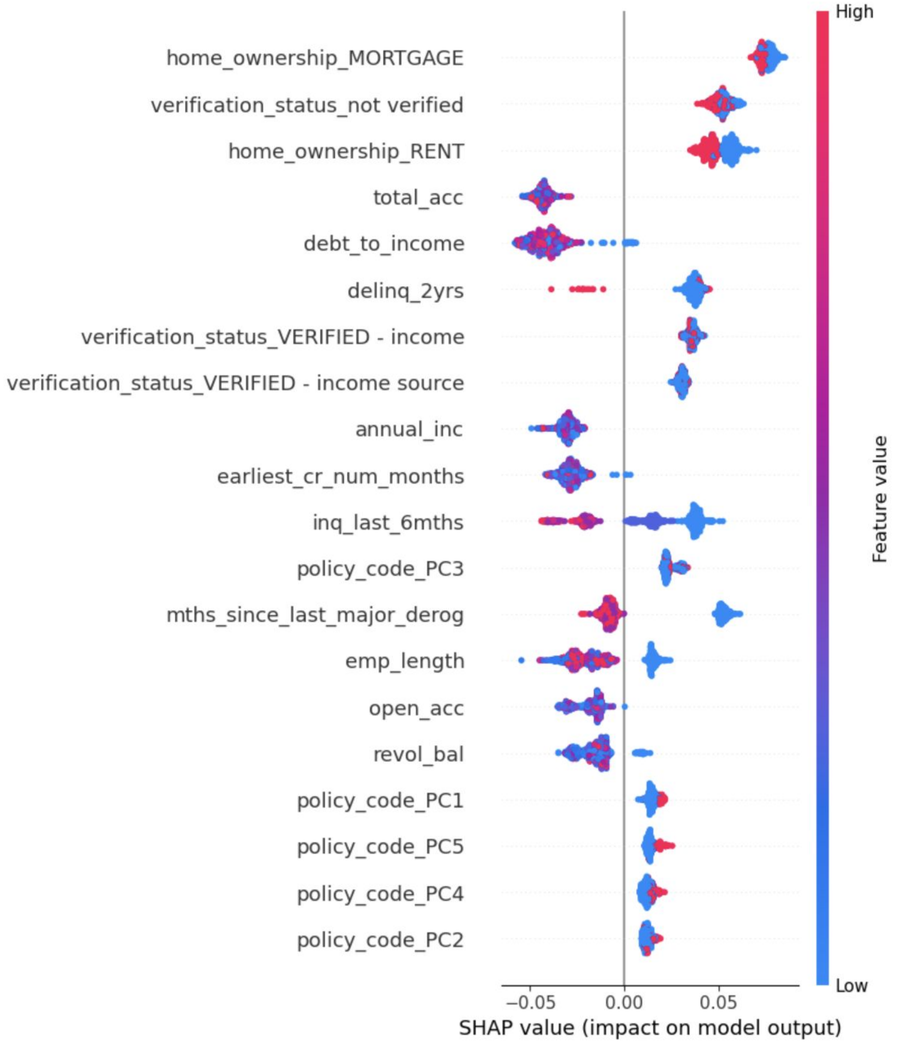
Final → combine trees → strong prediction

# Appendix - Random Forest

Tree 1 → predicts
Tree 2 → predicts
Tree 3 → predicts
...
Final Prediction → majority vote

# Shap Analysis

**SHAP** is a way to **explain how each feature in your model affects a prediction** — basically, it tells you *why* your model made a certain decision.



# Feature Importance

```
importances = pd.Series(clf_rf.feature_importances_, index=X_train.columns)
importances.sort_values(ascending=False).head(20)
```

```
inq_last_6mths                                      0.127160
mths_since_last_major_derog                         0.091324
emp_length                                          0.068218
revol_util                                          0.066540
annual_inc                                          0.059367
total_acc                                           0.058753
revol_bal                                           0.058376
open_acc                                            0.056218
earliest_cr_num_months                              0.055099
debt_to_income                                      0.054619
verification_status_not verified                    0.027056
purpose_cat_debt consolidation small business       0.024113
verification_status_VERIFIED - income               0.021528
home_ownership_RENT                                 0.021268
home_ownership_MORTGAGE                             0.020765
delinq_2yrs                                         0.020246
verification_status_VERIFIED - income source        0.015577
policy_code_PC2                                     0.014577
purpose_cat_debt consolidation                      0.014099
policy_code_PC3                                     0.013886
dtype: float64
```