PREDICTING CREDIT CARD DEFAULT

A PREPRINT

Group Name: `Goup N`
Department of Biomechanical Engineering
University College London
London, WC1E 6BT

January 10, 2021

1 Introduction

In light of UK Finance expecting the value of credit card payments to rise from £638 billion in 2016 to £942 billion by 2026 [1], the problem of credit card delinquency has continued to gain in relevance, not only as a determinant of bank profitability, but also as an important component of broader financial system risk. Reducing damage and uncertainty by identifying counterparty default risk has been shown to be the single most important purpose served by credit risk models utilised by financial institutions [13, 9].

The machine learning use case of gauging credit default risk has been explored at length in literature. Research has primarily gravitated towards analyzing the performance of various statistical models on these data [28, 11, 20, 4]. However, other less-explored avenues of investigation exist involving the impact of feature engineering [17, 23, 3] and feature reduction techniques [14, 26, 27].

We ground our approach to this well-studied problem in what are broadly considered the objectives of machine learning applied to consumer finance: accuracy, interpretability, and efficiency [12]. Our approach falls under the less-explored avenues of feature reduction and behaviour. Specifically, we: (i) engineer the featuers to extract signal from client behaviour; (ii) conduct a tripartite feature reduction pipeline to construct three datasets that seek to maximise either interpretability or information; (iii) train a Support Vector Machine Classifier for each reduced dataset; and (iv) construct a final predictor which combines the three models in a voting procedure.

Training performance indicates . . . .

2 Data Transformation & Exploration

2.1 Data Cleaning

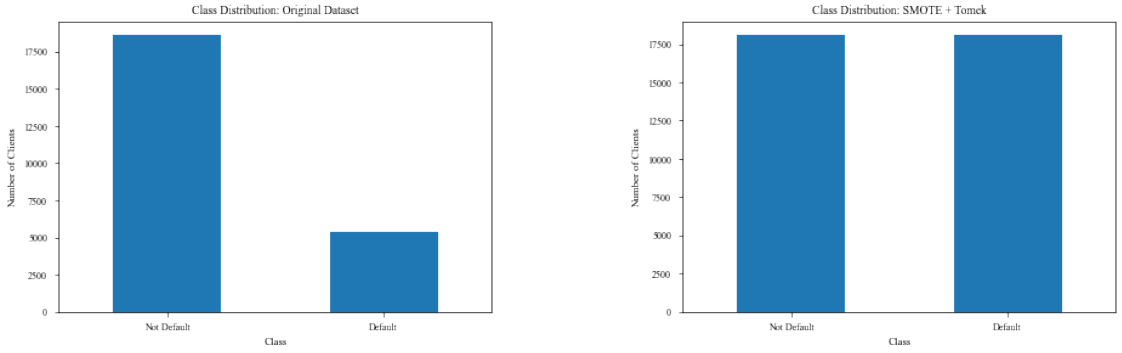
We begin the Data Transformation & Exploration stage with some simple data cleaning procedures to ensure the data is accurate, reliable and consistent. First, we find a limited number of instances where values with no pre-defined meaning are assigned to `MARRIAGE` and `EDUCATION`. For `MARRIAGE`, we group observations `0,3` together under `3` (Others) and for `EDUCATION`, we group observations `4,5,6,0` together under `4` (Others).

Secondly, we note that from the perspective of credit card default, small age deviations (e.g. 24 y.o. vs. 26 y.o.) are insignificant. Larger age deviations, however, are important to capture as they can implicitly provide information about income levels, job security and overall financial stability. Therefore, we aim to increase the signal-to-noise ration provided by the `AGE` feature by binning it, resulting in a new variable `AGE_BIN`.

Finally, we note that certain features with numerical values are, in fact, categorical. Therefore, we proceed by one-hot encoding `AGE, EDUCATION, SEX, MARRIAGE`. Note that we leave `PAY_X` unencoded because it is semi-continuous (i.e. the increase in the figure corresponds to an increase in months outstanding for payment).

2.2 Target Class Imbalance: Under- and Over-Sampling

A reasonable assumption for the dataset of credit card defaulters is to assume an imbalance between the defaulters vs. non-defaulters. In order to test this assumption, we plotted the number of clients for each class, as seen in Figure 1a. A clear imbalance between the two classes can be observed, with defaulters making up only 22.4% of the total training instances.



(a) Imbalanced Classes: Original Dataset

(b) Balanced Classes: Post-SMOTE-Tomek

Figure 1: Class Resampling

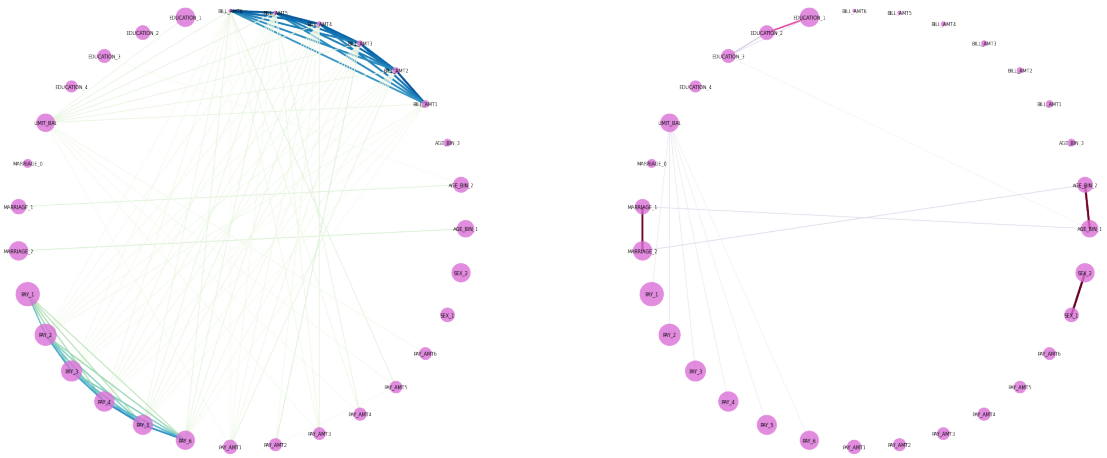
Given that the class of non-defaulters is represented by a much larger number of training examples compared to the other class, several inconveniences arise when training classification models on the dataset, as described in Ganganwar [10], Weiss [34]. Broadly speaking, the learning model will provide better accuracy for the majority class due to higher influence of the majority class over the training criteria. Thus, the fact that models are accuracy driven (their goal being to minimise the overall error rate, which is impacted significantly less by the minority class), often leads to the minority class having much lower precision and recall compared to the majority class. Additionally, degenerated models may be produced as the classifiers assume that errors coming from different classes have the same cost.

As a solution to problem, we opt for a data level method of handling imbalance [18]. We deviate from the oversampling approach to the same problem employed by Subasi and Cankurt [30], who use the Synthetic Minority Over-Sampling Technique (SMOTE) [7]. Driven by results from Batista et al. [5], who show that hybrid methods of oversampling and undersampling perform best on a dataset of similar size and class imbalance to ours, we choose the SMOTE + Tomek model. We start by oversampling with SMOTE and further proceed by identifying and removing Tomek links [32]. Thus, the class distribution is balanced and better-defined class clusters are created (fig. 1b). For practical purposes we use the implementation presented by Lemaître et al. [21].

2.3 Feature Correlation Analysis

To gain further insights into the composition of the dataset, we graphed the *independent* variables of the dataset as a network of nodes positioned along the perimeter of a circle with edges weighted and colored according to the correlation strength, and nodes sized according to each feature's correlation with the target variable. Figures 2a and 2b depict the resulting positive and negative correlation graphs, respectively.

Interesting insights can be gained from these visualizations. Both graphs clearly display areas of higher correlation densities. Adopting the nomenclature of Hu et al. [14], we refer to these groups of densely-correlated features as 'neighbourhoods' in the data. It is evident that the demographic neighbourhoods (`EDUCATION`, `SEX`, `MARRIAGE`, and `AGE`) are a simple statistical consequence of the prior one-hot encoding. On the other hand, because `PAY` and `BILL_AMT` represent the attribute of a client throughout 6 successive months, the formation of these neighbourhoods can be ascribed to the *sequential time dependency* of the feature groups. Put differently, consider that a client's repayment status and bill amount in May (`PAY_5` and `BILL_AMT5`) are dependent on the client's payment behaviour throughout April, and thus are closely related to the previous months' payment status and bill amount (`PAY_6` and `BILL_AMT6`). Hence, can relevant insights into clients' financial behaviour be extracted by analyzing these neighbourhoods in a time series fashion across the 6 months? This will be examined further in Section 3.1.



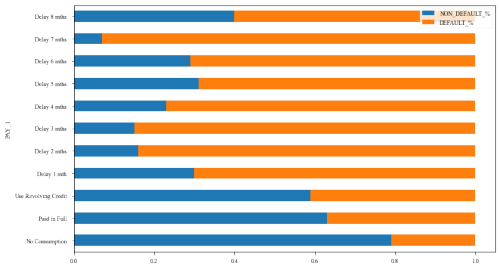
(a) Positive Correlations in Dataset

(b) Negative Correlations in Dataset

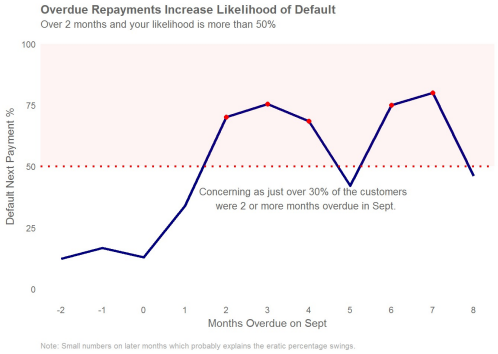
Figure 2: Feature Correlations

2.4 Target Class Distribution in Most Relevant Features

The prior correlation analysis highlights the importance of the PAY feature, in terms of both its relationship with other features and its sequential composition and ensuing neighbourhood. What information can be gleaned from it to extract further signals from the data? We divided the dataset into the two target classes, and for each month's PAY attribute we visualized the percentage distribution of defaults and non-defaults along each of its possible categorical values.



(a) PAY_1 Target Distribution



(b) Target Distribution Across all PAY Features

Figure 3: PAY Target Distributions

As figure 3 shows, there exist certain clusters of values for which default is less likely. As discussed in the prior correlation analysis, this feature is characterized by its sequential composition. Thus, clustering this behaviour over the 6 months may lead to valuable insights for a financial institution. This will be examined further in the feature engineering.

3 Methodology Overview

Credit risk modeling has been explored extensively in machine learning literature [25]. As is customary in this domain, the lion's share of research interest has been devoted to testing a wide variety of classifiers (KNNs, Logistic Regression, NNs, SVMs, etc.) to identify best performers [35]. Baesens et al. [4] find that neural networks and SVMs exhibit the best performance, although simpler methods (LDA, LR) also perform well. A single best performer elusive. On one hand, Huang et al. [15] conclude that the difference between SVMs and back-propagation NNs is insignificant. On the other, Li et al. [22] demonstrate SVM outperformance of NNs, but the size of their dataset disfavours the latter [6].

Other variants of the research have sought to drive performance through feature engineering to increase predictiveness. Agarwal et al. [3] idenfity that features describing client financial behaviour (e.g. FICO scores) exhibit clear correlations with risk of default. Kamil et al. [17] examine the impact of financial behaviour *a priori* by evaluating the relationship between clients' Financial Intelligence Quotient (FiQ) with credit card usage.

Besides accuracy, efficiency and interpretability are widely considered critical objectives in financial applications of machine learning [12]. Thus, feature reduction has also gathered important attention. Work by Hu et al. [14], Mbuvha et al. [26], Piramuthu [27] introduce novel feature selection methods that perform well in credit risk modelling. Because feature selection preserves variable meaning, it protects interpretability [24]. Dimensionality reduction approaches like PCA sacrifice interpretability, but have been found to outperform selection approaches [19].

Our approach to the present problem furthers work done in the feature engineering and reduction branches of the domain. First, we extract as much behavioural signal as possible from the data in novel ways by emphasising the time dimension of certain features in the dataset. Second, we reduce the dataset to maximise efficiency while protecting accuracy and interpretability in a tripartite fashion: selecting features using best-performing filter methods, selecting features using a simple heuristic, and reducing dimensionality through PCA. Third, we train a SVM Classifier on each of these reduced datasets in a Darwinist process. Fourth, we combine these three models in a final predictor that operates through a tripartite voting of these models.

3.1 Feature Engineering: Extracting Signal from Client Behaviour Over Time

3.1.1 Feature Creation

In line with our objective of encoding customer behaviour into the dataset, we engineered a series of features relating to the paying habits of the customers. In total, we added five features which have the general aim of capturing the repayment and billing situation of clients across the entire six month period included in our dataset.

The first 2 variables are looking at averages across the entire time-series. We computed a binary variable called "sufficiency" (`SUFF`), which is equal to `1` if the average bill amount is less than or equal the average payment amount (i.e. the payments are sufficient to cover the bill amounts over the recorded period) or equal to `0` otherwise. Additionally, we computed a simple average of the monthly change in repayment amount (`AVG_PAY_DELTA`), with the aim of recording the payment trend for a particular consumer. Here, a positive average MoM change in payment amount should be indicative of a solid financial situation for the client and we expect a negative correlation with the probability of default.

Then, we take inspiration from XX and create 3 "frequency" variables, motivated by the desire to record the number of occurrences of very favourable / unfavourable credit events during the six months included in the dataset. `PAY_DELAY_FREQ` stores the number of instances the customer was behind on payment for more than 3 months. `PAY_TIMELY_FREQ` is computed in similar fashion, but this time for the number of instances the client is on time with payment. Both features are created based on the `PAY_X` variables included in the original dataset. Finally, `REPAY_FREQ` is another "frequency" variables, this time computed based on the `PAY_AMTX` variables included in the original dataset. It stores the number of instances when the customer executed repayments (`PAY_AMTX` > 0). While we expect `PAY_TIMELY_FREQ` & `REPAY_FREQ` to be negatively correlated with the default probability and indicative of a solid customer profile, a high value for `PAY_DELAY_FREQ` is expected to be a predictor of default and a reflection of sub-optimal customer paying behaviour from the bank's perspective.

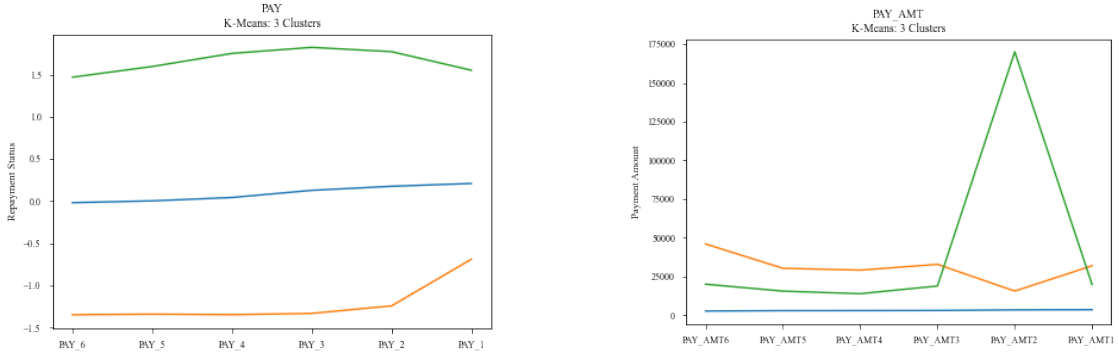
3.1.2 Clustering Latitudinal Client Behaviour by Time-Based Features

As noted in the exploration of the dataset's features in Sections 2.3 and 2.4, there exist motivations to further explore the characteristics of certain independent variables whose correlation densities formed neighbhourhoods in the data: `PAY` and `BILL_AMT`. Because our emphasis on characterising client behaviour implies a focus on their agency, we considered the examination of `PAY` and `BILL_AMT` to be dissimilar. While the former proxies for a client's repayment behaviour, the latter merely represents what the client *should* pay, not what they *did* pay. Thus, it made sense in the context of our analysis to instead consider `PAY_AMT`. Because each of these payment amounts are correlated with each of the `BILL_AMT` features, as exemplified in fig. 2a, this 'change of variable' is justified for our investigative purposes.

How should this behaviour be characterised? Following our analysis of the different 'clusters' of defaulters and non-defaulters apparent in the distribution of the `PAY` variable, we chose to try to identify trends in client repayment behaviour by finding clusters in these features across the 6 months spanned by the dataset. To do so, we considered client records for each feature along the six months as data points in a six-dimensional feature space. Note that due to

the characterization of these variables, proximity between these points can be measured as their distance in Euclidean terms[1] Hence, we approached this clustering task using the popular K-Means algorithm.

A K-Means model was trained on a range of 3 - 7 clusters for both variables. This reduced range was chosen deliberately with the aim of increasing the interpretability of results, which is important in the context of our use case. We identified the optimal number of clusters as $k = 3$ for both features using the Average Silhouette Method [29]. After training a K-Means model with 3 clusters on each feature group, we plotted the centroids of each cluster to reflect the groups of prototypical customer behaviours (see fig. 4).



(a) Clusters for `PAY`

(b) Clusters for `PAY_AMT`

Figure 4: Time-Based Feature Clusters

The graphs reveal interesting insights into the behaviour of the bank's clients. Repayment statuses display three clear variants. Clients that are generally behind in payments (higher risk of default); clients that use revolving credit on a recurrent basis (i.e. pay the minimum amount per month) (less risk of default); and clients that either do not consume or pay in full (least risk of default). Similarly, payment amounts reveal three characteristic schedules. Clients that pay near zero amounts across the 6 months (probably corresponding to those who do not consume); clients that pay regularly across the 6 months; and clients that pay large amounts towards the end of the period.

To enrich the data with these analyses, cluster allocations were encoded in the dataset in the form of two new features, `PAY_CLUSTER` and `PAY_AMT_CLUSTER`.

3.2 Feature Reduction: Feature Selection & Dimensionality Reduction

3.2.1 Feature Selection: Network Structure-based Feature Selection Algorithm (NSFSA)

The first feature selection methodology we adopted was motivated by the objective of selecting the most relevant variables in the data while preserving the dataset's underlying structure. To do so, we follow the approach devised by Hu et al. [14], dubbed Network Structure-based Feature Selection Algorithm (NSFSA). The goal of the algorithm is to select the best subset of features in the data that removes data redundancy while preserving an underlying network structure.

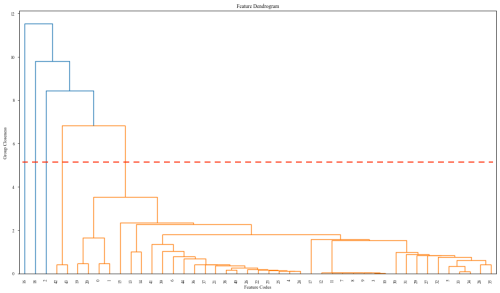
As the first step of NSFSA, we sought a principled approach towards organising the data's features into a coherent network structure. We present a novel methodology for this purpose. Given the different scales and units of the data's features and the varied categorical and numeric types, a typical distance-based clustering algorithm would not be suitable. An alternative course of action was motivated by the correlation graph presented in Section 2.3, where neighbourhoods of features are formed by areas of higher correlation densities. Thus, we attempted to use correlation as a proxy for the proximity between feature and defined a proximity metric, $p_{\mathbf{x}^{(i)},\mathbf{x}^{(j)}}$, as:

$$p_{\mathbf{x}^{(i)},\mathbf{x}^{(j)}} = \frac{1}{|\rho_{\mathbf{x}^{(i)},\mathbf{x}^{(j)}}| + \epsilon} - 1 \tag{1}$$

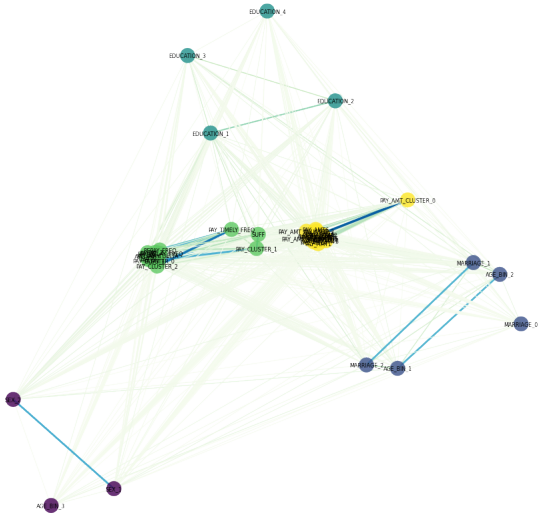
where $\epsilon \simeq 0$ to ensure validity for uncorrelated features, and 1 is subtracted such that the distance for perfectly correlated features is 0. Thus, highly-correlated features result in a smaller metric than uncorrelated ones.

[1] `PAY_AMT` is continuous, and thus suitable for this metric. While `PAY` is discrete, it also contains values along the reals, since the range represents increasing repayment delay. Thus, it is also suitable.

We adopt an Agglomerative Clustering model due to its greater flexibility and higher quality tree structure [33]. Despite these advantages, this model requires the preemptive specification of the desired number of clusters. To select this, we follow the Thorndike method, a popular heuristic approach in machine learning literature involvig dendrogram analysis to determine the optimal groups in a hierarchy [31]. This procedure reveals 5 clusters are optimal for our data (see fig. 5a). Because we seek to form neighbourhoods out of densely-correlated features, we trained an Agglomerative Clustering algorithm to form 5 clusters based on the single linkage method. Similarly, the single linkage method was desirable given it results in clusters in which elements in the cluster are more similar to at least one other neighbour than to external elements [8]. The resulting network structure can be visualized in fig. 5b.



(a) Feature Dendrogram: Thorndike Method



(b) Neighbourhood Structure

Figure 5: Network Structure Analysis

Second, for each feature, a proxy of its relevance in the neighourhood was computed as the sum of its correlations with all other features in that group, multiplied by the feature's correlation with the target variable.

The final step of NSFSA involves ranking the features in each neighbourhood according to the previous metric, and selecting the top n of each neighbourhood to form the best subse that retains the network structure. We chose the top two. Although a higher number of top features would increase the variance of the data, a smaller number of features is desirable for our analysis because it increases interpretability.

3.2.2 Feature Selection: Network-Based Heuristic

Driven by the overarching aim of reducing dimensionality and maximising interpretability, we motivated a simpler 'heuristic' feature selection method as seeking to not only summarise the set of features in the dataset but also the dataset's structure itself, thereby narrowing the focus into a single neighbourhood of features. From a use case standpoint, this approach is compelling: if a financial institution can zero in on a set of features with similar characteristics that can best predict credit risk, it may streamline its processes and enhance interpretability.

In this case, we segregate feature correlations with the target variable as the sole metric of interest. This is because a higher correlation of the overall feature subset with the target variable should lead to greater signal. To do so, we employed the network structure devised in the previous section. Then, for each neighbourhood, we omputed the sum of the features' correlation with the target variable. The neighbourhood with the highest computed metric was selected as the best subset of features.

3.2.3 Dimensionality Reduction: PCA

The motivation behind our third and final feature reduction methodology is to capture the information provided by the entire original feature set through a computationally smaller subset. More specifically, we are aiming to extract the most important information from our enhanced feature space and compress its size by only going forward with this important information. For this purpose we choose one of the oldest and most widely used multivariate statistical techniques, Principal Component Analysis (PCA) [2].

The procedure for computing the principal components in the basic version of PCA (used here) is fairly straightforward [16]. After the data is standardised, the sample covariance matrix is computed for the original dataset. Then, the eigenvectors and associated eigenvalues are computed for the sample covariance matrix. These eigenvectors are sorted in decreasing order of their corresponding eigenvalues. The principal components are the linear combinations of the original matrix and the eigenvectors (obtained as described below).

The results of our analysis can be observed in Figure 6. In order to select the number of components to be kept after the PCA step, we plot the cumulative proportion of variance explained for all principal components (scree plot). After setting our target at 95% of explained variance, the resulting feature subset contains 25 components.

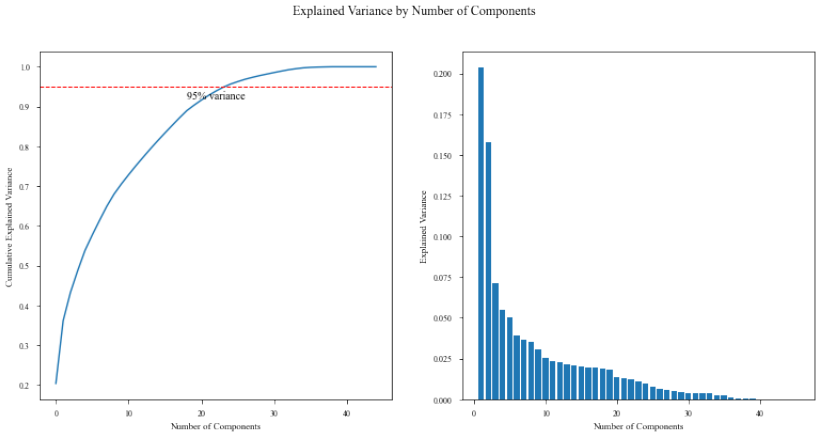


Figure 6: PCA Results

3.3 Model Selection, Training, & Validation

4 Model Training & Validation

5 Results

6 Final Predictions on Test Set

7 Conclusion

References

[1] URL `https://www.ukfinance.org.uk/`.

[2] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[3] S. Agarwal, S. Chomsisengphet, and C. Liu. The importance of adverse selection in the credit card market: Evidence from randomized trials of credit card solicitations. *Journal of Money, Credit and Banking*, 42(4): 743–754, 2010.

[4] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society*, 54(6):627–635, 2003.

[5] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[6] T. Bellotti and J. Crook. Support vector machines for credit scoring and discovery of significant features. *Expert systems with applications*, 36(2):3302–3308, 2009.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[8] A. El-Hamdouchi and P. Willett. Comparison of hierarchic agglomerative clustering methods for document retrieval. *The Computer Journal*, 32(3):220–227, 1989.

[9] A. Fatemi and I. Fooladi. Credit risk management: a survey of practices. *Managerial Finance*, 2006.

[10] V. Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[11] P. Giudici. Bayesian data mining, with application to benchmarking and credit scoring. *Applied Stochastic Models in Business and Industry*, 17(1):69–81, 2001.

[12] D. J. Hand and W. E. Henley. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541, 1997.

[13] T.-C. Hsu, S.-T. Liou, Y.-P. Wang, Y.-S. Huang, et al. Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1572–1576. IEEE, 2019.

[14] Y. Hu, Y. Ren, and Q. Wang. A feature selection based on network structure for credit card default prediction. In *CCF Conference on Computer Supported Cooperative Work and Social Computing*, pages 275–286. Springer, 2019.

[15] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision support systems*, 37(4):543–558, 2004.

[16] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[17] N. S. S. N. Kamil, R. Musa, and S. Z. Sahak. Examining the role of financial intelligence quotient (fiq) in explaining credit card usage behavior: A conceptual framework. *Procedia-Social and Behavioral Sciences*, 130: 568–576, 2014.

[18] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[19] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei. A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. *Journal of Retailing and Consumer Services*, 27:11–23, 2015.

[20] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen. Credit scoring using the hybrid neural discriminant technique. *Expert Systems with applications*, 23(3):245–254, 2002.

[21] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

[22] S.-T. Li, W. Shiue, and M.-H. Huang. The evaluation of consumer loans using support vector machines. *Expert Systems with Applications*, 30(4):772–782, 2006.

[23] H. Lu, H. Wang, and S. W. Yoon. Real time credit card default classification using adaptive boosting-based online learning algorithm. In *IIE Annual Conference. Proceedings*, pages 422–427. Institute of Industrial and Systems Engineers (IISE), 2017.

[24] M. Masaeli, G. Fung, and J. G. Dy. From transformation-based dimensionality reduction to feature selection. In *ICML*, 2010.

[25] E. Mays. *Credit risk modeling: Design and application*. Global Professional Publishi, 1998.

[26] R. Mbuvha, I. Boulkaibet, and T. Marwala. Bayesian automatic relevance determination for feature selection in credit default modelling. In *International Conference on Artificial Neural Networks*, pages 420–425. Springer, 2019.

[27] S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European journal of operational research*, 156(2):483–494, 2004.

[28] E. Rosenberg and A. Gleit. Quantitative methods in credit management: a survey. *Operations research*, 42(4): 589–613, 1994.

[29] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[30] A. Subasi and S. Cankurt. Prediction of default payment of credit card clients using data mining techniques. In *2019 International Engineering Conference (IEC)*, pages 115–120. IEEE, 2019.

[31] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[32] I. TOMEK. Two modifications of cnn. *IEEE Trans. Systems, Man and Cybernetics*, 6:769–772, 1976.

[33] B. Walter, K. Bala, M. Kulkarni, and K. Pingali. Fast agglomerative clustering for rendering. In *2008 IEEE Symposium on Interactive Ray Tracing*, pages 81–86. IEEE, 2008.

[34] G. M. Weiss. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19, 2004.

[35] I.-C. Yeh and C.-h. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.