WPI

Credit Score

CS539

Axel Luca, Preet Patel, Xiaojun Liu

Problem Motivation and Description

Credit scoring is a critical process in the financial industry, determining the creditworthiness of individuals applying for loans, credit cards, and other financial products. Accurate credit scoring models help financial institutions minimize risk, make informed lending decisions, and offer appropriate credit limits. By ensuring that credit decisions are based on reliable and comprehensive assessments, financial institutions can extend credit to deserving individuals and businesses who might otherwise be overlooked if the process was done by someone manually. However, building robust and accurate credit scoring models is challenging due to several factors: imbalanced data, complexity of features, and need for adaptability over time. We try and combat these factors with our model.

Dataset

Dataset

The Credit Score Classification dataset, available on **Kaggle**, contains information necessary to build and evaluate models for predicting credit scores.

Payment_of_Min_Amount: Minimum amount paid indicator.

Delay_from_due_date: Average number of days delayed in payment.

Credit_Mix: Credit mix indicator.

Outstanding_Debt: Total outstanding debt.

Changed_Credit_Limit: Credit limit change indicator.

Credit_History_Age: Age of credit history.

Payment_of_Min_Amount: Minimum amount paid indicator.

Dataset link: https://www.kaggle.com/datasets/parisrohan/credit-score-classification/data

WPI

Data Cleaning

When we first obtained the dataset from **Kaggle**, the data had several issues such as some data instances having values that weren't numbers, clear outliers in the data such as a negative age, and some data instances having strange symbols.

15	August	28	Teacher	34847.84	3037.986667	2	4	6	1	Credit-Builder Loan
16	January	34	_____	143162.64	12187.220000	1	5	8	3	Auto Loan, Auto Loan, and Not Specified
17	February	34	Engineer	143162.64	12187.220000	1	5	8	3	Auto Loan, Auto Loan, and Not Specified
18	March	34	_____	143162.64	NaN	1	5	8	3	Auto Loan, Auto Loan, and Not Specified
19	April	34	Engineer	143162.64	12187.220000	1	5	8	3	Auto Loan, Auto Loan, and Not Specified

No	18.816215	218.904344	Low_spent_Small_value_payments	356.078109	Good
No	246.992319	168.413703	!@9#%8	1043.315978	Good
No	246.992319	232.860384	High_spent_Small_value_payments	998.869297	Good
No	246.992319	10000.000000	High_spent_Small_value_payments	715.741367	Good
No	246.992319	825.216270	Low_spent_Medium_value_payments	426.513411	Good

Data Cleaning

To address these issues, several techniques were applied such as dropping irrelevant features such as the "Name," "SSN," "ID," and "Customer_ID, creating our own functions in Python to deal with outliers, removing the data instances that had strange symbols, and converting categorical variables that did not have a specific ranking or ordering into one-hot encodings, and converting the ones that did to numbers based on their rank or slot in their corresponding sequence.

15	8	28.0	34847.84	3037.986667	2	4	6	3
17	2	34.0	143162.64	12187.220000	1	5	8	13
21	6	34.0	143162.64	12187.220000	1	5	8	8
22	7	34.0	143162.64	12187.220000	1	5	8	8
23	8	34.0	143162.64	12187.220000	1	5	8	8

Occupation_Scientist	Occupation_Teacher	Occupation_Writer	Credit_Mix_Bad	Credit_Mix_Good	Credit_Mix_Standard	Payment_of_Min_Amount_No
1	0	0	0	1	0	1
1	0	0	0	1	0	1
0	1	0	0	1	0	1
0	1	0	0	1	0	1
0	1	0	0	1	0	1
0	1	0	0	1	0	1

WPI

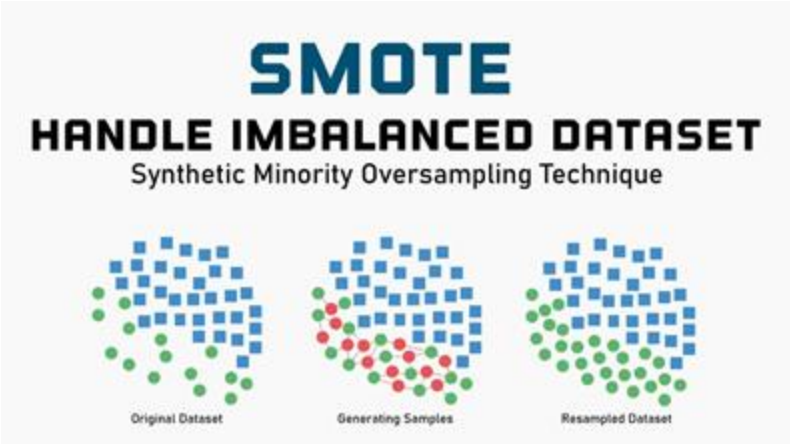
Synthetic Minority Over-Sampling Technique (SMOTE)

What is SMOTE?

- Addresses class imbalance in datasets
- Creates synthetic examples of the minority class

Advantages:

- Improves model performance
- Reduces overfitting
- Widely Applicable

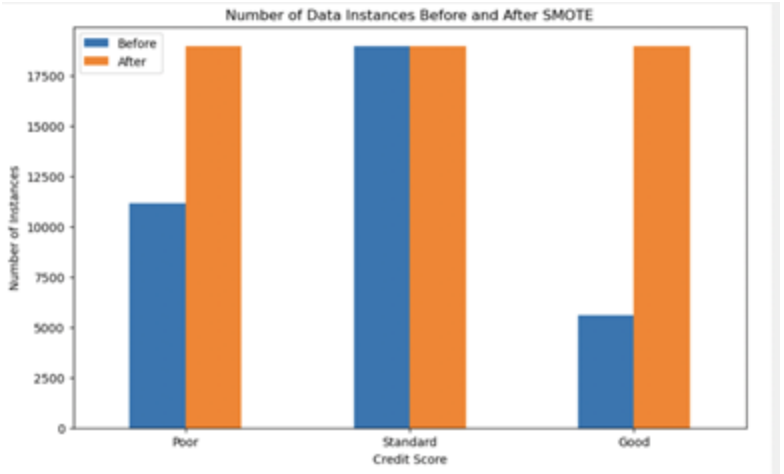


WPI

How it helped our project?

- Our dataset was imbalanced in the Credit_Score target variable:
  - Poor class: 11,139 instances
  - Standard class: 18,959 instances
  - Good class: 5,593 instances

- Applied SMOTE to make sure the target variable was equally represented in the dataset in order to train a fair and effective model



WPI

Elastic-Net

What is Elastic Net?

- A compromise between lasso and ridge, penalizing a mix of absolute and squared size
- These ratio of these two penalty types(l1_ratio) and the regularization strength(alpha) should be tuned
- Tuning of these hyperparameters depends on the dataset and problem

Fitting the Elastic Net Model

```
45    Payment_of_Min_Amount_Yes
Name: Feature, dtype: object
```

Alpha = 1, L1_ratio = 0.5

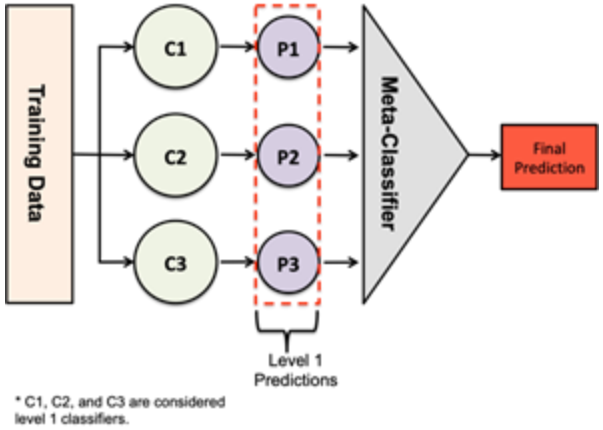
```
0                              Month
1                                Age
3                Monthly_Inhand_Salary
7                  Delay_from_due_date
9                Changed_Credit_Limit
11                   Outstanding_Debt
13                 Credit_History_Age
17                    Count_Auto Loan
19                 Count_Personal Loan
21                 Count_Not Specified
24     Count_Debt Consolidation Loan
25                   Count_Payday Loan
41                      Credit_Mix_Bad
42                     Credit_Mix_Good
43                 Credit_Mix_Standard
44           Payment_of_Min_Amount_No
45          Payment_of_Min_Amount_Yes
46    Spent_Amount_Payment_Behaviour
47    Value_Amount_Payment_Behaviour
Name: Feature, dtype: object
```

Alpha = 0.15,
L1_ratio = 0.1

WPI

Stacking Classifier

Stacking Classifier



As part of our project was to make a model that combines multiple machine learning classifiers to make predictions with the help of the classifiers' strengths, we decided to use a Stacking Classifier to accomplish the task. Here is information regarding a stacking classifier's typical components:

Base Classifiers: Several diverse base classifiers are trained on the same dataset. These can be different types of classifiers (e.g., decision trees, SVMs, neural networks) or the same type trained on different subsets of the data.

Meta-Classifier: A meta-classifier (or blender) is then trained on the predictions of the base classifiers. Instead of using the original features, the meta-classifier uses the outputs (predictions) of the base classifiers as its inputs. This meta-classifier then makes the final prediction.

WPI

K-Fold Cross-Validation

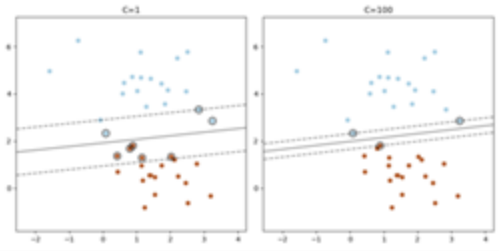
K-Fold Cross-Validation

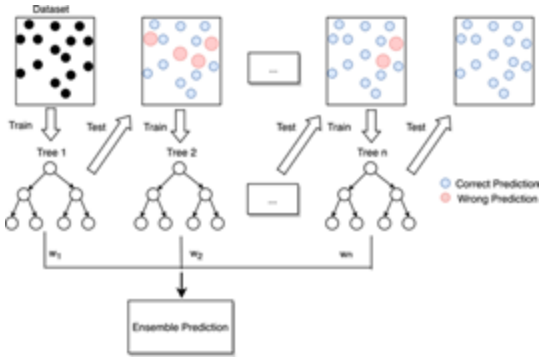
Advantage:

Provides a more accurate estimate of model performance compared to a simple train-test split. Utilizes the entire dataset for training and validation, which is beneficial especially when the dataset size is limited.

WPI

Building the Model



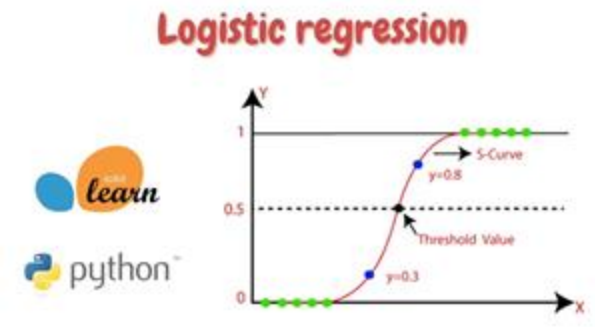


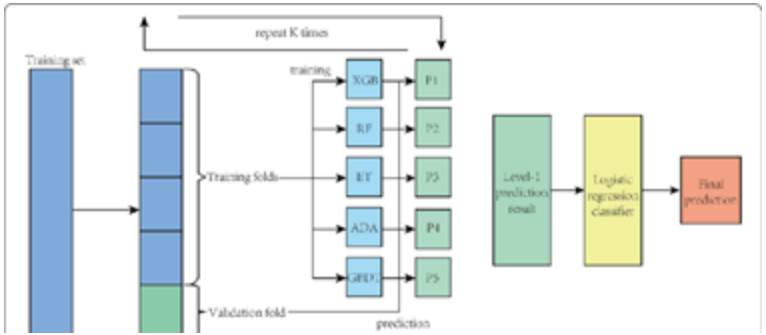
1) Defining Base Classifiers:

The base classifiers we selected were a linear SVM classifier as it is a good choice when dealing with datasets with a large amount of features and for modelling linear relationships between them. We then combined this classifier with an eXtreme Gradient Boosting Classifier, a powerful gradient boosting classifier used for large datasets and known for its performance and efficiency and its ability to model relationships between variables that are more complex than linear.

WPI

Building the Model





2) Defining the Final Classifier:

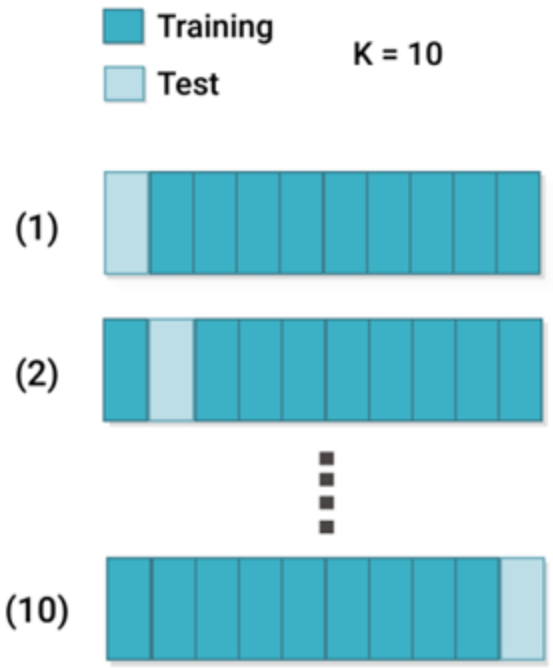
As the final estimator to combine the predictions from the base classifiers described above, we thought that the logistic regression was a good choice for the final classifier due to its simplicity, interpretability, and its effectiveness in predicting discrete variable outcomes.

3) Building the Stacking Classifier:

In our stacking classifier, cross-validation is used to split the our project dataset into multiple folds. For each fold, the models used as based estimators are trained the training portion of our dataset consisting of the remaining K-1 folds. Once that is done, the predictions of the base estimators on the current validation fold are used to train the stacking classifier's final estimator, which combines these predictions to improve the overall classification performance.

WPI

Training and Testing the Model



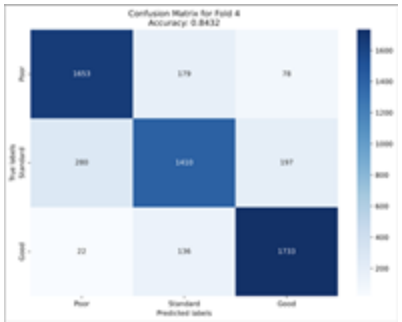
1) Setting Up 10-Fold Cross-Validation

In our project, we have set up K Fold cross-validation with 10 splits. For each fold, we had 90% of the data used for training the model and 10% of the data used for testing the model for both. In addition to this, shuffling was added to our model to ensure that each fold is representative of the entire dataset. Shuffling randomly distributes the data points, ensuring that each fold contains a diverse subset of the data.

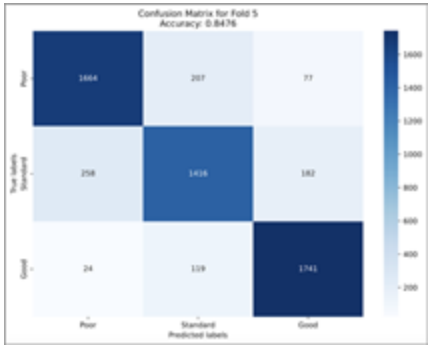
Results

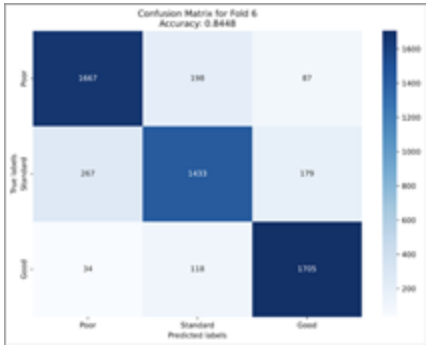
Results

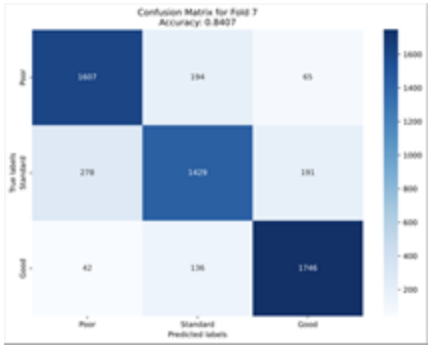
As the main goal of our project is to produce a machine learning model that can classify credit scores as accurately as possible, we decided to use **accuracy** as the machine learning metric to evaluate our finalized model. Our finalized model performed relatively well with a median accuracy of approximately 84.07% and a mean accuracy of approximately 84.11%.

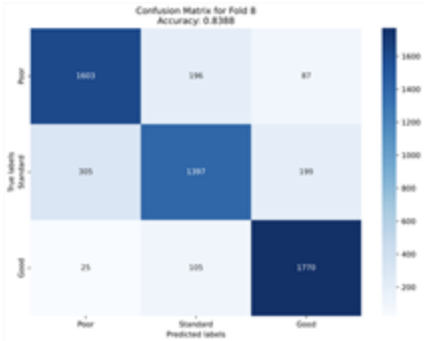


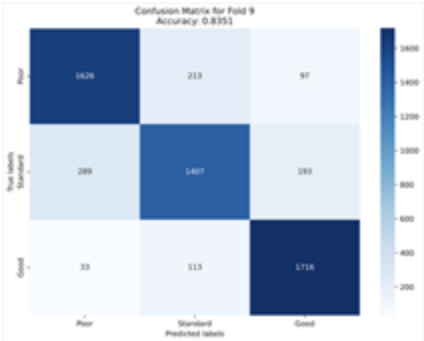
Results


Confusion Matrix for Fold 5
Accuracy: 0.8476
Poor
Standard
Good
True labels
Predicted labels
1664
207
77
258
1416
182
24
119
1741


Confusion Matrix for Fold 6
Accuracy: 0.8468
Poor
Standard
Good
True labels
Predicted labels
1667
198
87
267
1433
179
34
118
1705


Confusion Matrix for Fold 7
Accuracy: 0.8407
Poor
Standard
Good
True labels
Predicted labels
1607
194
65
278
1429
191
42
136
1746


Confusion Matrix for Fold 8
Accuracy: 0.8388
Poor
Standard
Good
True labels
Predicted labels
1603
196
87
305
1397
199
25
105
1770


Confusion Matrix for Fold 9
Accuracy: 0.8351
Poor
Standard
Good
True labels
Predicted labels
1626
213
97
289
1407
193
33
113
1716


Confusion Matrix for Fold 10
Accuracy: 0.8324
Poor
Standard
Good
True labels
Predicted labels
1583
211
70
316
1403
192
35
129
1748

WPI

Thank you!

Bibliography

Bibliography

1. Highradius. (n.d.). *Credit scoring models: Types and examples*. Highradius. Retrieved from https://www.highradius.com/resources/Blog/credit-scoring-models-types-and-examples/
2. Varga, G. (2023, July 11). *Understanding credit scoring for fintechs*. Oscilar. Retrieved from https://oscilar.com/blog/credit-scoring-guide
3. Fengff1292. (n.d.). *Credict score prediction by NGBoost*. Kaggle. Retrieved July 18, 2024, from https://www.kaggle.com/code/fengff1292/credict-score-prediction-by-ngboost
4. Satpathy, S. (2020). *SMOTE for imbalanced classification with Python*. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321-357.
6. Train in Data. (n.d.). *Overcoming class imbalance with SMOTE: How to tackle imbalanced datasets in machine learning*. Retrieved from https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote-how-to-tackle-imbalanced-datasets-in-machine-learning/
7. Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in knowledge discovery and data mining: 13th Pacific-Asia conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 proceedings 13* (pp. 475-482). Springer Berlin Heidelberg.
8. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301-320.

WPI

Bibliography

8. Sharmasaravanan. (2023, August 25). Understanding Stacking Classifiers: A Comprehensive Guide. Medium; Medium. https://sharmasaravanan.medium.com/understanding-stacking-classifiers-a-comprehensive-guide-195bfab58e48
9. Nalepa, J., & Kawulok, M. (2019). Selecting training sets for support vector machines: A review. *Artificial Intelligence Review, 52*(2), 857-900.
10. Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning, 54*, 255-273.
11. Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of machine learning algorithms with different K values in K-fold cross-validation. *International Journal of Information Technology and Computer Science, 13*(6), 61-71.
12. Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
13. Shruti Dhumne. “What Is Lasso Regression ? - Shruti Dhumne - Medium.” Medium, Medium, 4 Mar. 2023, medium.com/@shruti.dhumne/what-is-lasso-regression-bd44addc448c#:~:text=One%20of%20the%20main%20disadvantages. Accessed 21 July 2024.
14. Evidently AI. “Accuracy vs. Precision vs. Recall in Machine Learning: What’s the Difference?” Www.evidentlyai.com, www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Accuracy%20is%20a%20metric%20that.

WPI

Bibliography

15. Evidently AI. “Accuracy vs. Precision vs. Recall in Machine Learning: What’s the Difference?” Www.evidentlyai.com, www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Accuracy%20is%20a%20metric%20that.
16. “What Is Model Accuracy in Machine Learning.” Iguazio, www.iguazio.com/glossary/model-accuracy-in-ml/.
17. “Accuracy vs. Precision vs. Recall in Machine Learning: What’s the Difference?” Www.evidentlyai.com, www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Pros%3A. Accessed 21 July 2024.
18. Wijaya, Cornellius Yudha. “The Limitation of Accuracy Score.” Non-Brand Data, 17 Feb. 2023, cornellius.substack.com/p/the-limitation-of-accuracy-score.
19. Simic, M. (2022, February 25). Gradient Boosting Trees vs. Random Forests | Baeldung on Computer Science. Www.baeldung.com. https://www.baeldung.com/cs/gradient-boosting-trees-vs-random-forests

WPI