Credit Score Classification

Kyle Jow, Jimmy Nguyen, Kyle Pickle, Jacob Tuttle, Kris Wong

December 2023

Introduction & background

A common problem in the finance and banking industry is assessing the risk involved with lending money to clients. To better assess a borrower's creditworthiness, lenders formulate and assign "credit scores" to their clients, based on preconceived data like income, spending, and credit history. This score can then be used by a bank as an indicator of whether a client will default or otherwise miss a payment on a given loan. The rise of cloud-based computing in the 21st century has made it more advantageous and easier than ever for banks to share a standardized database of credit scores and financial information on their clients. As a result, credit scoring is being used more and more frequently to determine "worthiness" for almost anything – qualifying for mortgages, insurance, and even more nuanced decisions like cell phone plans and determining your employability.

With this increase in credit scoring has come an increased interest in financial literacy among clients as to how to understand and improve one's score. As such, credit standards like FICO and VantageScore have begun giving their clients a way to view an estimation of their credit score, often provided through the bank and credit card services that utilize them. These credit score estimates are predicted in a way that is simple for the client to understand, often broken down into categories like "poor," "standard," and "good" and presented alongside graphs displaying factors like age bracket and income to put them into perspective. This demand for straightforward and transparent credit scoring by governments and clients alike has interestingly compelled banks to consider less intricate or "black box" machine learning models, creating a delicate balance between accuracy and simplicity.

credit scoring has always been a common, concrete example of machine learning, as – in the world of economics – even the tiniest increase in accuracy can save billions of dollars in the long-run. As a result, researchers have utilized a variety of machine learning algorithms to better predict credit scores and risk. Nearly all client data is now digitized by banks, and with numerous complex variables that drive one's score, credit score prediction through machine learning is an ideal way to not only predict creditworthiness but constantly tune the equations and hyperparameters used to determine it in the midst of a fluctuating, real-world economy.

Literature review

Economists and researchers alike have utilized a variety of machine learning algorithms to better predict credit scores and investment risk. Some common implementations include decision trees, logistic regression, and Support Vector Machines (SVM).

With the history of credit scoring tracing its roots back to the 1950s, bank decisions were initially guided by the 5 C's approach – Character, Capital, Collateral, Capacity, and Condition – relying on subjective, physically-recorded information it its assessments. While

we could now consider this approach a decision-tree model, Dastile et al. describe how the financial industry's regulatory need for transparency in lending decisions has led to the prevalence of logistic regression models, known for their simplicity and interpretability. While sophisticated, higher-accuracy machine learning models have arisen over time, their opacity continues to raise concerns. In response, Dastile et al. go on to note several key machine learning techniques between 2010 and 2018 that have created a guided framework of credit scoring, combining both accuracy and transparency in lending decisions.

One modern approach to credit scoring comes from Dumitrescu et. al., who use a particular model called Penalised Logistic Tree Regression (PLTR) for credit score classification with an improved logistic regression model that has nonlinear decision tree effects. It is able to predict credit score more accurately than the benchmark logistic regression commonly used in the industry, and on a level that is comparable to more accurate random forest models. This is because it is able to capture recursive, multi-variable traits in the data unnoticed by a typical logistic regression. Dumitrescu et. al. use multiple datasets to test the robustness of the model and argue that PLTR preserves the needed transparency and interpretability that come with logistic regression.

Dataset description and exploratory data analysis

The dataset contains 27 features. The features we will be focusing on in order to classify a person's credit score, which is categorical, into either "good", "standard", or "poor" credit will be features that had the most correlation with credit score according to the heat map generated from this data and features used in real-life credit score assessing. Although the Credit_Utilization_Ratio has little correlation to credit score, this feature is used to assess credit in real life, so we kept it. (Figure 1).

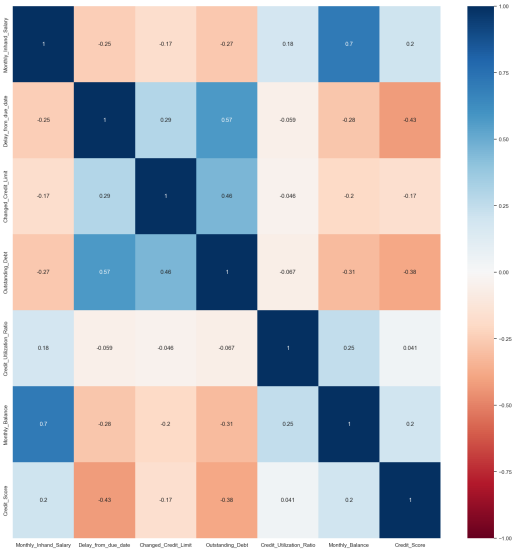


Figure 1: Heat map correlation.

Proposed methodology

From the literature review and the fact that this problem involves multiclassification, we decided to make and compare

two SVM models with a linear and RBF kernel and one multinomial logistic regression model. We trained the logistic regression model with and without outliters to determine the impact on the accuracy. For all three of our models, we used the standard 80:20 training to testing split in order to benchmark the performance on unseen data. Additionally, we used Streamlit to host our model online.

Experimental results

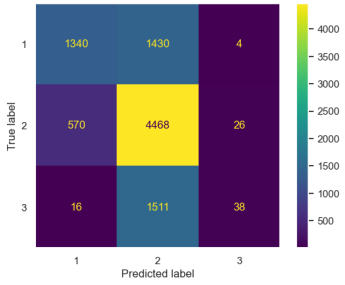
The Python library scikit-learn was used to create all four models. Both SVM models preserve default values aside from the change in kernel type. The two multinomial logistic regression models also preserve default values aside from setting the maximum iterations to two thousand, specifying the multi_class parameter to "multinomial" and using the limited-memory BFGS solver. For all models, the "Standard" credit score was most likely to be categorized correctly, followed by "Poor" and "Good". However, all models have relatively low accuracy overall, ranging from 0.59 to 0.62. The SVM with a RBF kernel performed the best with an accuracy of 0.62 (Figure 2).

In an attempt to improve the accuracy of the logistic regression model, which was performing the poorest, we removed outliers within the dataset. This had very little impact on the overall accuracy, but the f1-score for the "Good" credit score improved the most, going from 0.25 to 0.36 (Figure 5). Interestingly, the SVM model with a linear kernel was unable to classify a "Good" score at all, with zeroes for precision, recall, and f1-score (Figure 3). All models had a tendency to misclassify the "Standard" category the most as either "Poor" or "Good" at similar frequency.

Since our data was imbalanced, we tested if oversampling or undersampling can improve the accuracy of our SVM models. Overall, the SVM models performed slightly worse when oversampling or undersampling than not accounting for the dataset's imbalance. Additionally, there was no difference in accuracy for each SVM type when oversampling or undersampling, with an accuracy of 0.54 for the SVM with a linear kernel (Figures 6 and 8), and an accuracy of 0.56 for the SVM with a RBF kernel (Figures 7 and 9).

	precision	recall	f1-score	support
1	0.7	0.48	0.57	2774
2	0.6	0.88	0.72	5064
3	0.56	0.02	0.05	1565
accuracy			0.62	9403
macro avg	0.62	0.46	0.44	9403
weighted avg	0.62	0.62	0.56	9403

(a) Classification report for SVM with RBF kernel

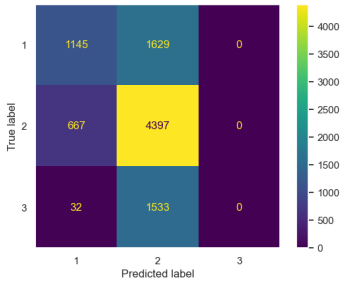


(b) SVM, RBF kernel

Figure 2

	precision	recall	f1-score	support
1	0.62	0.41	0.5	2774
2	0.58	0.87	0.7	5064
3	0	0	0	1565
accuracy			0.59	9403
macro avg	0.4	0.43	0.4	9403
weighted avg	0.5	0.59	0.52	9403

(a) Classification report for SVM with linear kernel

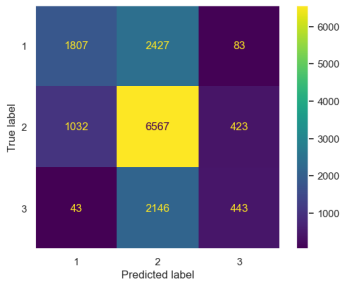


(b) SVM, linear kernel

Figure 3

	precision	recall	f1-score	support
1	0.63	0.42	0.5	4317
2	0.59	0.82	0.69	8022
3	0.47	0.17	0.25	2632
accuracy			0.59	14971
macro avg	0.56	0.47	0.48	14971
weighted avg	0.58	0.59	0.56	14971

(a) Classification report for logistic model with outliers

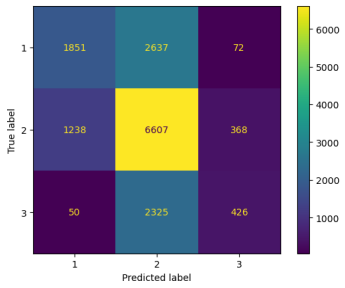


(b) Logistic with outliers

Figure 4

	precision	recall	f1-score	support
1	0.6	0.37	0.45	3483
2	0.61	0.81	0.69	7642
3	0.5	0.29	0.36	2719
accuracy			0.59	13844
macro avg	0.57	0.49	0.5	13844
weighted avg	0.58	0.59	0.57	13844

(a) Classification report for logistic model with no outliers

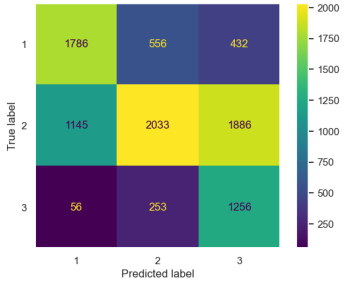


(b) Logistic with no outliers

Figure 5

	precision	recall	f1-score	support
1	0.6	0.64	0.62	2774
2	0.72	0.4	0.51	5064
3	0.35	0.8	0.49	1565
accuracy			0.54	9403
macro avg	0.55	0.62	0.54	9403
weighted avg	0.62	0.54	0.54	9403

(a)

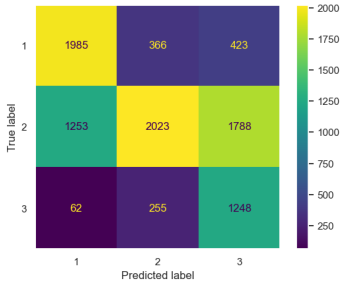


(b)

Figure 6: SVM linear w/ oversampling

	precision	recall	f1-score	support
1	0.6	0.72	0.65	2774
2	0.77	0.4	0.52	5064
3	0.36	0.8	0.5	1565
accuracy			0.56	9403
macro avg	0.58	0.64	0.56	9403
weighted avg	0.65	0.56	0.56	9403

(a)

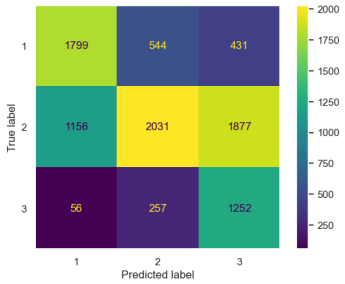


(b)

Figure 7: SVM RBF w/ oversampling

	precision	recall	f1-score	support
1	0.6	0.65	0.62	2774
2	0.72	0.4	0.51	5064
3	0.35	0.8	0.49	1565
accuracy			0.54	9403
macro avg	0.56	0.62	0.54	9403
weighted avg	0.62	0.54	0.54	9403

(a)

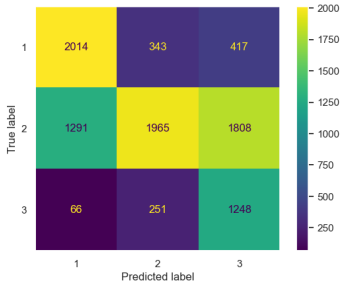


(b)

Figure 8: SVM linear w/ undersampling

	precision	recall	f1-score	support
1	0.6	0.73	0.66	2774
2	0.77	0.39	0.52	5064
3	0.36	0.8	0.5	1565
accuracy			0.56	9403
macro avg	0.57	0.64	0.56	9403
weighted avg	0.65	0.56	0.55	9403

(a)



(b)

Figure 9: SVM RBF w/ undersampling

Conclusion and discussion

Of the three different models we trained, SVM model using the RBF Kernel yielded the highest accuracy of 0.62. In our logistic model, removing the outliers caused a decrease in precison and recall when predicting poor credit scores, but an increase when predicting standard and good scores. Coincidentally, both logistic models ended up with an identical overall accuracy.

The SVM model using the RBF kernel, had a noticibly higher precision when predicting poor credit scores, meaning that a detected poor credit score is likely to be correct. This model also had a very high recall for standard credit scores, meaning that it was able to correctly identify almost all instances with a standard score.

In all of our models, it was very difficult to correctly identify a high credit score. This could have been caused by a much lower sample size when compared to entries of a poor or standard score. In particular, the SVM using the linear kernel was unable to report precision, recall, and f1-score for high credit scores, meaning that there was an insufficient amount of data correctly predict it. To fix some of these issues, we could modify the training data to contain an equal number of good, standard, and poor credit score samples.

References

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

Literature Review
Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. Applied Soft Computing, 91. doi:10.1016/j.asoc.2020.106263.

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. European Journal of Operational Research, 297(3), 1178-1192. doi:10.1016/j.ejor.2021.06.053