LendingClub

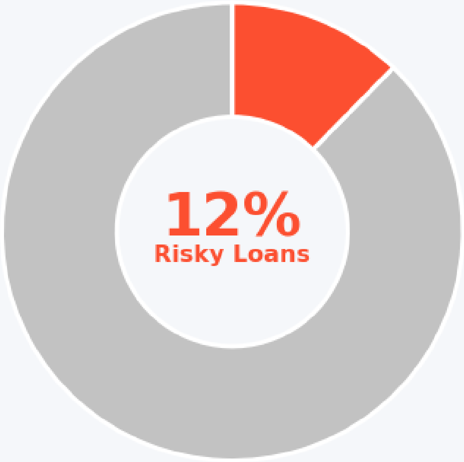
Loan Credit Risk Prediction

Author: Adhang Muntaha

Background

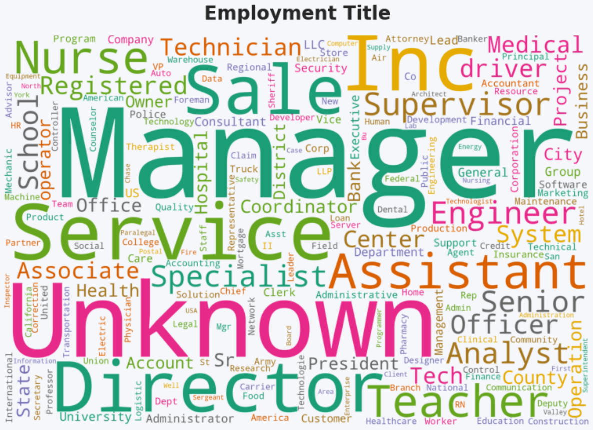
- A lending company has to **make a decision** whether they will accept or decline a loan application
- If the applicant is likely to pay off the loan but they don't approve their application, it may result in a **loss of income**
- If the applicant is not likely to pay off the loan but they approve their application, it may result in **financial loss**

What Happened?



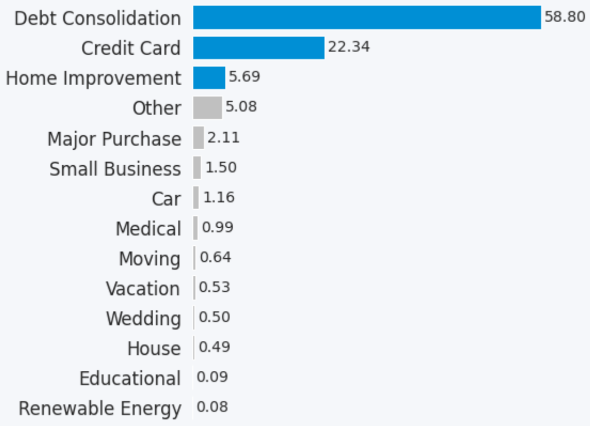
The ***Good*** status is when the loan status is either ***Current*** or ***Fully Paid***, otherwise it's ***Bad*** (risky credit)

Who Are The Borrowers?



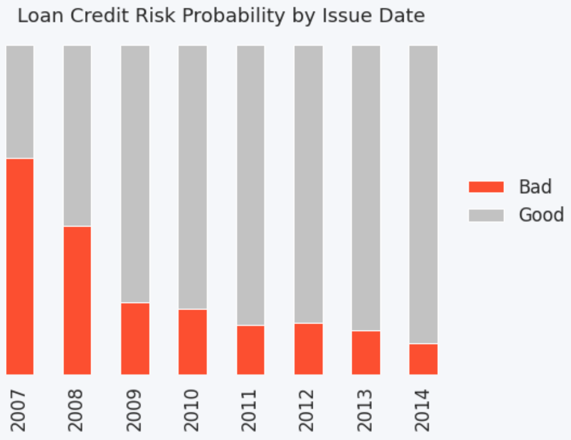
- Many borrowers have the words ***Manager***, ***Service***, ***Director***, ***Assistant***, ***Sale***, ***Teacher***, or ***Nurse*** in their employment title
- Many borrowers didn't write their employment title, so it's marked as ***Unknown***

Why Did They Apply For A Loan?



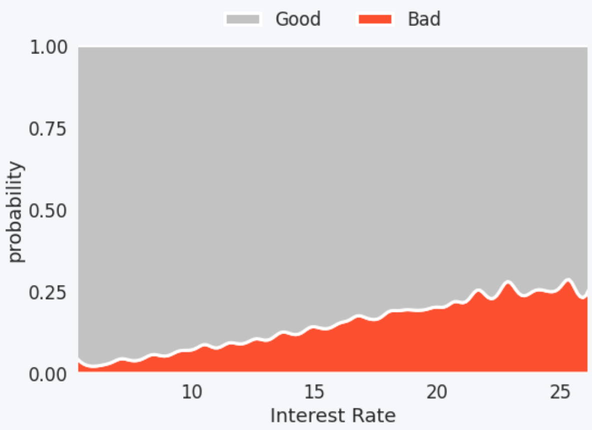
Most borrowers apply for loans for the purpose of **debt consolidation**

Are They New Borrowers?



The **earlier the issue date** is, the higher the probability of a borrower to have a **bad loan status**

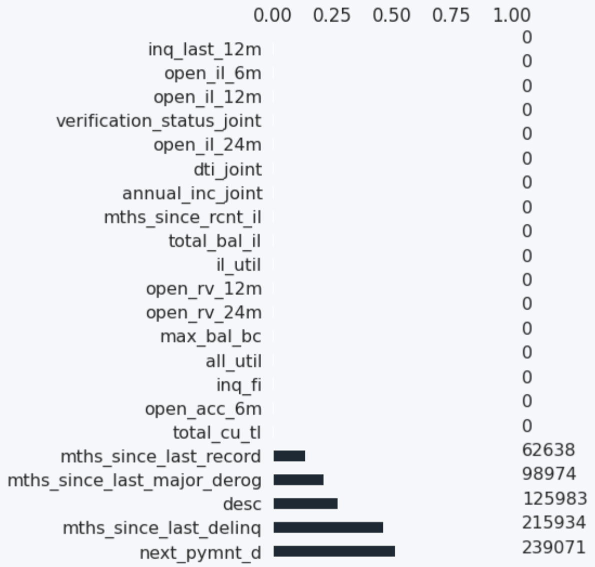
Do Interest Rates Matter?



Borrowers with **high-interest rates** have a higher probability to have a **bad loan status** than those with a low-interest rate

Missing Values

Some features have a lot of missing values. Moreover, there are some features that contain no data at all.

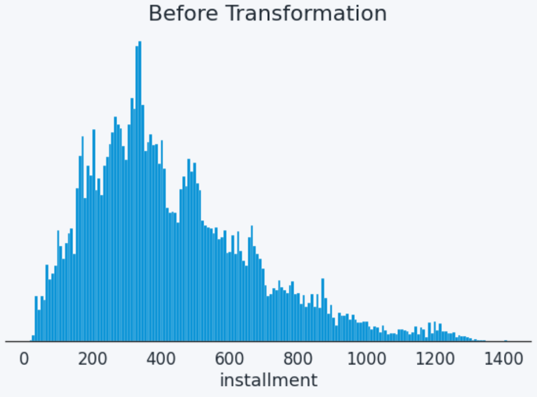


Solution:

- **Remove features** that have too many missing values
- **Fill in** the missing values using a univariate or multivariate imputation

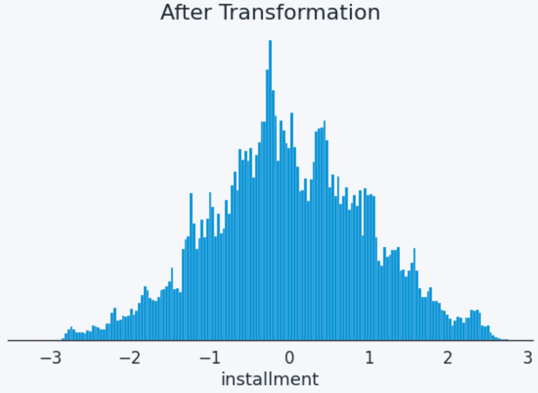
Feature Normalization

Before Transformation



Because some of the numerical features are skewed, I used the **yeo-johnson transform** to make the data more Gaussian-like

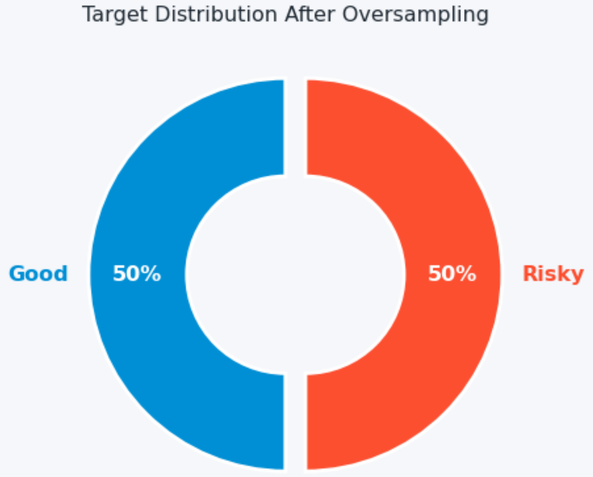
After Transformation



Oversampling With SMOTE

This dataset is imbalanced. I use SMOTE to make it balanced.

Target Distribution After Oversampling



Model Development

I use the **gradient boosted trees** model (XGBoost & LightGBM) for model development

Check out the detailed project on my GitHub **adhang.github.io**

Model Optimization

I use **Optuna** for hyperparameter tuning with tuning strategies:

- I want to avoid either high false negatives or high false positives, therefore I will use the **F1 score** for model evaluation
- I'm still paying attention to the **accuracy** score as well since this metric is easier to interpret

Model	Feature Selection	Accuracy	F1 Score
XGBoost	Using All Features	0.971	0.875
	Using 75% Features	0.971	0.876
	Using 50% Features	0.969	0.867
	Using 25% Features	0.955	0.826
LightGBM	Using All Features	0.975	0.891
	Using 75% Features	0.975	0.890
	Using 50% Features	0.972	0.877
	Using 25% Features	0.963	0.850

Conclusion

- **Selected model**: LightGBM using 75% features
- We should **pay more attention** to borrowers who meet the criteria below:
  - Earlier issue date
  - High interest rate
- Use **targeted ads** for potential borrowers based on their needs and occupations