Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет економічних наук

Кафедра фінансів

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему «**ОЦІНЮВАННЯ КРЕДИТНОГО РИЗИКУ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ**»

CREDIT RISK ASSESSMENT USING MACHINE LEARNING METHODS

Спеціальності:

072 Фінанси, банківська справа та страхування

Галузинський Артем Сергійович

Керівник: Долінський Л. Б.

доктор економічних наук, доцент

Рецензент: В.В. Вітлінський

Кваліфікаційна робота захищена
з оцінкою «_____»

Секретар ЕК _____
«_____» _____ 2024р.

Київ 2024

Contents

INTRODUCTION

Machine learning is a field, originating back from the middle twenties and has already rich history in its applicability and among researchers regardless of the sphere where it was used. However, considering recent significant growth of popularity of artificial intelligence, it creates an even more importance to research these algorithms properly when it comes to financial backgrounds. Originally, machine learning algorithms were more related as objects of mathematical models, providing solutions for much more complex and non-linear tasks opposing to regression and statistical models, limitations of rely heavily on the assumption conditions. While non-linearity usage does indeed have a trade-off in interpretability and complications in specifying optimal models, the potential of its usage remains supreme.

Among all the possible phenomenon's to be modelled in finance, there are several fields that exhibit properties that make machine learning relevant for consideration. Specifically, this paper will hold its attention on the aspect of credit risk modelling with a general overview of what this feature represents, what are its drivers and how they may be approached from the perspective of machine learning models. The connection arises from the popularity of such algorithms in probability estimation and solving classification problems, which fit efficiently modelling specific credit risk properties.

The purpose and tasks of the research: the core aim of this qualification paper is to overview already existing solutions for specific problems of credit risk and unify them in one specific algorithm solving a practical task in credit risk management.

Object and subject of research: the object of the research is credit risk modelling from perspective of machine learning with the corresponding subject of estimating specific effects realized by the forecasted volumes of credit risk through default probabilities with usage of quantitative approaches.

Methodology of the research: the main methods are based on machine learning and statistical algorithms, which hence provide the results of mathematical modelling in respective section of the paper.

Informational basis of the research: as for informational source were taken international requirements and recommendations for conducting credit risk assessment, specifically Basel and IFRS 9 and open-source databases of credit risk information. Approaches overviewed based both classical articles aging back to the discoveries of the algorithms Frank Rosenblatt, Leo Breiman and Adele Cutler, as well all the cumulated research that were gathered and organized thanks to Jomark Pablo Noriega, Luis Antonio Rivera and José Alfredo Herrera. Also, some approaches recommended by consulting companies, such as S&P500 were analyzed.

The practical meaning of such research is to broaden perspective over the credit risk solutions which are capable to be addressed to machine learning, what are the specifications of usage, possible limitations opposed to the capabilities and strengths for each specific situation relevant to credit risk.

The research itself is structured in the next way: introduction, section of establishing meaning of how credit risk is defined, what are its drivers and what threats it introduces, how are these estimated using machine learning algorithms, addressing practical problems of credit risk management using more comprehensive algorithms and opposing them to commonly used one with providing overall conclusions.

SECTION 1

PRINCIPLES AND MOTIVATION OF ML IN CREDIT RISK MODELLING

1.1 Derivation of core credit risk concepts

Credit risk – likelihood associated with loss occurrence or undergoing profits due to inability of client to continue credit repayment. [2]

This phenomenon is associated with various types of financial instruments, where such are determined as contracts that simultaneously leads to the emergence (increase) of a financial asset in one enterprise and financial liability or equity instrument in another.

This contract is one which provides a documentary claim on the existence and ownership of financial assets or financial liability, both of which are to be determined below.

1.**Financial assets**:[2]

- cash and cash equivalents.

- a contract giving the right to receive cash or another financial asset from another enterprise.

- a contract giving the right to exchange financial instruments with another enterprise on potentially favorable terms.

- equity instrument of another enterprise.

2. **Financial liability**: [2]

- to transfer money or another financial asset to another enterprise.

- to exchange financial instruments with another enterprise on potentially unfavorable terms.

Note should be added that not all these positions are subject to credit risk. Hence a more detailed derivation is proposed with how exactly credit risk affects the quality of the security. [2]

1. **Equity securities** - securities certifying participation of their owner in the authorized capital, giving the owner the right to participate in the management of the issuer and receiving part of the profit (in a form of dividends or part of the property in case of liquidation of the enterprise). Equity securities include:

- stocks
- investment certificate
- certificate of funds of real estate operations

Price of such fluctuates depending on the market expectation over financial health of the issuer, where credit risk increase is likely to draw the price down as the risk premium of holding such security correlates with the risk of issuer default. Consequently, when issuer defaults, equity holder is likely to lose all the capital invested in the security. [2]

2. **Debt securities** - securities certifying a loan relationship between two counterparties and providing for the issuer's obligation to pay in the fixed term of funds in accordance with the obligation. Debt securities include:

- corporate bonds
- government bonds
- bonds of local loans
- deposit certificates
- promissory notes

Such are considered most sensitive to credit risk, as these securities form direct credit relationships between the borrower and the lender, and the credit risk is present as defined. [2]

3. **Derivative securities** - these are securities, the mechanism of issue and circulation of which is associated with the right to purchase or sell securities, other financial and/or commodity resources within the term determined by the agreement (contract).

Credit risk in derivative securities arises when one party fails to fulfill their contractual obligations, leading to potential losses for the counterparty engaged in the derivative contract with respective base asset. [2]

4. **Mortgage securities** - securities, the issuing of which is secured by mortgage coverage (mortgage pool) and which prove the right of the owners to receive their funds from the issuer. Mortgage securities include:

- mortgage bonds
- mortgage certificates
- collateral mortgage

Similarly to the debt securities, while having the mortgage as a collateral analogy, making the credit risk arise respectively. [2]

5. **Commodity securities** - securities that give their holder the right to dispose of the property specified in these documents. [2]

Credit risk is additionally driven by any other risk factor yet remarks a much more severe nature and usually is found by a consequence of deterioration of financial asset quality. Such key risk-factors that can be used while performing credit risk assessment are presented below: [22]

- **Instrument Maturity**

Time concept is a necessary factor to adjust financial value to since a longer time has a higher potential of accumulating more possible risks of any kind. This indicates longer exposure period, hence higher cumulative probability of the default value, that most notably requires higher risk appetite for being exposed to risk of such specific amount.

- **Interest Rate/Estimated Return Rate**

May be considered as a controversial risk-factor, since its nature assumes additional risk premium for the increase default likelihood, however the higher interest pressure by itself may increase the likelihood of the default. Hence this double-sided nature of this risk factor encourages on both sides of the financial contract to converge

to the maximization of expected returns given the obliged side agrees to comply with the contract.

- **Credit History and Current Debt/Liquidity State**

Additional debt obligation may be considered as possible only when it does not breach any critical statements while also remains a healthy credit history to assure the expected experience of the side being exposed to the credit risk.

- **Microeconomic/Macroeconomic Risk Factors**

Given the exposure is granted by the legal entity, its solvency may be driven by any drifts in its business activity, where latter holds strong impacts both by internal business process risks and economic environment influence, respective study of sensitivity to these changes, given additional consideration of ones that are most expected given current state of the economy holds crucial conclusions while assessing credit risk both on business, industry, country ad international levels of the assessment. Internal risks may be associated with analyzing counterparties engaged in the process of developing product, presence of systematic audit, and overall efficiency of the management decisions. In addition, financial risks are also included in the list of credit risk impact, associated with drifts of interest rate levels, currency shocks, depreciation or appreciation of respective products price worsening the costs or sales of the business, worsening of labor market etc.

- **Collateral Value**

Secured loans should account for the factors affecting market value of the collateral, such may be market demand and price fluctuations, time-value depreciation,

By establishing the approach to determine the quality of the asset with usage of such terms as financial value and underlying risk further explanation should be provided to classify these terms.

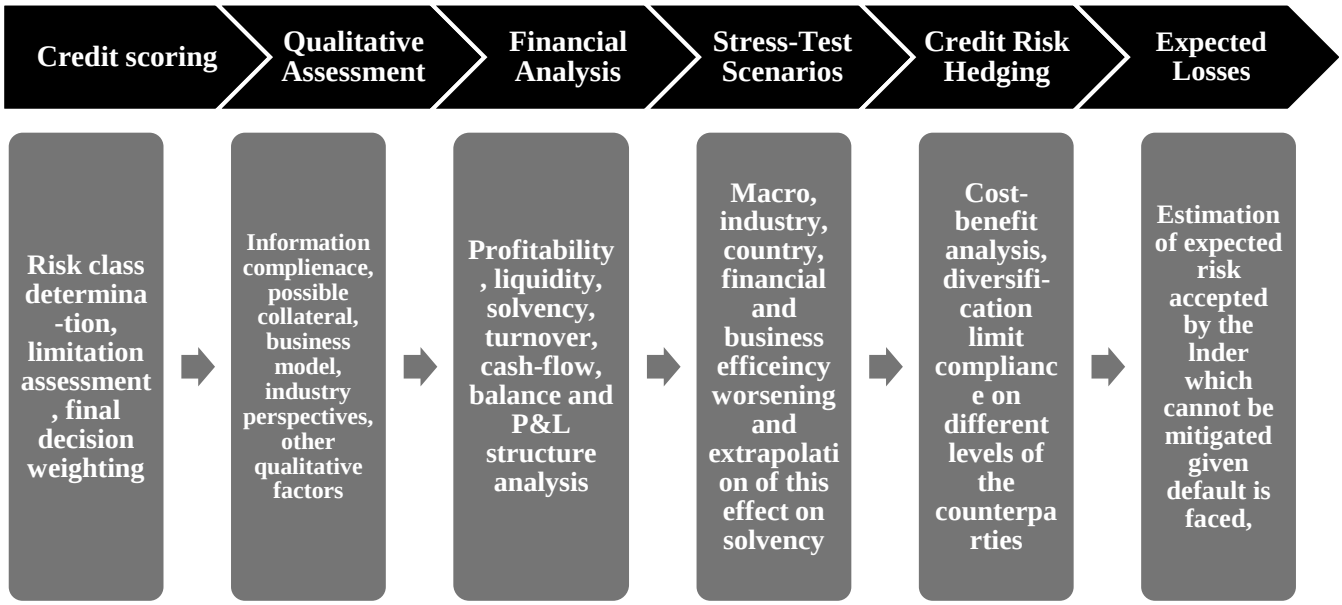
1.2 Machine learning frameworks in credit risk assessment

Machine learning comes as a robust primary of secondary estimator of risk class quantitative assessment and then further adjustments on the initial limitations of

financial contract and final decisions over lending a potential financial asset or making an investment. Additionally, since expected losses are associated with such parameters as loss given default and probability of default. this parameter estimation has been more and more by machine learning techniques on contrast to statistical models.

Considering recent breakthrough on language and visual model's, they may become a popular tool while performing a qualitative assessment to make credit risk estimation much more robust. Concerning similar as previous statement on AI expanding beyond purely mathematical concepts, this may serve a great tool to capture patterns given language model is trained specifically on evaluation of credit risk and financial assessment.

Stress-test scenarios prove their most robustness using statistical tools and historical simulations alongside with possible statistical model incorporation based on severe stress assumptions. At the same time, credit risk hedging remains currently mainly qualitative assessment subject, requiring significant effort and expertise on derivative.



Fig, 1.1. A general example of credit risk assessment components

Not only do these problems distinguish appropriate machine learning algorithm while assessing credit risk, yet additionally require different collected data. Relevant examples when either full or partial specified credit risk assessment is obligatory is presented below:

1. *Credit Card Approval*

   As mentioned earlier, assessing creditworthiness of an applicant while approving card among private individuals is a crucial for any banking entity that considers credit cards as significant part of their business model. It mostly incorporates models analyzing personal data and credit history data making a correct class of declining, acceptance, and further analysis to be conducted. [3]

2. *Mortgage Lending*

   On the contrast to previous issue, when assessing credit risk in mortgage lending lenders require similar information to make informed decisions on applicants' creditworthiness and likelihood of mortgage default as this is still private individual oriented product, these are more associated with provisioning financial resources on case of default, hence require probabilistic outputs.

3. *Acquiring of investment instruments:*

   Consideration of a purchase regarding various investment instruments such as bonds, stocks, or other securities, reliance on machine learning algorithms to analyze extensive sets of data is crucial. Assessment of the creditworthiness of the issuing entities and calculate the probability of default based on broader market trends, economic indicators, and the financial health of the issuer. ML models contribute by providing probabilistic outputs for such that guide investors in making well-informed decisions as well as to suffice provisioning levels in investment banking. [7]

4. *Vehicle Leasing*

   While being similar to previous two, they can be extended for both necessities of private individuals and legal entities, where algorithms can then differ as well as the collected data. Additionally, both provisioning and scoring decisions were applied in this field when making credit decisions as a credit risk indicator.

5. *Corporate Loans*

   Shifting to plane legal entities, in such area estimating such parameters as probability of default and respective losses given default remain target of the algorithm, yet the data related to this field is derived mainly from financial

analysis: financial ratios serve as main accepted approach to determine set of significant factors affecting credit quality given it is granted to respective legal entity. Machine learning models can analyze business financial statements, cash flow, credit history, industry trends, and other relevant features to predict the probability of loan default and hence find the amount of expected loss per exposure. [7]

6. *Peer-to-Peer Lending*

   This type of lending can be popular within the stock exchange platforms or recently more and more popular cryptocurrency exchange platforms, where such pose themselves as objects granting high quality of peer-to-peer interaction (also referred as p2p), in other words exchange platform is an intermediary, whereas having a high quality of service comes in line with distinguishing fraud motivated individuals. Such can be assessed using classification algorithms on the point of registration as well as consequent algorithmic monitoring, where machine learning models can leverage borrower and lender information. [4]

7. *Sovereign and banking default risk*

   As well as any corporate business, financial institutions both on government and financial sector levels require credit risk assessment. Yet difference arises as these have different purposes and ways to perform financial activities, in case of sovereigns such even have different purposes locating their motivation as a non-profit purpose, yet rather regulatory. That is why such category demands different approaches for credit risk in terms of variables and logic underlying the algorithm.[5]

Hence one could break it down that parts of credit risk assessment concerning machine learning are mostly associated with either credit scoring on the client and hence assigning a relevant score to a certain credit risk class, or calculating the amount of exposure expected to be lost based on a relevant estimated collateral value weighted on exposure maturity and risk associated with realization of such and probability that a certain individual or entity experience default.

1.3 Derivative usage of ML results in credit risk

Credit Scoring [3]

Credit scoring – is a specific algorithm that while handling input information of an entity or and individual subject to credit debt outputs a specific score value associated further with an internally defined risk classification.

Considering that, machine learning comes in handy when estimating parameters and way of data processing necessary to output latent score and even proceed with an accurate class prediction. One could incorporate machine learning techniques in already existing credit scoring models, since they already prove certain amount of robustness on a general landscape, while introduction of ML would tweak such to enhance performance for more specific environments.

Loan Loss Provisioning (LLP)

Loan loss provisioning, particularly under the International Financial Reporting Standard 9 (IFRS 9) framework, that remarks an income statement expense to cover expected volume of uncollected loan repayments. [8]

Genuinely, the last most notable changes in loan loss provisioning framework are represented below: [8]

1. Shift from Incurred Loss to Expected Credit Loss (ECL) Framework, where such requires banks to always recognize expected credit losses, considering past events, current conditions, and forecasted information opposing to usage of exceptionally recognized losses only when evidence of loss was apparent.
2. Mentioned above ECL framework applies to financial assets subject to IFRS 9's impairment accounting, including loans, lease receivables, loan commitments, and financial guarantee contracts and consequently is recognized in one of three stages:
    I. Stage 1: Initially, when a loan is originated or purchased, ECLs resulting from default events that are possible within the next 12 months are recognized. This is referred to as 12-month ECL. Interest revenue is calculated based on the loan's gross carrying amount.

II. Stage 2: If a loan's credit risk increases significantly since initial recognition and is not considered low, lifetime ECLs are recognized.

III. Stage 3: If the loan's credit risk increases to the point where it is considered credit-impaired, interest revenue is calculated based on the loan's amortized cost, and lifetime ECLs are recognized.

3. Additionally, maturity-wise there are two distinctive approaches. Twelve-month ECL represents the portion of lifetime ECLs associated with the possibility of a loan defaulting in the next 12 months, hence parameter estimation is performed in one year forecast horizon. At the same time, lifetime ECLs are an expected present value measure of losses that arise if a borrower defaults on its obligation throughout the life of the loan. [8]

   1) Meaning that under one year perspective, maturity factor is not used in determination of credit risk severity, rather the calculations are capped, and probabilities of default are uniformly distributed along the loan lifetime, cumulating yearly PD through the horizon for each specific contract i: [8]

$$\mathrm{ECL_{1Y}} = \sum_{\mathrm{i}} \min\left(\frac{\mathrm{M_i}}{365}\mathrm{PD_i^{1Y}}, \mathrm{PD_i^{1Y}}\right) \times \mathrm{L_i|D} \times \mathrm{EAD_i} \quad (1.1)$$

   where:

   $\mathrm{EAD_i}$ – exposure at default

   $\mathrm{PD_i^{1Y}}$ – one year-horizon probability of default

   $\mathrm{L_i|D}$ – losses given default occurred

   $\mathrm{M_i}$ – maturity denoted in days

   2) As an opposing perspective, expected losses under no maturity limitations are derived as follows: [8]

$$\mathrm{ECL} = \sum_{\mathrm{i}} \sum_{\mathrm{t}} \mathrm{L_i|D}_t \times \mathrm{CF_{i,t}} \times \mathrm{PD_i^t} \times \exp(-\mathrm{r}_i \mathrm{t}) \quad (1.2)$$

   where:

   $\mathrm{CF_{i,t}}$ – cash-flow associated with loan repayment in the contract schedule

   $\mathrm{PD_i^t}$ – probability of default at point in time t

   $\mathrm{L_i|D}_t$ – losses given default occurred at point of time t

r_i – continuous discount rate

These ensure sufficient improvement over timely recognition and provision for credit losses are important for safe and sound banking systems. The Basel Committee on Banking Supervision (BCBS) has recognized the close relationship between capital and provisions. [8]

Unexpected losses calculation in ICAAP economic perspective

Machine learning outputs are additionally to be subjects of consequential usage in risk modelling. Specifically, probability outputs per exposure, even though primarily used for credit allowance calculation and hence bank's provisioning, such represents exclusively portfolio`s loss expectation (EL). While such volume forms reserves, there is always a chance that the reserves volume may be breached when observing empirical losses over the forecast period (further as UL for unexpected losses). Given loss overflow is witnessed relatively to the provisions, such losses are to be incurred by the bank's capital. Provided such, second pillar of Basel II first introduced Internal Capital Adequacy Assessment Process (ICAAP), which requires banks to calculate credit risk volume which is relevant for the capital. Since this volume of losses specifically exceeds reserves and is then reflected as loss in respective financial statement, then such losses are hence deducted from the bank's capital. This specifically creates necessity for bank's capital to be sufficient for each additional risk accumulated in every financial risk category, credit risk being one of them. This is both captured in economic and normative perspectives of ICAAP, yet severe stress scenarios are mostly captured in one-year shock simulation for economic perspective. [21]

Given necessity of the risk modelling was reflected above, elaboration on the arrival at the UL value will be provided further. Suppose each client in the portfolio has an estimated probability of default, granted by the machine learning algorithm. This forms a certain distribution of the credit risk in portfolio and given that these are usually represented by qualitative categorical factors, such may be clustered as homogenous portfolio sections, as given ones have exact same factors, PD output is granted to be same consequently. Each cluster then is to be simulated in perspective of

one year until desired metric (UL) absolute difference converges within specified necessary margin of error. [21]

The usage of machine learning in credit risk assessments plays crucial role in quantitative determination of a specific level of severity associated with default process. Most common interpretation of such is either in a form of probability output of a certain direct or tangent event influencing credit risk volume, or straightforward determination of an object inherence to a certain risk class from an introduced range.

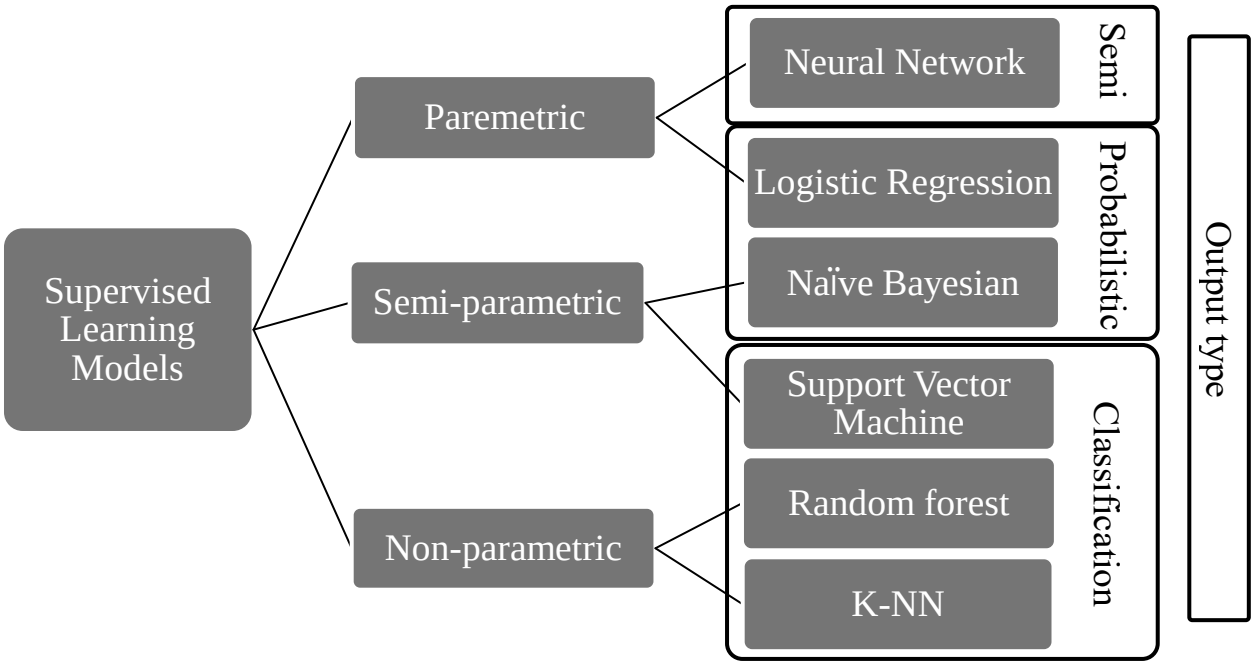
SECTION 2

CREDIT RISK MODELLING SOLUTIONS AND ITS IMPLEMENTATION

2.1 Common machine learning algorithms in credit risk

While not yet completely recognized on the regulatory level to be proposed as baseline algorithms for assessing credit risk, it is becoming more comprehensive and used as internal risk-based approaches. Referring to most recent research on this topic, the proposed algorithms to address various challenges of credit risk have been introduced. [3]

Among those also overviews were made with regards to data types and credit risk specifics, variables of the assessment and metrics used with limits. From observed above, this specific paper will observe risk modeling ML algorithms broken down by either type of estimation or the type of output that is granted by the model.



Fig, 2.1. Classification of common credit risk assessment algorithms.

As such may be observed from the presented scheme, there are parametric, non-parametric and semi-parametric algorithms involved in the classification problem solving. Such implies that a model can either exhibit a certain parameter selection that afterwards determines the latent variable transformation that directly influences risk

class or introduces a non-parametric object that defines a certain plane separation for the input data to determine the class further.

These can also exhibit other classification type on regards if the granted output and nature inside the algorithm, specifically being probabilistic, classifying and semi (can handle both outputs). They can be used differently according to the specific problem addressed by the modelling, for example calculation of loss provisions per exposure would require expected loss calculation, which is associated with probability of default estimation. At the same time determination of the risky client while handling decision on if loan should be accepted can be better achieved with classification algorithm, which will be observed more in-depth in the next pages.

In the context of determining factors for credit scoring models, demographic indicators constitute the first category, encompassing regional, gender, and other differences. While not the most critical, these variables, such as age, gender, and marital status, play a role in distinguishing between risky and less risky applicants. The second category involves financial situation data, crucial for a bank to assess a household's available resources, income, costs, and potential monthly payment capacity. The third set comprises variables related to income source and employment status, with stability in employment and its duration being significant factors in scoring. Lastly, the fourth category focuses on behavioral characteristics, providing valuable information for credit scoring. A client's history with a bank, including loan repayment and collateral value, helps mitigate information asymmetry. Collateral, particularly real estate, emerges as a dominant factor influencing both a bank's lending decision and a client's commitment to debt repayment, given the potential risk of losing valuable assets like a house in the event of default.

2.2 Derivation of the overviewed algorithms

Financial Data Pre-processing

It is essential to maintain high quality of training data to achieve correct relationship between the predictor and predicted variables, and such are achieved as next:

Normalization

Most notable from such are min-max and standard data rescaling, however in terms of financial data second one is found more usable since it does not treat normalized variables to be in closed interval. [11]

$$\mathrm{N}_{\mathrm{Min-Max}} = \frac{\mathrm{x}_i - \min(\mathrm{x})}{\max(\mathrm{x}) - \min(\mathrm{x})} \quad (2.1)$$

$$\mathrm{N}_{\mathrm{Std}} = \frac{\mathrm{x}_i - \sum_j \frac{\mathrm{x}_j}{N}}{\sqrt{\sum_k \frac{\left(\mathrm{x}_k - \sum_j \frac{\mathrm{x}_j}{N}\right)^2}{N}}} \quad (2.2)$$

Outlier cut-off

If data is already normalized, depending on what was the algorithm, there are different approaches for cleaning outliers. One is based on cutting of values falling out of a specific z-score value corresponding to a selected percentile, after which values are considered outliers, and quantile filtering, where data falling in defined quantiles from both tails are classified as outliers and hence are deleted from the data. [11]

Splitting

Once previous steps are done, data is hence split into samples for training, validation and testing purposes. Additionally, training data in terms of financial credit risk modelling can be severely imbalanced since occasion of default is an extreme event. For handling such, synthetic minority oversampling technique (SMOTE) is applied, which is interpolating data based on closest neighbors, synthetically creating similar events of default, yet providing additional deviation and hence new information for the process of parameter learning. [11]

Logistic Regression

The main concept of this algorithm in credit risk assessment is that such given observed variables X may form a latent expression, which is rather a score after being passed through sigmoid activation reflect a probability output of an observation Y equality to a certain risk class.

Proper model performance is constrained by compliance with the following assumptions: observation independence, absence of outliers and multicollinearity, homoscedasticity of the residuals and equality to one of sum within observation adherence to each class.

To produce a latent variable, referred also as logit, identity matrix of input X is linearly transformed using ω matrix and hence equalized to an expression of log-odds as to associate score of the latent variable with a probability expression. The reason why direct use of probabilities $\mathrm{p_i}(\mathrm{Y}=1)$ is neglected over log-odds are because since probability range is defined as [0,1], it should be transformed to a continuous infinite plane, where $\frac{\mathrm{p_i}(\mathrm{Y}=j)}{\sum_{k\neq j}\mathrm{p_i}(\mathrm{Y}=k)}$ grants $[0, \infty)$ range and $\ln\left(\frac{\mathrm{p_i}(\mathrm{Y}=j)}{\sum_{k\neq j}\mathrm{p_i}(\mathrm{Y}=k)}\right)$ grants $(-\infty, \infty)$ range:

$$\mathrm{logit}\left(\mathrm{X_i}, \omega_j\right) = \mathrm{X_i}\omega_j^{\mathrm{T}} = \ln\left(\frac{\mathrm{p_i}(\mathrm{Y}=j)}{\sum_{k\neq j}\mathrm{p_i}(\mathrm{Y}=k)}\right) \quad (2.1)$$

Solving for $\mathrm{p_i}$ grants sigmoid function:

$$\mathrm{p_i}(\mathrm{Y}=\mathrm{j}) = \frac{1}{1+e^{\mathrm{X_i}\omega_j^{\mathrm{T}}}} \quad (2.2)$$

Estimation of the weight and bias matrix ω is done by maximizing log-likelihood of the observations, more referred to in machine learning as negative cross-entropy loss. Assuming observations are independent, it maximizes the probability to observe current data belonging to respective class j estimates by fitting each parameter of ω.

$$\mathcal{L}(\omega|\mathrm{X},\mathrm{Y}) = \prod_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{p_i}(\mathrm{Y}=j) \quad (2.3)$$

Since log-space does not shift the extrema points the minimization expression is next:

$$\mathrm{argmax}\,\mathcal{L}(\omega|\mathrm{X},\mathrm{Y}) = \mathrm{argmax}\left(\ln\left(\mathcal{L}(\omega|\mathrm{X},\mathrm{Y})\right)\right) = \mathrm{argmax}\left(\sum_{\mathrm{j}=1}^{\mathrm{k}}\ln(\mathrm{p_i}(\mathrm{Y}=j))\right) \quad (2.4)$$

There are three most used and researched types of logistic regression:

- Binary – is used to predict two-class probability (e.g. default/non-default), the algorithm itself was presented above.

- Ordinal – is used to predict a multiclass probability set yet considers ordered nature of the class (e.g. ordered risk feature classification). Typically, assignment to the class is performed by partitioning the logit score in certain interval bins:

$$Y_i = \begin{cases} 1, & \text{if } X_i\omega^T \leq \hat{t}_1 \\ 2, & \text{if } \hat{t}_1 < X_i\omega^T \leq \hat{t}_2 \\ & \dots \\ N, & \text{if } \hat{t}_{N-1} < X_i\omega^T \end{cases} \quad (2.5)$$

- Multinomial – is used to predict a multiclass probability set which does not exhibit same property (e.g. given company experienced default and is the distribution of LGD is being estimated according to categories: “collateral covered”, “court: uncovered”, “collateral depreciation: uncovered”, etc.). To perform such each class has its own unique weight and bias parameters influencing overall odds of observing such a class, which is summarized by output of a SoftMax function:

$$p_i(Y = j) = \frac{e^{X_i\omega_j^T}}{\sum e^{X_i\omega_k^T}} \quad (2.6)$$

Concluding the stated, ability of the logistic model to output probability of different risk class on a set of predictor variables makes it well-suited for predicting and quantifying credit risk, however it is still lacking interpretability in terms of classification and requires thresholding approaches to neglect them. Adding to that, following all assumptions is a rare case to be satisfied, lowering the consequent quality of the model. [9]

Bayesian Network

This classification algorithm uses core principle of Bayesian statistics such that conditional probability of certain arbitrary event A given another independent event B are found by exclusion of the probability of individual event B from its mutual probability: [10]

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = P(A)\frac{P(B \mid A)}{P(B)} \quad (2.7)$$

Following this as a baseline, joint probability of observing certain class C and factors X is product of its subset conditional probabilities: [10]

$$P\left(C_k, \bigcap_{X_i \in X} X\right) = P(C_k) \prod_{i=1}^{n} P(X_i | X_{i+1}, \dots X_n, C_k) \quad (2.8)$$

Assuming independence next statement is concluded of factors X, observing X_i given C_k is exactly same as observing $X_{i+1}, \dots X_n$ alongside:

$$P(X_i | X_{i+1}, \dots X_n, C_k) = P(X_i | C_k) \quad (2.9)$$

Since option available for calculation is $P(X_i | C_k)$, a property of proportionality can be expressed as: [10]

$$P\left(C_k | \bigcap_{X_i \in X} X\right) \propto P(C_k) \prod_{i=1}^{n} P(X_i | C_k) \quad (2.10)$$

Including this proportionality statement, the fractions at which expressions are proportional is $\frac{1}{p(X_{i+1}, \dots X_n)}$, being evidence scaling factor: [10]

$$P\left(C_k | \bigcap_{X_i \in X} X\right) = \frac{P(C_k) \prod_{i=1}^{n} P(X_i | C_k)}{p(X_{i+1}, \dots X_n)}$$

Given our assumption of independence same Bayesian formula introduced at the start was achieved. Enhancing formula further by expanding the $p(X_{i+1}, \dots X_n)$ statement grants the final step: [10]

$$P\left(C_k | \bigcap_{X_i \in X} X\right) = \frac{P(C_k) \prod_{i=1}^{n} P(X_i | C_k)}{\sum_{C_k \in C} P(C_k) P(X_{i+1}, \dots X_n | C_k)}$$

Hence the class decision is determined of maximum a posteriori: [10]

$$k = \text{argmax}_k P\left(C_k | \bigcap_{X_i \in X} X\right) \quad (2.11)$$

Since argmax value is at demand we can use logarithmic scale due to perseverance of extrema locations: [10]

$$k = \operatorname{argmax}_k \ln\left(P\left(C_k \middle| \bigcap_{X_i \in X} X\right)\right) \qquad (2.12)$$

Depending on the data type there are several approaches for Naïve Bayesian classification:

- Multinomial – assumes data follows categorical distribution. Can be generated from discretizing continuous distribution in interval bins and assigning each bin a category.

  The probability distribution of data point class is given by: [11]

  $$P\left(C_k \middle| \bigcap_{X_i \in X} X\right) = \frac{\left(\sum_{X_i \in X} X_i\right)!}{\prod_{X_i \in X} X_i!} \prod_{i=1}^{n} P(X_i | C_k)^{X_i} \qquad (2.13)$$

  Omitting constants and using log-space the score of the classifier is linearized to: [11]

  $$k = \operatorname{argmax}_k \left(\ln(p(C_k)) + \left(\ln\left(P(X_i | C_k)\right)\right)^T X\right) \qquad (2.14)$$

- Continuous parametric – assumes data follows continuous parametric distribution, typically gaussian normal distribution.

  The probability distribution is given by any assumed probability density function (e.g. normal): [11]

  $$P(X = x \mid C_k) = \frac{\exp\left(-0.5(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}{\sqrt{(2\pi)^D \det(\Sigma)}} \qquad (2.15)$$

  Generally, fails to satisfy assumptions of parametric distribution chosen to model data behavior, yet incorporates more continuous granularity in the score output and hence smoothens the discriminatory property.

- Bernoulli – assumes data is represented by Boolean variables. Can be generated from continuous data using “if-else” statements. [11]

$$P(X = x \mid C_k) = \prod_{i=1}^{n} P(X_i \mid C_k)^{X_i} \left(1 - P(X_i \mid C_k)\right)^{1-X_i} \quad (2.16)$$

When it comes to credit risk assessment the correlation within the predictors is likely to be present, suggesting lack of independence within the estimated probabilities.

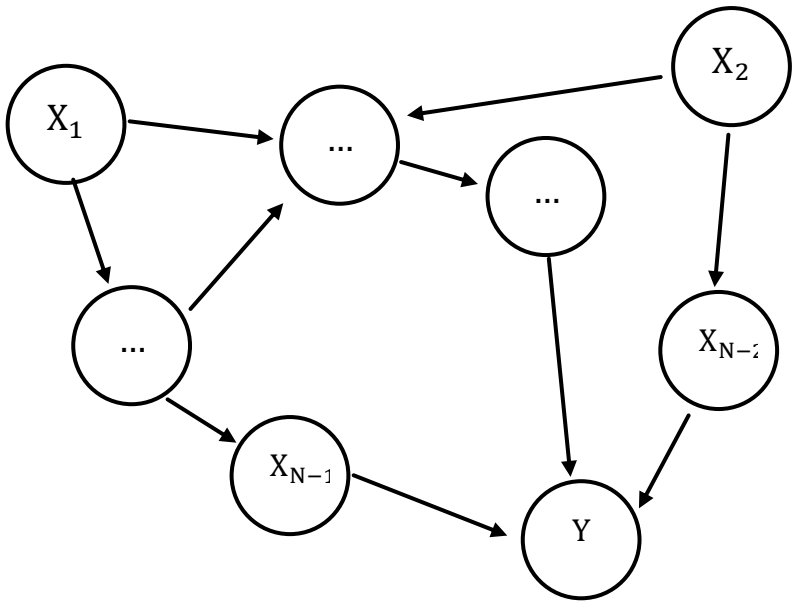


Fig. 2.2. Bayesian Network structure [generated by author]

Conceptual background underlying in such formula is that the posterior probability of an event can be represented as likelihood of such event multiplied by the prior probability with exclusion of the evidence associated prior. Such holds an explicit assumption of the independence of the conditional probabilities, which remains true for the model itself. However, such assumption is never meant to be followed completely, as such is impossible in the real-world practice. At the same time, if the independency level within the data tends to the given assumption, Naïve Bayesian potential to outperform other models rises respectively.

The structure itself of a Bayesian Network represents a Fig. model, where certain connections between nodes, variable state probability distribution, reflect active paths off the impact, migrating through as a prior to posterior until reaching the output node. Bayesian networks, provide advantage relevant to random forest algorithm, where actual interaction throughout the data may be well inspected, however on contrast to ensemble logic, it incorporates how, and which exactly certain priors affect respective

posteriors in probabilistic manner. What makes it more usable than random forest, is that the figure describing the relationships in between the variables is limited to number of variables, whatever the complexity of the connections, while partitioning per each random tree can amount to an enormous amount of variable processing. [11]

Typically, Bayesian networks are non-lazy algorithm, meaning that in credit risk assessment the logic among the variables is constructed manually. Such may be capable in handling while variable number is sufficiently small, however, one could speculate that there can be an automation involved, testing each structure, given a specific separation of the variable on input and target output ones. The structure is considered optimal if negative cross entropy loss between target output predictions and real output observations is maximized. Such structure can additionally provide valuable insight on the data structure: specifically which variables are formed in the offspring relationships. [11]

Generally, advantage of Bayesian networks allows for interpolating most expected information regardless of already present one. Hence its usage becomes most uniquely valuable to fill in most expected values for gaps in the data to suffice the assessment. Since overall research argue that independence assumption affects prediction output of the Bayesian classifier heavily, its usage will be limited to its possibility to interpolate data to its most likely values. Given the probability distribution is the output per each variable, its expected value, matrix product of probabilities per value is taken as variable interpolation is considered. Interpolation may additionally provide error distribution, for each collection of variable observations, error between the combination of interpolated values and real observed value in case of its absence may be tested, by constructing the error distribution and calculating RMSE/accuracy depending on the data and hence Bayesian network type. This approach additionally provides margin of error in the interpolation and hence may output respective confidence level for the interpolated value. This is valuable since in credit risk assessment, risk conservancy prevails over accuracy, yet overestimating risk is a low-quality indicator of the model.

Random Forest

A random forest is an ensemble learning method in machine learning that builds in credit risk would be collection of independent unique decision trees predicting certain risk class. Decision tree by itself performs a discriminative analysis of how the data should be partitioned using thresholds to arrive sufficient factor separation such that output is correctly clustered. [12]

The reason of the multiple tree decision is aggregated then into one “voting” solution is because singularly threshold partitioning leads to extensive data overfit. Hence for a given collection of factors X and output classes of credit risk assessment Y parameters determining structure of random forest is the number of trees and number of partitioning in each tree. Random trees by themselves are overly flexible to the data input, making their variance excessive and bias low. Therefore, random forests introduce variance correction by a more determinant averaged decision. [11]

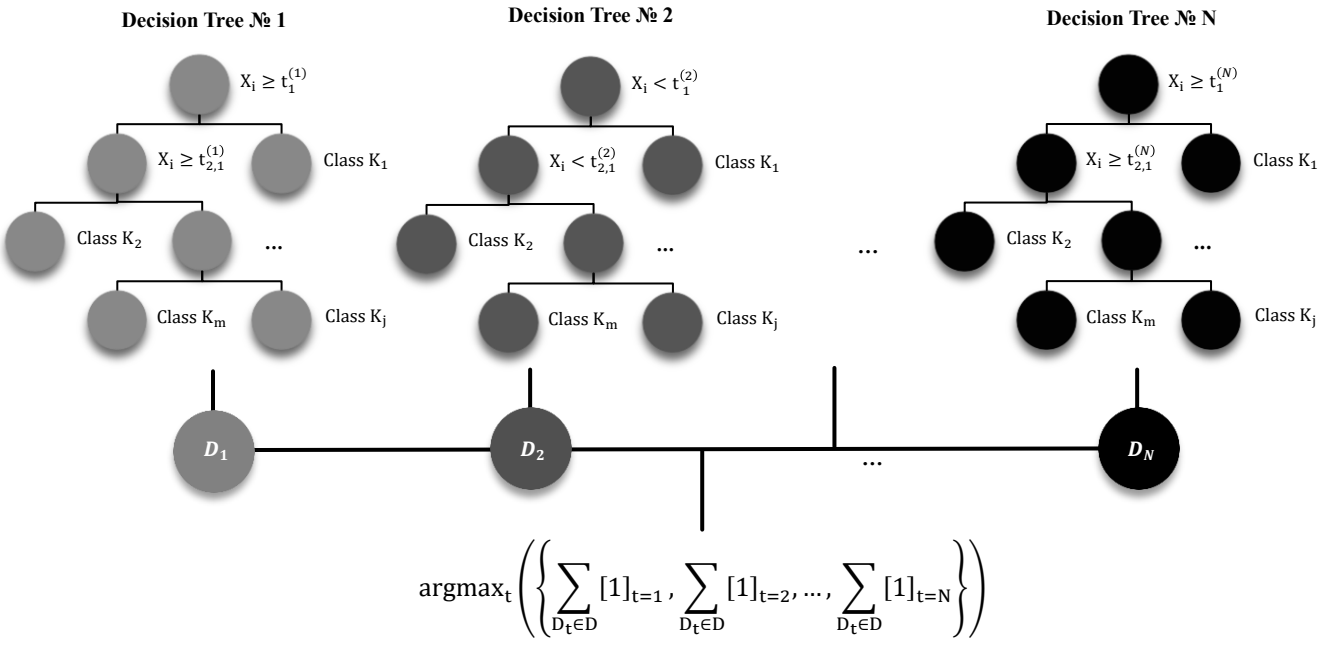


Fig. 2.3. Random Forest classification process. [generated by author]

Another important feature of random forest that unifies random trees lies in the learning algorithm. Thresholds by itself are determined by an algorithm, where data I input is bootstrap aggregating and then recursively splitting data on the principle of most significance. This introduces the independence within different decision trees due

to randomness caused by bootstrap sampling. Considering the feature is truly random, random forests reduce the divergence withing caused by the features mentioned previously.

However, this algorithm exhibits struggles when facing high dimensionality of the factor input, imbalanced datasets, overfitting noise in the data. Yet given the drawbacks this algorithm provided some alternative methodologies for determining feature importance within the data set. Those can be variable, permutation and impurity importance metrics of the model, in a way qualifying data quality. Additionally, it can be performed for cauterization of previously unclassified training dataset, where research shows that it tends to outperform simple k-means and PAM algorithms, making it a more robust unsupervised machine learning algorithm.

Support Vector Machine

SVMs on contrast to logistic regression are typically constructed as such to output not an accurate probability estimate, but rather a more precise class. The reason for that is because rather than applying a likelihood loss estimator for the parameters, the estimation differs depending on a margin class, which can be divided into soft and hard. [6]

Hard margin classifier finds the most separating hyperplane, where each different class belongs to respective region created by hyperplane separation. On the other hand, soft margin classification involves minimization hinge loss function, which is penalizing model more for unconfident predictions, motivating it to grant more extreme values when applying classifying threshold, while still granting output as a certain “confidence” value. Since credit risk may incorporate a lot of outliers, hard margin classifiers will most definitely include hyperplane overfitting, resulting to a false data classification, hence soft margin approach will be observed. [11]

The mathematical expression of hinge loss is given as following: [11]

$$\mathrm{L}(\mathrm{Y},\mathrm{X},\mathrm{w},\beta) = \left[\frac{1}{\mathrm{n}} \sum_{\mathrm{Y_i}\in\mathrm{Y}\cup\mathrm{X_i}\in\mathrm{X}} F(Y,\zeta(X,w,\beta))\right] + \lambda\|\mathrm{w}\|^2, \text{ where:} \quad (2.17)$$

$$\zeta(X, w, \beta) = \max\left(0, 1 - \mathrm{Y_i}(\mathrm{X_i w^T} - \beta)\right), \text{as inverse ReLU, linear penalization}$$

$$\zeta(X, w, \beta) = \ln\left(1 + \exp\left(-\mathrm{Y_i}(\mathrm{X_i w^T} - \beta)\right)\right), \text{as inverse softmax}$$

where:

- Y – observation class,
- X – observation factors,
- w – factor weight matrix,
- β – bias matrix,
- λ – partitioning cost

The core principle lying inside SVMs is that they are developing a certain hyperplane or a vector, that segregates the classes into different areas, where each class type is defined by the membership of respective region. These can be divided into kernel types, such as simple linear, polynomial, and radial. [6]

$$\mathrm{K(u, v, \gamma, \alpha, k)} = \begin{cases} \mathrm{u^T v}, & \text{if type: linear} \\ (\gamma \mathrm{u^T v} + \alpha)^{\mathrm{k}}, & \text{if type: polynomial} \\ \exp(-\gamma|\mathrm{u} - \mathrm{v}|^2), & \text{if type: radial} \\ \tanh(\gamma \mathrm{u^T v} + \alpha), & \text{if type: sigmoid} \end{cases} \quad (2.18)$$

where:

- γ – data leveling complexity,
- k – polynomial degree,
- α –inhomogeneity coefficient

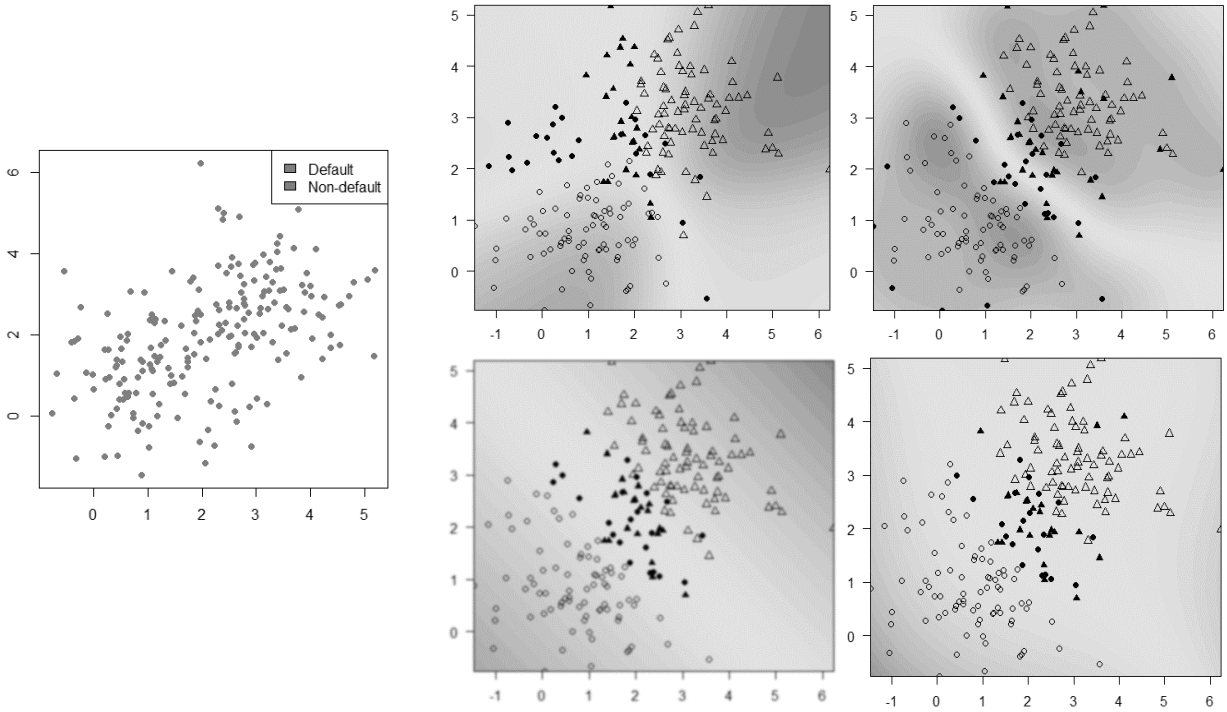


Fig. 2.5. SVM algorithm visualization based on the different kernel types. [generated by author]

Kernels allow the non-linearity within input-output relationships of the classification problem. The main parameters except of kernel types can be broken down to gamma (to what extent a solitary training instance's impact extends), regularization (extent to which you aim to minimize misclassification for each training instance). [6]

In summary, Support Vector Machines (SVMs) prioritize precise class output over probability estimates. The choice between hard and soft margins in SVMs is crucial, with soft margins preferred for credit risk scenarios to avoid overfitting. It is essential to maintain the robustness of SVMs lies in their ability to handle outliers and non-linear patterns, making them a valuable tool for risk assessment.

K-nearest Neighbors

K-nearest neighbors is a non – parametric machine learning algorithm, that classifies an observation based on the minimum normalized Euclidian distance principal conducted within sample based on k-closest observations. It can be also extended to a weighted version, where each observation distance can be weighted

depending on the distance, namely the score of the closer observations is granted more impact on the classification output. Such introduces improved accuracy, because of the distances being treated non-uniformly.

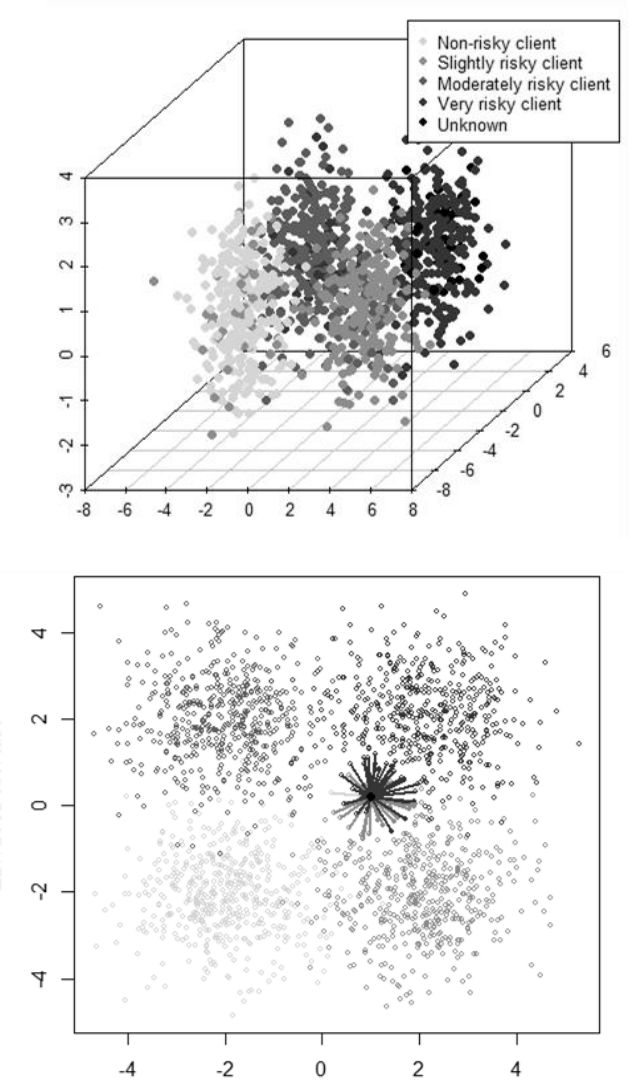


Fig. 2.6. K-NN algorithm visualization. [generated by author]

Suppose an unclassified point (also referred as "query point") determined Y_q in a multidimensional plane by n distinct parameters $\{X_{q,1}, X_{q,2...} X_{q,N}\}$, that can take category values from a set $C = \{C_1, C_{2...} C_K\}$, where length of the observations is L. Hence classifying sample is then determined by the k closest observations, where distance of j-th observation Y_j with assigned rank r_j is given by: [13]

$$d_j = \sqrt{\sum_{i=1}^{N} \left(X_{q,i} - X_{j,i}\right)^2} \rightarrow (d_j, r_j, Y_j) \qquad (2.19)$$

.

The count of each category in subset minimal distance sample summarizes the score of an ordinary k-NN algorithm, where index of its maximum is the index of the classified category m: [13]

$$m = \operatorname{argmax}\left(\left\{ \sum_{j=1}^{N} [Y_j = C_1], \sum_{j=1}^{N} [Y_j = C_2], \ldots \sum_{j=1}^{N} [Y_j = C_K] \right\}_{r_j \leq k} \right)$$

If weighted distance approach is applied, then the score is determined by some by sum of weights, which respectively may be represented as a certain function of the distance. A common practice is to use reciprocal relationship with a degree of distance significance α, however it can be represented as any continuous monotonously decreasing function defined on the range of (0, ∞):

$$w_j(d_j) = \frac{\alpha}{d_j} \qquad (2.20)$$

$$m = \operatorname{argmax}\left(\left\{ \sum_{j=1}^{N} w_j(d_j)[Y_j = C_1], \sum_{j=1}^{N} w_j(d_j)|[Y_j = C_2], \ldots \sum_{j=1}^{N} w_j(d_j)|[Y_j = C_K] \right\}_{r_j \leq k} \right)$$

(2.20)

While k-NN does not require training parameters, making it a fast and intuitive algorithm as well as flexible for various data types, enhancing its applicability in credit risk classification, it is highly sensitive to outliers and was noted to have a “curse of dimensionality”, meaning that in high-dimensional spaces, the concept of proximity becomes less meaningful leading to reduced performance. In addition, optimal value of k is a major impact the performance of the algorithm, making it complex to retrieve the optimal number of observations to weight on.

Neural Network

Such type of machine learning algorithm is sufficiently recognized for its flexibility and diversity. It uses a complex layer system which may vary depending on the purpose of the modeling. The very core of ones lies in such objects as hidden layers, neurons, activation functions, weights, and biases.

Core advantages and challenges of such models can be summarized in such phenomenon as overfitting, while also introducing powerful solution to non-linearities in the data.

Suppose a $n_N \times k_x$ matrix to be processed by the neural network representing a certain input affecting an observed result in a form of a matrix with dimensionality $n_N \times k_y$, where n_N is number of sampled observations, k_x and k_y are numbers of factors and results observed respectively. This output is then linearly translated on each neuron of consecutive hidden layer with unique weight for all the observations of each input type along with additional constant bias. Hence linear transformation is passed into an activation function on the specific neuron to present non-linearity in the system, being the essential part of the neural network flexibility. This process is then recursively translated into each consecutive hidden layer, until it reaches its output state. Such process is summarized below: [14]

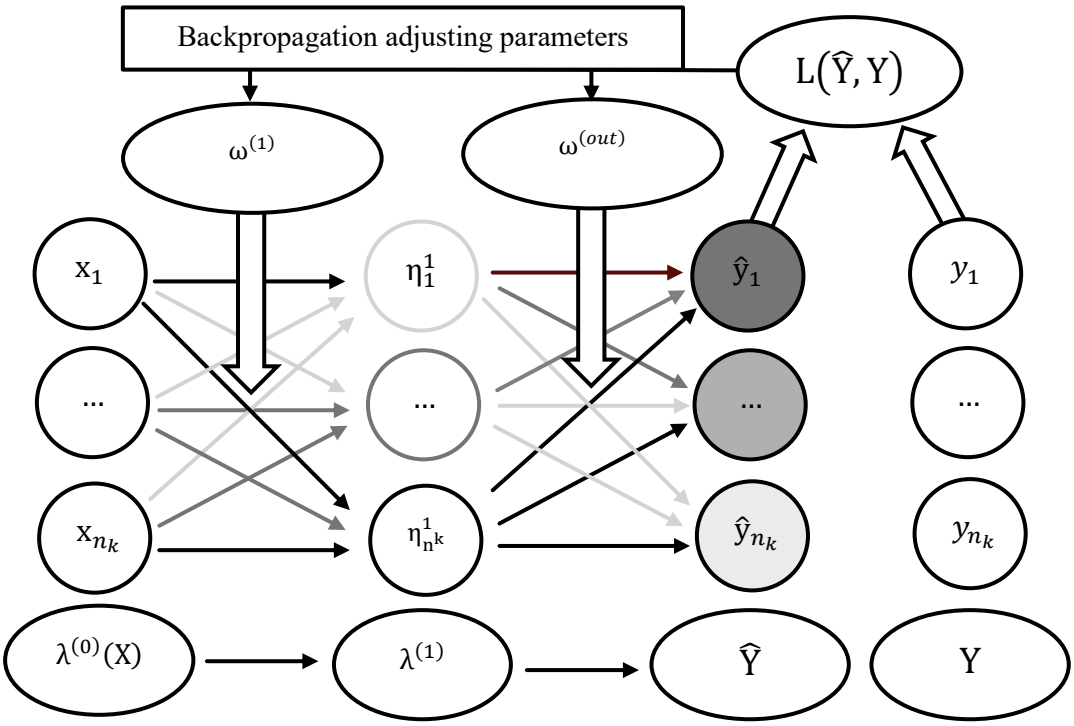
$$X = \lambda^{(0)} \rightarrow \text{for } k \in \{1,2 \dots N\}: \lambda^{(k)} = f_k\left(\lambda^{(k-1)}\left(\omega^{(k)}\right)^T\right) \rightarrow \lambda^{(N)} = \hat{Y} \quad (2.21)$$

$$\omega^{(k)}, \beta^{(k)} \rightarrow \nabla L\left(\hat{Y}, Y\right) = 0$$

- $X -$ input matrix
- $Y -$ result matrix
- $\omega^{(k)} -$ parameters, defined by i^{th} input on j^{th} neuron of k^{th} layer
- $f_k(x) -$ activation function on j^{th} neuron of k^{th} layer
- $L\left(\hat{Y}, Y\right) -$ loss function
- $\eta_j^{(k)} -$ j^{th} neuron output of the k^{th} layer
- $n^{(k)} -$ objects (neurons/inputs/ouputs) of neurons on k^{th} layer
- $N -$ number of layers

- $\lambda^{(k)} - k^{th}$ hidden layer output
- $\hat{Y} -$ result matrix

Its logic is derived that certain result may have non-linear dependency relative to the factor, hence catching such non-linearities demands respective transformation on the input side of the model, which is performed on the hidden layer, where all the properties are gained to be transferred to the output state. The algorithm is summarized in a scheme below: [20]



Forward propagation deriving output

Fig. 2.7. Scheme of neural network algorithm (NN) [generated by author]

Such is a core type among neural network algorithms, as mentioned before it has one hidden layer to process the input for transforming it further into an output state. While in fields where artificial intelligence is at its core such type is rarely used for its lack of flexibility and facing issues while learning on the data, such can be irrelevant for credit risk analysis, where oversophisticated models can be overfitting the results and show poor results in comparison to models that are less complicated. Moreover, such network requires less parameters to estimate, making gradient descent more stable

on a low data volume, reaching global minimum of the loss function with less friction. [15]

If number of hidden layers is extended in a such manner, that consecutive hidden layer recursively takes input of current one till reaching output state, then such extension of artificial neural network grants feed-forward neural network. Following the name of the algorithm it assumes that certain input requires multiple transformations to reach true output state. [20]

As well as artificial neural network it is one of the first types of net algorithms to be discovered, yet such is more sophisticated in its manner as may be observed. While it can introduce more insightful patterns in the data, it may be more sensitive in the noise in the data. [15]

The reason why neural networks can address non-linearity with high efficiency lies inside the neuron activations, more specifically non-linear transformations (linear transformation may be considered too once other layers contain non-linear activations) that transform the output in the manner to best fit the correspondence of output layer to the true observations.

2.3 Additional Usage of ML outputs in terms of Credit Risk

The target provided above shall be researched under the following logic. Given the information of the credit risk severity (estimated value of default probability) suggesting both business specific and macroeconomic qualities have been already reflected, the distribution of portfolio expected loss can be calculated with usage of respective exposures and recovery rates. Theoretical background suggests, that assuming the feature of default occurrence is independent inside the portfolio or specific portfolio clusters, for a given combination of realized defaults j, losses are distributed using multinomial distribution of N binary variables, where such corresponds to number of contracts:

$$\mathrm{PL}(\mathrm{D}_1 = \mathrm{d}_1, \mathrm{D}_2 = \mathrm{d}_2, \dots, \mathrm{D}_\mathrm{N} = \mathrm{d}_\mathrm{N}) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{P}\left(\mathrm{D}_\mathrm{i} = \mathrm{d}_{\mathrm{i,j}}\right) \quad (2.22)$$

In such case $P(D_i = d_i)$ corresponds to probability of observing clients state D_i to current empirical state d_i:

$$P(D_i = d_{i,j}) = \begin{cases} PD_i, & D_i = 1 \\ 1 - PD_i, & D_i = 0 \\ \emptyset, & D_i \notin \{0,1\} \end{cases} \quad (2.23)$$

Hence, the parametrization of the loss distribution is given that for each combination of losses exists a certain probability of occurrence given scenario of defaults j:

$$(L_j, PL_j) = \left(\sum_{i=1}^{N} d_{i,j} \times EAD_i \times L_i|D, \prod_{i=1}^{N} P(D_i = d_{i,j}) \right) \quad (2.24)$$

EAD_i – exposure at default

PD_i – probability of default

$L_i|D$ – losses given default

These sets are then reorganized in a manner that of increasing order subject to losses.

Supposing cumulative loss distribution subject to percentile in one year perspective $CLD(\alpha)$, such broken down as follows.

$$CLD^{-1}(\alpha) = (L_j)_{\sum PL_j = \alpha} \quad (2.25)$$

However, to achieve such function all the possible combinations of default events should be considered, making estimation of such $O(2^N)$ complexity, meaning the calculations grow exponentially for an additional contract in the loan portfolio. Given a sufficiently large portfolio, there is no available computational power to solve such a task.

Core parameters are defined as follows from formula suggested in one year perspective impairment forecast (formula 1).

Supplementary parameters are given as follows:

- η – minimum sample size
- ε – accepted error margin
- k – window of error retracement
- ω – window of insurance
- α – survival rate (proportion of occurring losses to be covered by economic capital)

$$\mathrm{UL}_{\alpha} = \mathrm{CLD}^{-1}(\alpha) - \mathrm{LLP}, \qquad (2.26)$$

where:

$\mathrm{CLD}(\alpha)$ – cumulative loss distribution in one year perspective

LLP – loan loss provisions

The formula is then broken-down step by step in a following manner:

For each simulation n vector d_n is generated, which contains simulated scenarios regarding each independent default event of the contract, based on the next logic: [16]

$$\mathrm{d}_{\mathrm{i,n}} = \begin{cases} 1, & \mathrm{PD}_{\mathrm{i}} \leq \mathrm{u} \\ 0, & \mathrm{PD}_{\mathrm{i}} > \mathrm{u}, \end{cases} \qquad (2.27)$$

$$\mathrm{u} \sim \mathrm{U}(0,1)$$

Consequently, for each generated scenario respective losses are calculated:

$$\mathrm{L}_n = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{d}_{\mathrm{i,n}} \times \mathrm{EAD}_{\mathrm{i}} \times \mathrm{L}_{\mathrm{i}} | \mathrm{D} \qquad (2.28)$$

Afterwards, n-th scenario loss is appended to all the previous generated set of losses: [16]

$$\widehat{\mathrm{CLD}}_{\mathrm{n}}^{-1} = \widehat{\mathrm{CLD}}_{\mathrm{n-1}}^{-1} \cup \mathrm{L}_{\mathrm{n}} \qquad (2.29)$$

Hence, losses that are considered unexpected is value of LD_{n} that corresponds to α-th quantile of the set, excluding allowances as such are to be carried by economic capital, as given in initial formula.

However, simulated set $\widehat{\mathrm{CLD}}_{\mathrm{n}}^{-1}$ is only an ***approximation*** of theoretical inverse distribution $\mathrm{CLD}^{-1}(\alpha)$. Such means a certain error may be produced due to unlikely scenarios being not captured in the algorithm output. These are located in both left and right tails of the distribution, that significantly affect unexpected loss estimate at extreme values of α, which are used regardless to satisfy risk-conservancy (e.g. 99%, 99.9% etc.). To resolve such discrepancies, bank estimates sufficient number of simulations by allowing a certain sufficiently small accepted margin of error ε. [18]

As a first step, absolute difference within the loss distribution at j-th iteration when additional simulation is appended. [18]

$$\Delta_{\mathrm{j}} = \left|\mathrm{UL}_{\alpha,\mathrm{j}} - \mathrm{UL}_{\alpha,\mathrm{j}-1}\right| \qquad (2.30)$$

However, scenarios are simulated with a higher likelihood to remain close to expected loss value, meaning that overall error remains relatively small, showing rare major error burst-outs when unlikely outcome is being generated. To remain consistent, bank introduces set of last m smoothed errors Δ_{j} to be monitored using moving average algorithm with window size k, introducing lagged conservancy.

$$\delta_m = \left\{\frac{1}{\mathrm{k}} \sum_{\mathrm{n-k-m}}^{\mathrm{n-m}} \Delta_{\mathrm{j}}\right\}_{0 \le m \le \omega} \qquad (2.31)$$

Hence, distribution convergence is admitted provided next conditions are satisfied:

- $\mathrm{K}_1 \Rightarrow \bigwedge(\delta_m < \varepsilon)$ – first condition ensuring all smoothed errors δ_m are below allowed margin of error ε.
- $\mathrm{K}_2 \Rightarrow \mathrm{n} \geq \eta$ – second condition ensuring all minimum sample size η was generated.

These two are hence unified under condition K showing both are satisfied.

To remain buffer on the potential uncaptured errors, bank takes in biggest absolute error burst-out among last k values and adds it to the value of loss distributions.

$$\mathrm{CLD}^{-1}(\alpha) \lesssim \widehat{\mathrm{CLD}}_{n}^{-1}|K + \max\left(\{\Delta_j\}_{n-k\leq j\leq n}\right) \lesssim \mathrm{CLD}^{-1}(\alpha) + \varepsilon \qquad (2.32)$$

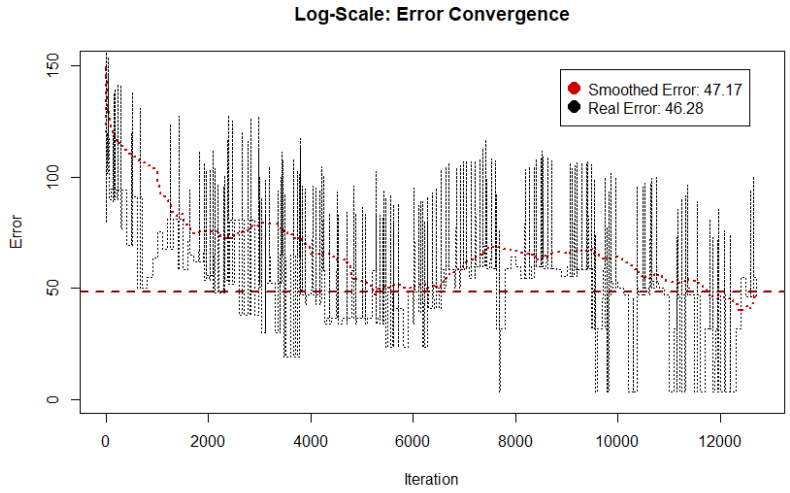


Fig. 2.8: Loss distribution inter-iteration error evolution. [generated by author]

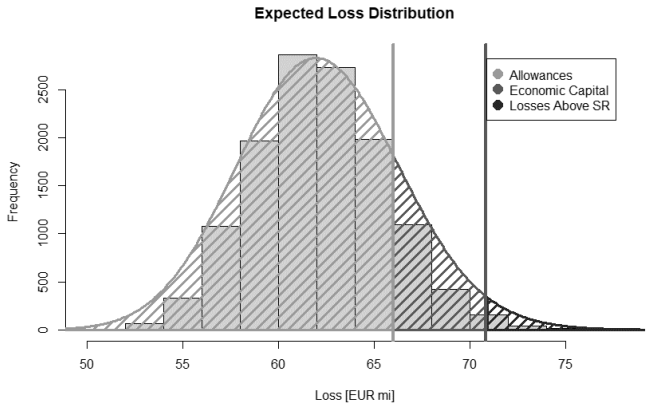


Fig. 2.9: Loss distribution results. [generated by author]

Generated extreme loss UL_{α} already assumes universal default occurrence increase independently over all clients. However, even though such does incorporate specific severity in correlations, the severity may be increased performing simulations per specific clusters and hence extracting per cluster unexpected losses. The reason for that, is that such approach reorders losses inside each cluster c, meaning that rank correlation matrix of such is specifically identity one:

$$\Sigma = \begin{pmatrix} 1 & \cdots & \rho_{i,1} \\ \vdots & \rho_{i,j} & \vdots \\ \rho_{1,j} & \cdots & 1 \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & 1 & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \qquad (2.33)$$

$$\rho_{i,j} = 1 - \frac{6\sum_{\alpha\in[0,1]}\left(R\left(UL_{\alpha,i}\right) - R\left(UL_{\alpha,j}\right)\right)}{N(N^2-1)} = 1 \qquad (2.34)$$

This corresponds to a higher severity of loss in the first place when observing extreme right tails of the distribution. Suppose a three-cluster separation of the general population, that hence is individually simulated as provided below:

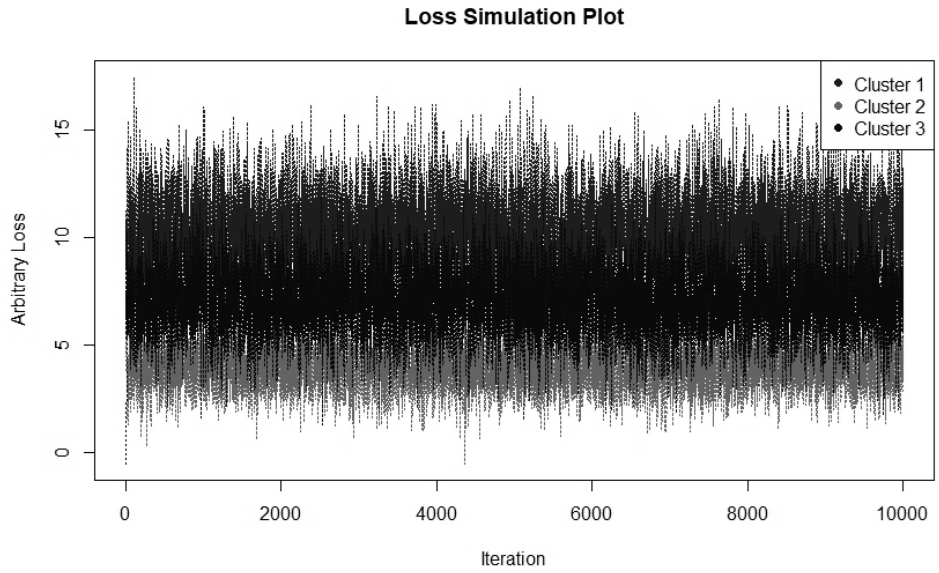


Fig. 2.10: Simulation of an arbitrary losses per cluster. [generated by author]

Then, calculation performed on both first aggregating losses and extracting loss according to survival rate and first individually performing such operation for such to be aggregated in a loss afterwards:

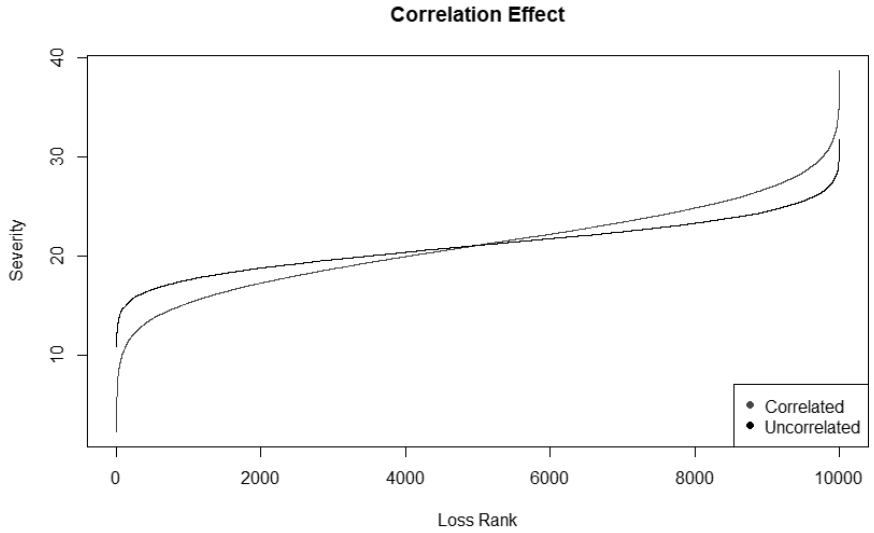


Fig. 2.11: Visual representation of rank correlation effect.

While this may serve as severity increase of the outcomes, it is worth mentioning that they must be backed up by respective assumptions: correlation-wise stress underlies the idea of systematic crisis affecting all the clusters in the manner relative for each, where percentile tweaking may adjust the rate of which number of losses generated financial institution is willing to mitigate given overall crisis appears.

If the initial data does not provide any insight on possibility of data clustering, unsupervised machine learning algorithms can come in handy while seeking patterns in high dimensional data. From perspective of most used algorithms in financial plane, K-Means Clustering algorithm will be overviewed. [11]

K-means focuses on determining specific centroids μ_i quantity of which is pre-defined, around which subset samples S_i are formed, variance of which is to be minimized.

$$\text{minimize: } \mathrm{VAR}(\mu_i, N) = \sum_{i=1}^{N} \sum_{s \in S_i} (s - \mu_i)^2 \qquad (2.35)$$

SECTION 3

PRACTICAL ML SOLUTIONS IN CREDIT RISK ASSESSMENT

3.1 Framework overview

The following framework will be estimating loan loss provisioning and unexpected losses based on ICAAP economic perspective for the corporate loan portfolio based on financial statement analysis as predictor variable.

Provided all the weaknesses and advantages of each algorithm were discussed, they can be combined in such logic that each of them is used either in model data-preprocessing and analysis, or output granting given the conditions of usage quality suffice.

Collected samples may include potentially wide range of observations and factors, where such must be filtered to qualify the data, lowering the dimensionality and noise in the dataset. Sampling is performed randomly with 80/20 split for testing and training data, with additional 20% validation proportion on training data while learning weights and parameters.

Pre-algorithmic processing and data analysis is also a valuable data preparation step before incorporating the dataset directly into the model input. Except of data cleaning, outlier filtering and normalization, methodology of which will be defined further on, overall independent factor observation and its internal interaction. Therefore, since data is not an ordered time series, its singular histogram graphical representation may provide a necessary visual insight both in general dataset as well as in each of its subsets in accordance with the risk classification chosen to be modeled. When it comes to data relationship, correlation heat-map provides a significant information on the linear relationship in the dataset.

As it was already mentioned, random forest algorithm is most valuable to determine significant factors that influence the class. While K-NN treats the data locationally in the multidimensional plane, ensembling logic of random forest is more sophisticated in comparison, as the range of possible factor and class relationship widens significantly.

The classifying model itself puts on a serious discussion on topic of the "complexity over accuracy" dilemma. Fairly most recognizable model in the financial credit risk assessment is logistic regression algorithm, as such provides a direct logic, that a probability of a certain risk class is influenced by a latent score, which is latter transformed in a probabilistic representation. Given its assumptions are met and linearly influencing factors are specified correctly, this is the best candidate when it comes to deriving credit risk quantification tool.

However, observing this condition may be pointed out, that such is extremely rarely satisfied. While theoretically a certain migration in credit risk is truly being influenced by a specific factor in deterministic way, the factor itself is also being derived from the environment variables, which are continuously changing over time. Hence while non-linearity may estimate current unobservable environmental parameters influence on the consequent factor impact on the credit risk, it also introduces relevance diminishment as the financial environment progresses on. Overfitting can drastically impact the data output over time, hence its validity of the model should be revised in line with the change in the financial environment, while setting adequate forecasting horizons and parameter retraining. Revisiting the issue, the more complex is the algorithm, the more it is prone to overfit to the current state of the system, hence introductions of additional latent transformations may be tested on out-of-bag samples.

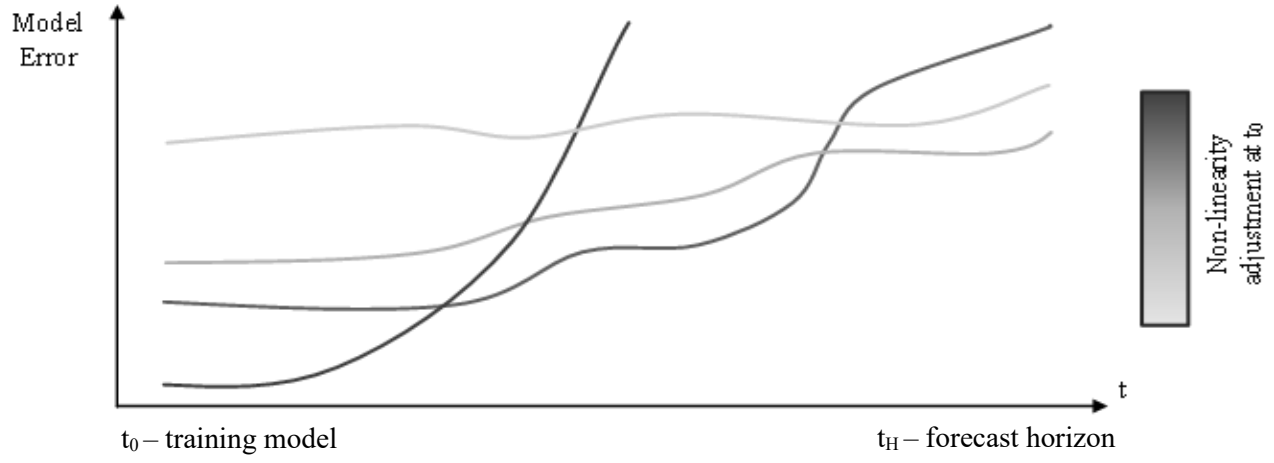


Fig. 3.1. Visual intuition on the overfitting problem.

If probabilistic output is negligible and only specific risk class determination is preferable, then it was studied that SVM shows better result in comparison with thresholding probability output of logistic regression. Hence these two will be used separately in these two scenarios.

Considering mentioned above, exists such optimal model, where its non-linear contribution grants best performance when transposed to the environment of forecasting horizon. While there is no metric that ensures model output compliance with the system progression, a general rule suggests that stable factors which are not overfitted to the training observations reflect best progression over time. Another viable solution is to incorporate certain assumptions either correcting the input data or the output itself based on the sense and firmness of the statement.

As there are only two algorithms specifically relevant for probability of default outputs and do may be loop brut forced without revision (neural network and logistic regression), non-linearity will be added in form of additional hidden layers and neurons based on the metrics provided, opposing to linear model, which is logistic regression itself as a basis algorithm.

Metrics

A valuable consideration when choosing metrics for validating quality of the model lies in the data properties itself (while being normalized, cleaned of outliers and factors with low predicative potential). Risk-shifts distinguishing finds most challenge due to its low frequency and possible unexpectedness, where risk can be severely impacted by the environment itself. Hence metric should correctly interpret quality on imbalanced output data.

Assuming output of the model comes in form of class probability, we can use ROC-curve as basis to derive own-defined metric. ROC curve evaluates relationship between specificity and false positive rate classifications per class, derived as two of four quadrants of confusion matrix. [12]

$$\begin{pmatrix} TP(t) & FP(t) \\ FN(t) & TN(t) \end{pmatrix} = \frac{1}{N}\begin{pmatrix} \sum_{P_i \in P} [P_i > t \cup C_i = 1] & \sum_{P_i \in P} [P_i > t \cup C_i = 0] \\ \sum_{P_i \in P} [P_i \leq t \cup C_i = 1] & \sum_{P_i \in P} [P_i \leq t \cup C_i = 0] \end{pmatrix} \quad (3.1)$$

The most common metric comes area under such parametric curve, interpreted as cumulative added value of true positives when false positive changes by differential of threshold difference (typically in range of unique probability outputs).

$$TPR(t) = \frac{TP(t)}{TP(t) + FN(t)} \quad (3.2)$$

$$FPR(t) = \frac{FP(t)}{TN(t) + FP(t)} \quad (3.3)$$

$$AUC = \int_0^1 TPR(t) \times FPR'(t) \times dt \quad (3.4)$$

Hence metric assessing such uniformly per each class is: [12]

$$M_1 = \frac{1}{C_n} \sum_{C_j \in C} \sum_{t \in \{P_i\}_{i \in \{1,\dots n\}}} TPR_{C_j}\left(FPR_{C_j}^{-1}(t)\right) \quad (3.5)$$

Valuable consideration will be also passed to the best model cross-entropy loss on the testing sample, such given as follows: [12]

$$M_2 = \sum_{i=1}^{N} y_i \times \ln\big(p(C = i)\big) \quad (3.6)$$

Once performance was reviewed and model is selected using mentioned above logic, further statistical analysis is applied to review internal dependencies. Since data is to be categorized preliminary in the dataset, usage of unsupervised clustering is not necessary.

Additionally, once probabilities were estimated, various stress-scenarios concerning unexpected loss computation will be estimated with more in depth specifics described further, having correlation effects, probability severity, etc.

Overall, the summarization may be accessed in annex 5.

3.2 Probability of default estimation

In this research, various financial statements, indicators, and financial performance measures were taken as factors to determine probability of default for each observation of arbitrary portfolio that hence is subject to be simulated on the deviation of losses [23]. These deviations will be hence used to determined future possible unexpected losses under ICAAP economic perspective.

As data was imported, data was pre-processed and cleaned, raw data descriptive statistics and cleaned data distributions may be revised in first and second annexes respectively.

Overall, dataset contains 2409 observation, each having 72 predictive variables. Clean data provides some measure of correlation within the data, however most of them prove to be uncorrelated, yet the factors themselves are to be revised via random forest variable importance algorithm.

Fig. 3.2: Correlation matrix heat-plot on clean data.

Afterwards as mentioned, this algorithm provided next results, where important variables were selected 10 most significant on overall rescaled results using min-max normalization with 1000 trees to be trained (names omitted due to unreadability, full table may be accessed in annex 3), which results yielded next selection of variables:

As there are a big quantity of predictor variables, these will be averaged according to this categorization:

- Profitability
- Liquidity
- Solvency
- Funding
- Turnover
- Valuation
- Other

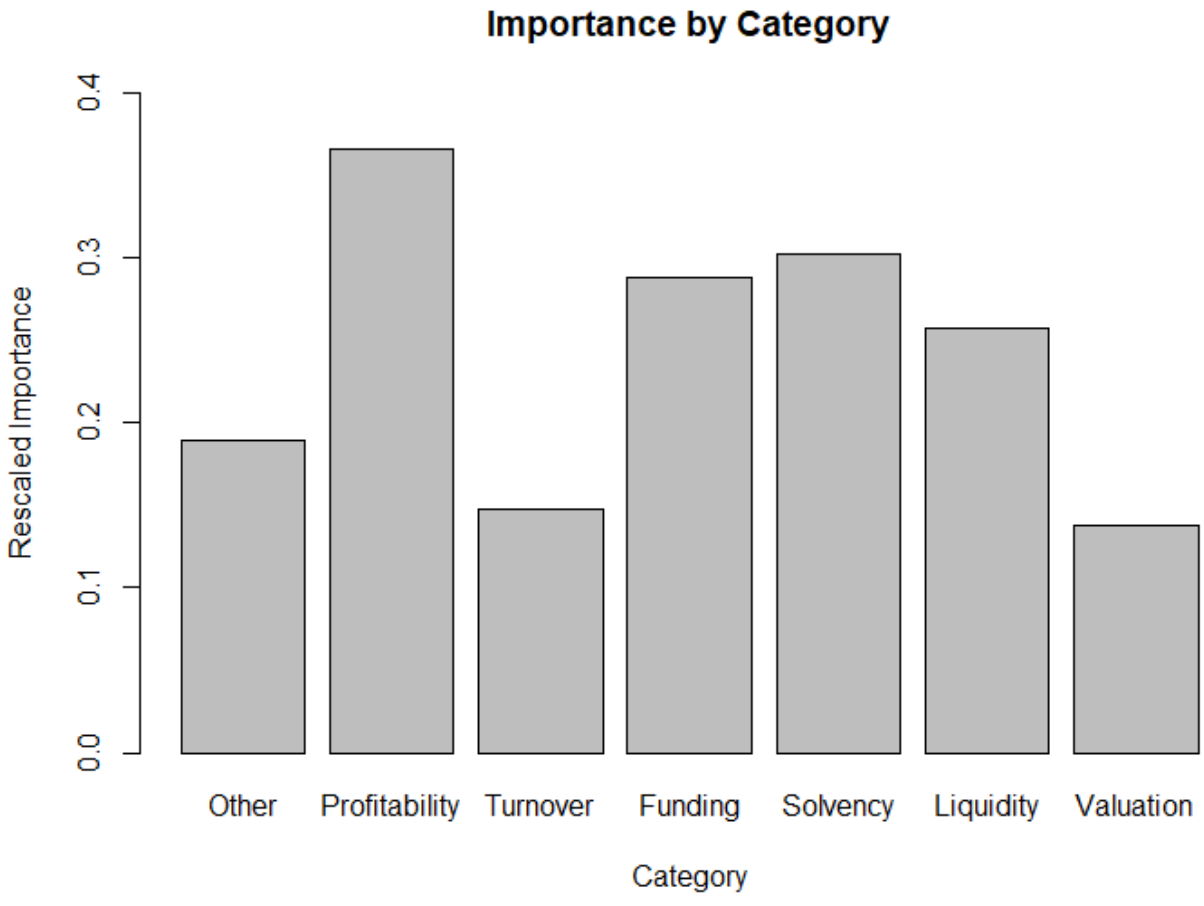


Fig. 3.3: Variable importance averaged by financial category.

As it is visible, variable with most efficient value in predicting defaults along with solvency in the next place, which corresponds to general logic of credit risk.

These were found most significant drivers of credit risk, which mostly focus on profitability of the business client.

This algorithm hence resulted in selection of next predictive variables for neural network:

1. "Profit after tax/Total assets"
2. "PBDITA/Total assets"
3. "PBT/Total assets"
4. "Cash/profit"
5. "Cash profit/Total assets"
6. "PAT/Net Worth"
7. "Reserves.and.funds.Total_Assets"
8. "Shareholders funds/Total assets"
9. "Quick ratio"
10. "Debt to equity ratio times"

All of these are tightly associated with capability of business to accumulate profits, remaining solvent while having lending counterparty exposed to lower proportions credit risk, with some accounting for current short-term liquidity needs and overall debt pressure with potential funding from equity holders.

Next step is selecting optimal neural network structure. As currently no specific optimal method for such is available, structures were brut forced based on the metrics discussed in the framework description, ranging from simple logistic regression up two feed-forward neural network of three layers ten nodes each.

The learning process converged to a neural network with two hidden layers with 20 neurons each and 10% dropout chance for each neuron (see Annex 2) based on binary cross-entropy loss, and this model will be opposed to simple linear regression to see the non-linearity impact.

As the network was trained, the loss function development was revised on check whether values converge similarly on validation and training samples. Observing plots below, it is visible that progress had a misconception, after which it has appreciated matching the training loss.

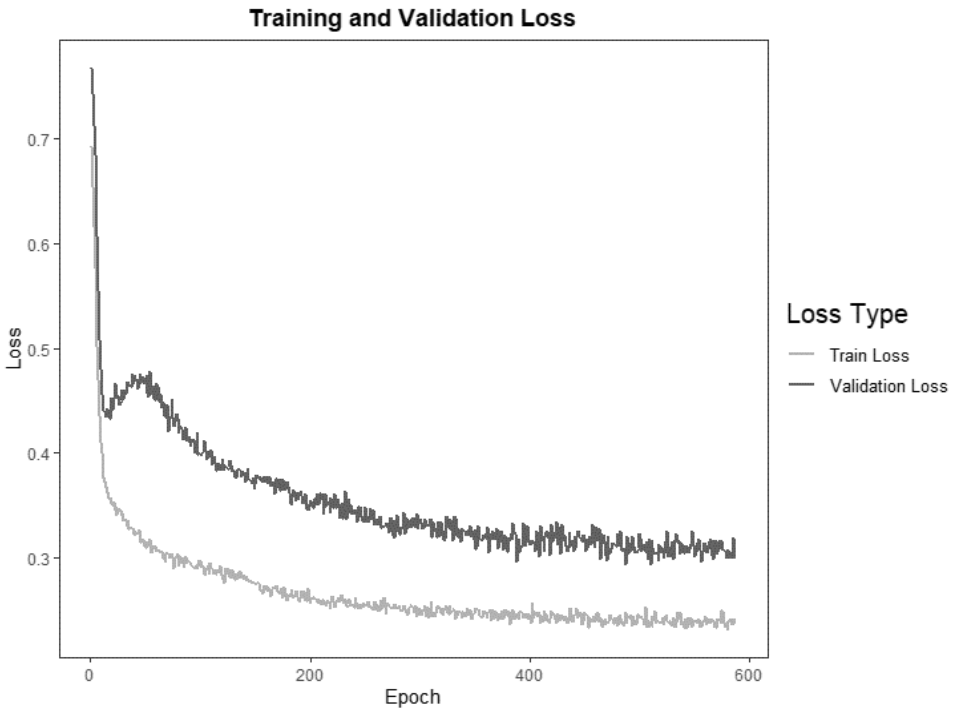


Fig. 3.4: Loss function during parameters learning.

From the analysis of the outputs, it seems the non-linearity effect has a significant improvement on specifying patterns which are relevant for riskier clients.

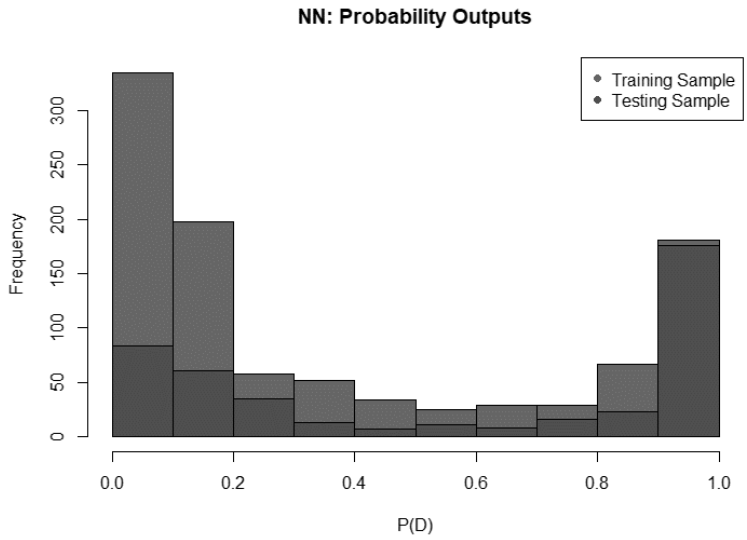


Fig. 3.5: Neural network outputs on training and testing sets.

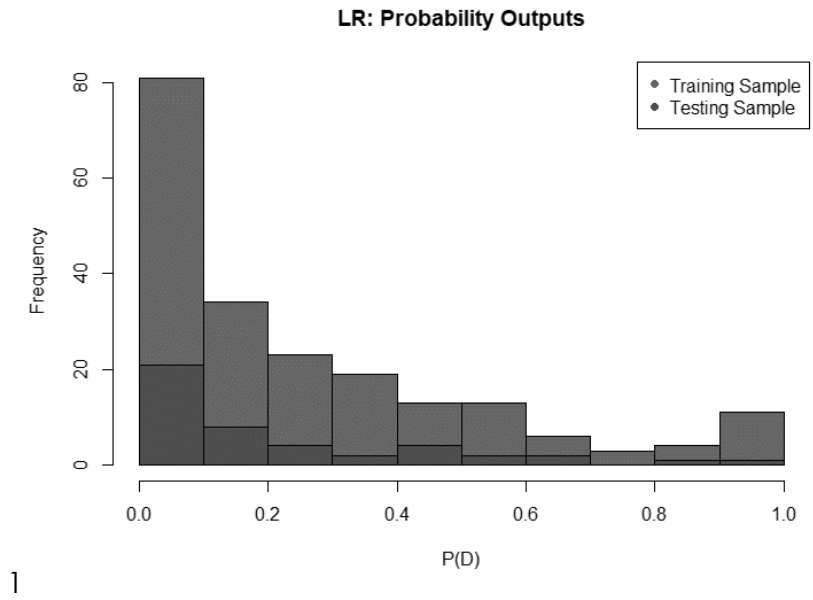


Fig. 3.6: Logistic regression outputs on training and testing sets.

Afterwards, revision of the confusion matrices on testing set is performed. Conclusion may be stated that neural network seems to treat default evens more critically, which gives a privilege in terms of risk management usage, of course with a certain trade-off in precision.

Neural Network	Non-default observed	Default observed
Non-default predicted	0.8849206	0.003968254
Default predicted	0.0952381	0.015873016

Logistic Regression	Non-default observed	Default observed
Non-default predicted	0.95238095	0.007936508
Default predicted	0.02777778	0.011904762

Table 3.1. Confusion matrix comparison tracing non-linearity

Additional metric to be revised will be ROC-curve and AUC metric again on the testing sample. From visual test and the area output, it may be stated that neural network overall, except the risk-conservancy, suggests a better trade-off in precision and recall for indeterminant threshold.

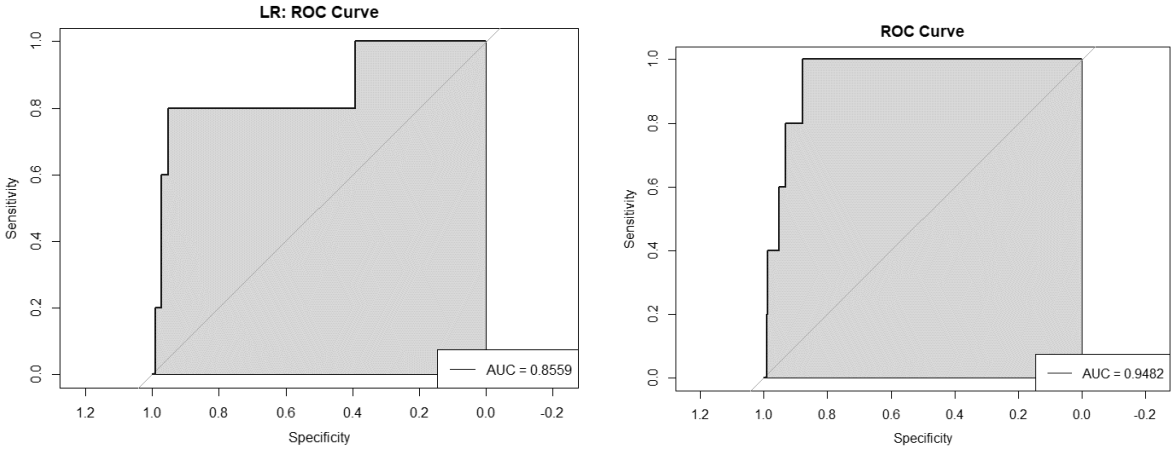


Fig 3.7. ROC curve comparison between algorithms.

These comparisons suggest sufficient evidence that usage of neural network is more beneficiary, introducing a bigger likelihood to determine more risky business clients, while also having ROC-curve supremacy as additional degree of insurance towards such choice.

3.3 Loss distribution and unexpected losses

In this step Monte-Carlo process was launched to calculate loss deviations based on outputs granted by neural network.

Referring to formula (1.1), expected losses and hence allowances are calculated and then weighted by the portfolio volume which granted results of 13.38% (gross LP: 1,932,850,284 arbitrary currency units) as a proportion expected to be lost, which a very significant ratio to start with.

Loss given default parameter was set to 0.7 uniformly, the population distribution of exposures may be revised below (simulated from multiple normal distributions to create multimodal realistic sample of exposures):

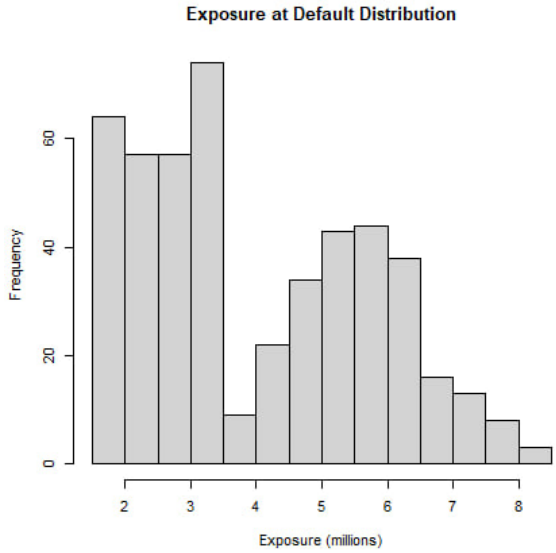


Fig 3.8. EAD distribution. [generated by the author]

For this practical research, let various presented by adversity stress tests are proposed, all losses conducted subject to survival rate 0.99 and 100 as an accepted margin of error:

- Stress scenario 1 – unexpected losses based on the simulation.
- Stress scenario 2 – probabilities of default severity drifting of 1%.
- Stress scenario 3 – segment-wise correlated losses.
- Stress scenario 4 – previous two combined stress.

For scenarios 3 and 4, clustering will be performed as given in annex 7 using k-means algorithm already reviewed in the section 2.3, limiting cluster number to five as multimodal distribution had this much of simulating independent gaussians. The parameters of clustering will be revised as follows based on the conducted random forest important variables concluded in section 3.2. Once clustered, such can be reviewed in terms of PD estimates.

Algorithm has defined that out of all possible simulations four clusters were determined as most representative using elbow point method, PD estimated of which really do form different patterns in its severity:

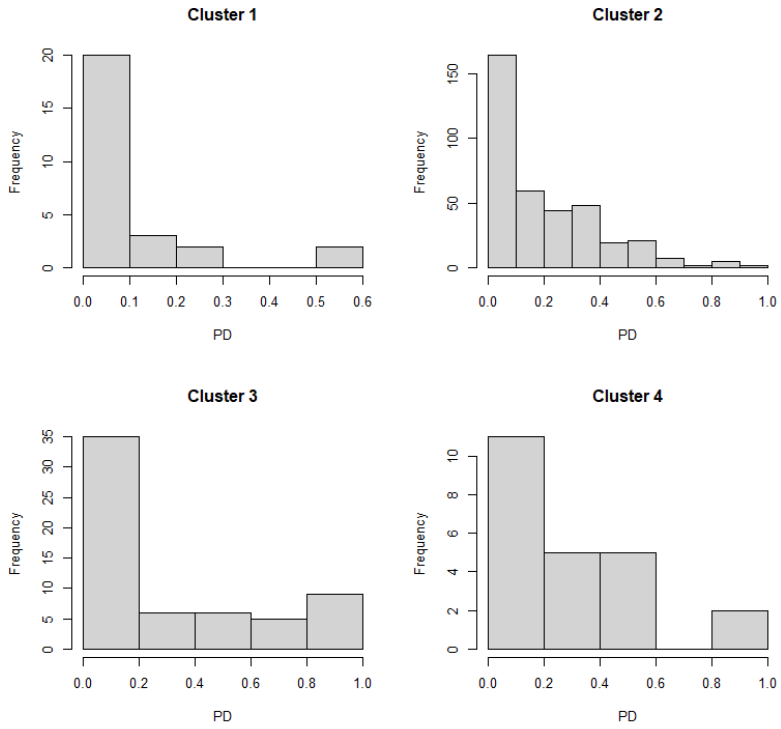


Fig. 3.9. Clustering results: PD overview.

The metric of within cluster sum of squares per iteration from which optimal clustering was chosen is represented below:

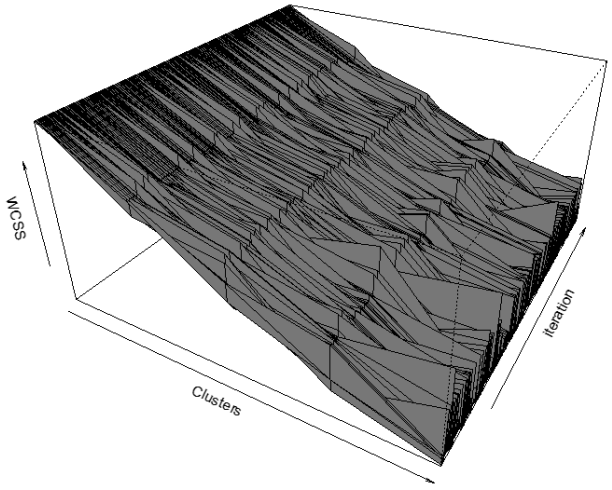


Fig 3.10. Cluster simulation process.

Given this step was fulfilled, further analysis is being conducted by performing Monte-Carlo simulations:

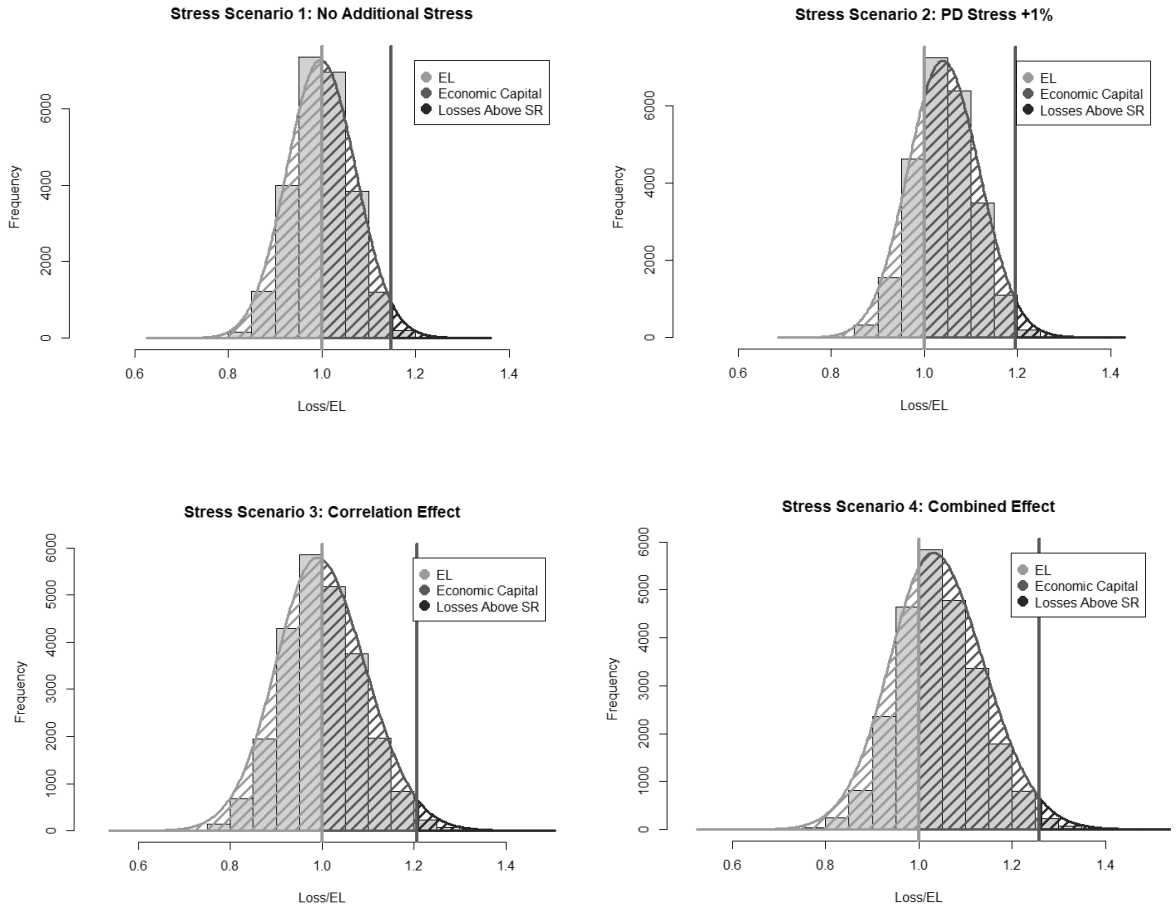


Fig 3.10. Stress-testing results.

What can be concluded and hence used is that these various stresses differently affect the loss distribution. Correlation effect affects the non-linearly distribution when

going closer to the tails, while effect vanishes gradually around the mean, introducing a **variance effect** on the distribution. Such means that severity of the outcomes is much more serious as values get more extreme since the behavior is duplicated cluster-wise. The distribution generally increases instability in cases of losses, which is relevant when crisis occurs. On the contrary, probability shift is a **mean effect** on the losses, linearly shifting the loss, indicating that system has already a bias towards greater losses, opposing to mentioned before.

The results for each scenario are given respectively: 17.24%, 19.79%, 20.12% and 23.4% of additional uncovered losses, relative to the volume of allowances formed for gross LP presented earlier.

Concluding point suggests that correlation implication is a must have step to define clusters that are significantly different, yet their loss may be much more severe once facing a credit risk driver affecting all clusters in a similar manner, being a systematic macroeconomic, political, or country risk, yet if behavior around expected losses is preserved as in general healthy economy.

However, if the financial entity already has information on systematic severity included in the historical data used for estimating probabilities of default, then such stress is already set up inside the parameters, and the variance is significantly inflated. This implies that a systematic factor has already influenced the credit risk through empirical loss occurrences, yet at the same time given there is no ground evidence of situation to become favorable, financial entities may consider worsening PDs by some equivalent proportion as a systematic risk would arguably affect the state of the business.

At the same time there is another approach to stress-testing, which involves factor of PD parameter stress, which is then extrapolated on the credit risk. However, this provides very volatile behavior given simulation is not supervised by concrete logic, which is hard to be implemented in black-box models, yet may be indeed achievable, requiring a consistent roadmap of how shock logic is built in the simulation. To prove this point, randomly simulating shocks can provide unrealistic scenarios, where liquidity/solvency generates a very poor outcome at the same time

with remarkable profit increase, which provides discrepancies and likely data which was never closely observed by the network. Such results are volatile since great non-linearity is implemented, this likely would not have been the case for simpler and more interpretable models (e.g. logistic regression), so insurance of logic preservation is crucial. For reference a shock on average will be produced, where each variable will be shocked by drawing arbitrary shock from normal distribution, parametrized with respective in-cluster deviation.

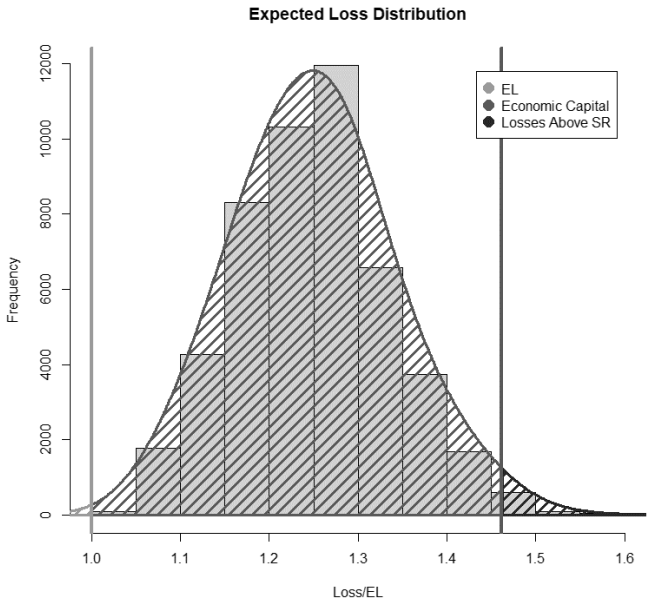


Fig 3.11. Volatile representation of the factor simulation.

As assumed, such resulted in significant losses, deviating far past all the other simulations which are more reliable representation due to its control past the neural network domain. As stated earlier, this is due to simulation of illogical scenarios, creating severe outputs, solution of which comes from setting up concrete rules to omit situations where scenarios contain inconsistent results in the factor generation.

Conclusions

Quantitative assessment is essential part of credit risk management, and as techniques evolve past regression models and econometrics, it introduces necessity to provide sufficient research before implementing such as a part of credit risk modelling instruments. Current one provides sufficient evidence of the neural network outperformance towards more traditional approaches recognized both by the regulatory institutions or financial consulting and audit companies (logistic regression) along with usage of random forest variable importance algorithm. This implies much more precise estimates of default probabilities, and hence in researched context better loan loss provisioning estimates and higher quality parameters for simulation performance. Additionally, valuable insights were found on how specifically various stress testing assumptions influence the output distributions and when consequently they could be use when applying Monte-Carlo simulation to estimate unexpected losses in addition to solving what additional economic capital along with reserves should be formed by financial entity.

However, possible critics are sufficiently justified towards conclusions of the neural network usage in a way that tests were conducted only on one sample for a specific credit risk modelling problem, which involved many various financial factors affecting corporate clients. This means that feedback may be not as univariate when having different portfolio properties, so a disclaimer here must be foreseen. Another challenging argument is the sample size, which for more ground conclusions should involve a much bigger historical input to ensure all the experience of financial stress was incorporated.

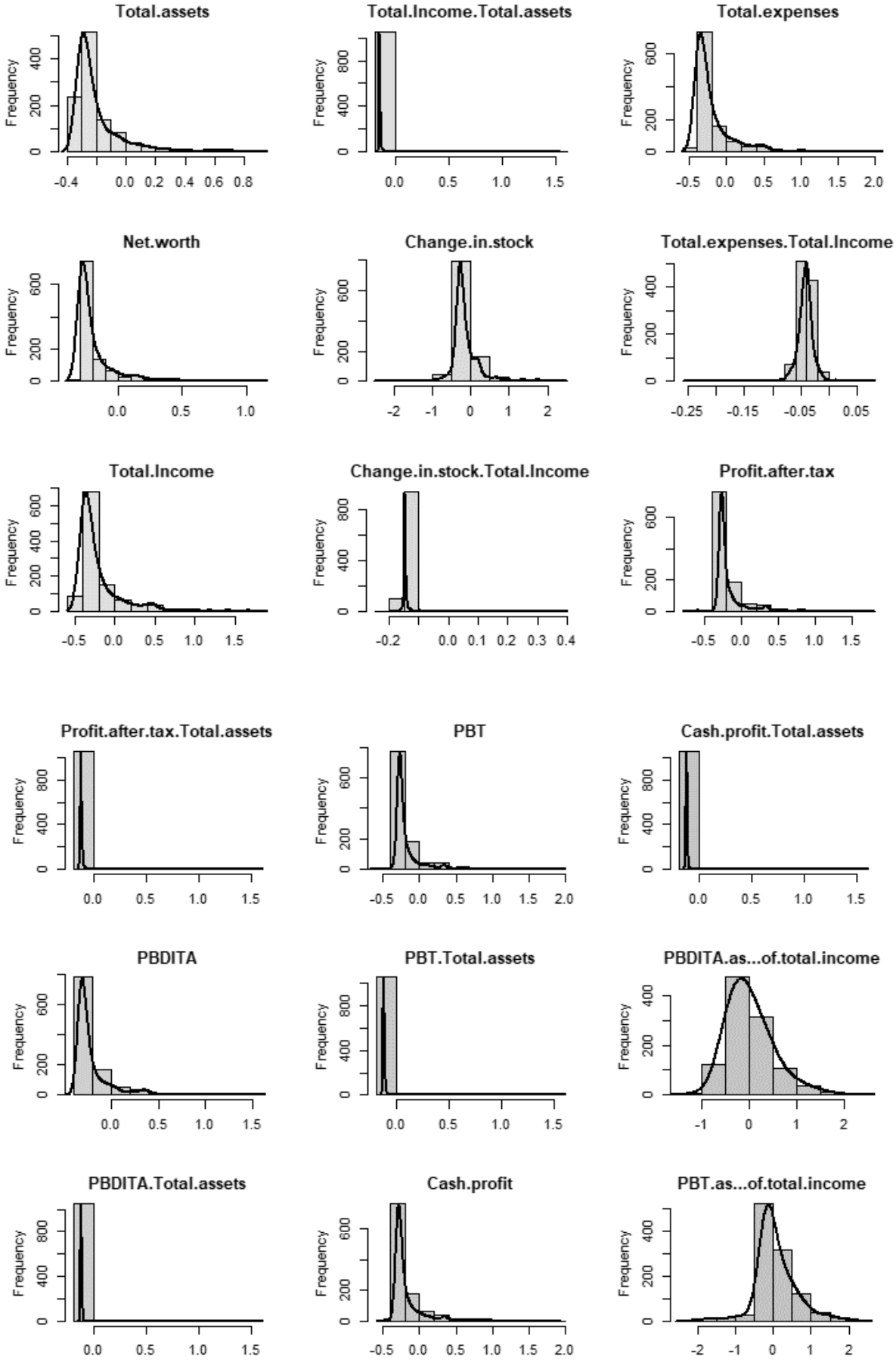
At the same time, having overviewed the algorithms, theoretical basis suggests that proper neural network structure using various techniques as drop-out layers, different hidden layer architectures, other techniques for data pre-processing such as convolution and pooling provide a much more enhanced opportunities to fit a specific problem to the domain of its solution. Therefore, neural networks both theoretically and practically can provide a much higher quality estimates than traditional algorithms in case high non-linearity is demanded to reach the solution.

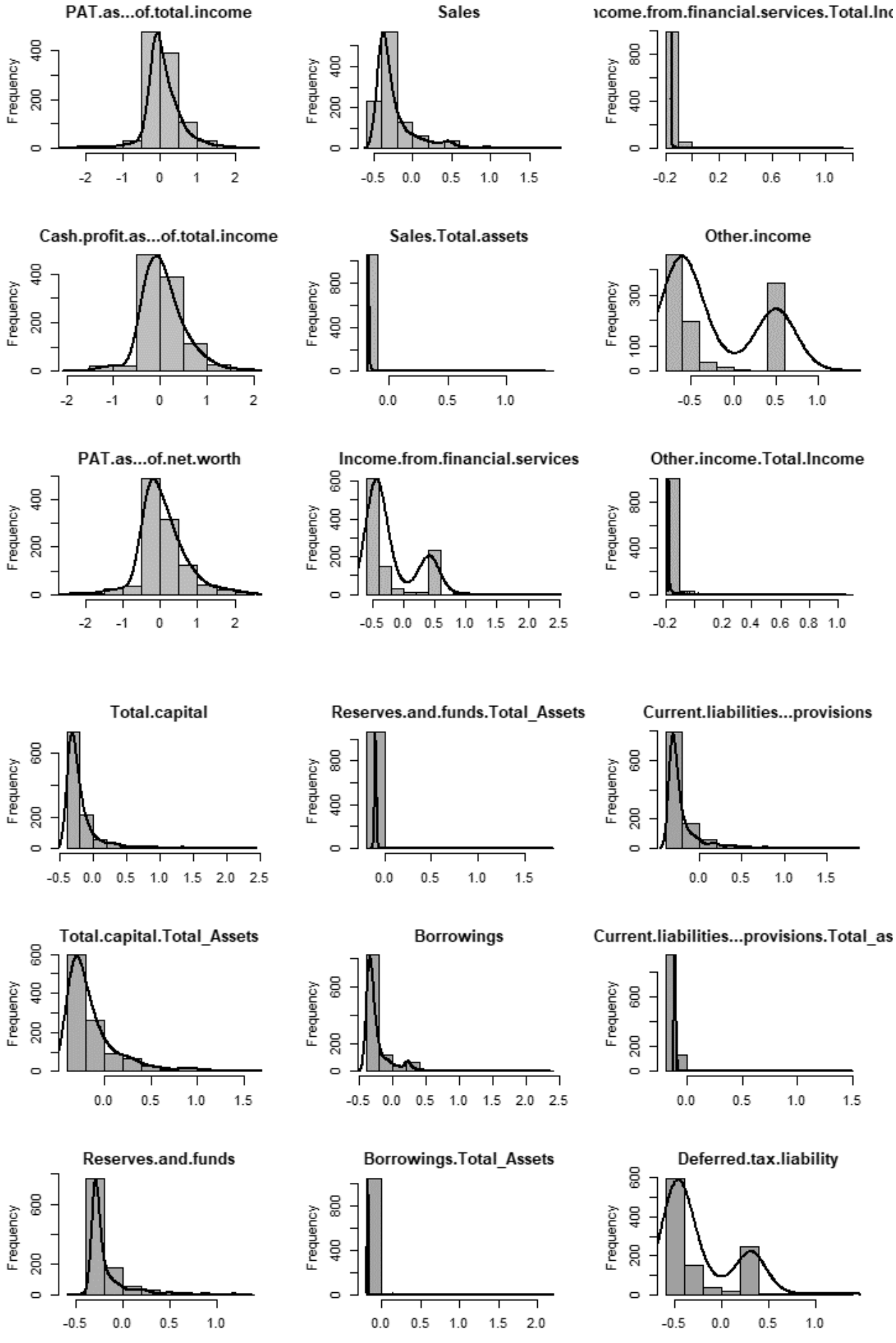
In conclusion, neural networks as well as provide much promise in their usage in credit risk modelling, while the basis of research on these topics only grows through time. Financial entities may adapt these solutions to overviewed credit risk problems yet considering sufficient demand over high quality model validation with sophisticated background to reason chosen approaches both from the perspective of common sense as well as optimizing for benchmarking metrics of the model performance.

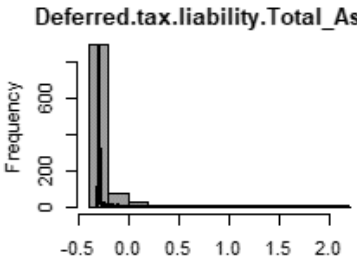
Annex 1: Descriptive statistics of the dataset

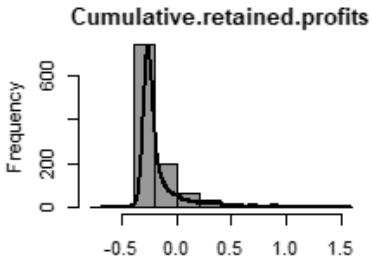
Statistical summary	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Total.assets	2409	2003.926	6285.073	319.3	657.538777	408.7528	4.2	54287.5	54283.3	6.176064	42.829089	128.0536
Net.worth	2409	725.2378	2397.772	103.2	226.934318	132.9892	0.5	21290.98	21290.48	6.625745	49.228331	48.85281
Total.Income	2409	2176.605	5312.328	505	1050.69207	689.409	0.458	42941.21	42940.75	5.400491	34.114446	108.2347
Total.Income.Total.assets	2409	17.81732	109.648	1.1350461	1.24785115	0.724371	0.000426	1107.327	1107.327	8.163974	70.355123	2.233996
Change.in.stock	2409	24.59145	89.40276	4.3	13.0164893	16.3086	-237.194	607.53	844.724	3.609448	21.096172	1.821514
Change.in.stock.Total.Income	2409	1.4369	9.742557	0.009008	0.01559262	0.024105	-6.85714	91.47163	98.32877	8.172273	68.794884	0.198497
Total.expenses	2409	1977.089	4908.395	445.2	909.780948	613.4999	0.2	38558.23	38558.03	5.315254	32.712765	100.0049
Total.expenses.Total.Income	2409	1.175647	4.792292	0.9765396	0.97123006	0.061184	4.30E-05	214.4	214.4	38.47282	1648.7886	0.097639
Profit.after.tax	2409	128.8538	464.689	9.9	39.3754835	17.0499	-190.688	4239.678	4430.366	6.810046	52.96743	9.46769
Profit.after.tax.Total.assets	2409	0.742269	5.757217	0.0311526	0.04023694	0.045187	-4.14286	67.08359	71.22645	9.754625	99.992624	0.117299
PBDITA	2409	289.5571	881.5312	40.9	104.829775	58.11792	-23.11	7445.624	7468.734	6.10693	42.146972	17.96053
PBDITA.Total.assets	2409	1.587637	11.98906	0.1153239	0.11906335	0.07591	-4.14286	139.8027	143.9456	9.766913	100.16998	0.244268
PBT	2409	176.2913	626.2419	13.9	54.4643419	22.83204	-195.412	5653.056	5848.468	6.746339	51.587714	12.75921
PBT.Total.assets	2409	1.029757	7.964937	0.0449695	0.05630171	0.062463	-4.14286	92.82899	96.97185	9.760387	100.0722	0.16228
Cash.profit	2409	190.1973	626.4959	21.4	64.885243	31.72764	-108.666	5637.986	5746.652	6.647915	50.400068	12.76438
Cash.profit.Total.assets	2409	1.063931	8.1335	0.0692661	0.07350128	0.061719	-4.14286	94.82335	98.96621	9.765261	100.14398	0.165714
PBDITA.as...of.total.income	2409	10.7346	14.90623	9.57	10.4348582	7.605738	-61.2704	54.8008	116.0712	-1.06508	8.2072729	0.303703
PBT.as...of.total.income	2409	3.893856	12.8015	3.28	4.28883878	4.862928	-31.9608	51.3564	83.3172	-0.05001	3.5397936	0.260821
PAT.as...of.total.income	2409	2.422543	10.93942	2.31	3.02648004	3.424806	-28.7076	42.0744	70.782	-0.2904	3.6674845	0.222882
Cash.profit.as...of.total.income	2409	5.817123	13.48239	5.56	6.21563504	6.07866	-49.9664	51.5132	101.4796	-1.07457	7.2673443	0.274694
PAT.as...of.net.worth	2409	9.83074	26.41655	8.01	10.2804355	11.87563	-90.2508	97.137	187.3878	-0.73527	5.1646013	0.538217
Sales	2409	2205.133	5162.804	526.7	1131.27501	718.3197	0.6	41549.13	41548.53	5.327366	33.454517	105.1883
Sales.Total.assets	2409	22.43041	121.4548	1.1429924	1.28180081	0.734077	0.000627	1096.196	1096.196	7.245493	55.504872	2.47455
Income.from.financial.services	2409	42.481	92.47521	5.5	27.3400428	8.00604	0	778.776	778.776	5.63706	38.634299	1.884113
Income.from.financial.services.Total.Income	2409	2.801011	18.18968	0.0083821	0.07631333	0.011611	0	177.6031	177.6031	8.440762	73.621266	0.370601
Other.income	2409	23.59917	35.89573	7.5	19.0216859	10.97124	0	290.886	290.886	4.328499	26.78297	0.731349
Other.income.Total.Income	2409	1.726949	9.792191	0.0076765	0.07723165	0.010537	0	91.0347	91.0347	7.680975	61.769774	0.199509
Total.capital	2409	144.0897	377.9475	42.6	61.3990546	49.22232	0.5	2975.118	2974.618	5.541471	34.441611	7.700396
Total.capital.Total_Assets	2409	0.26296	0.66073	0.1154313	0.15889526	0.122056	0.000297	15.71412	15.71382	13.94467	274.92626	0.013462
Reserves.and.funds	2409	616.5941	2028.539	59.5	191.845509	96.81378	-228.18	17692.64	17920.82	6.308357	45.221094	41.32996
Reserves.and.funds.Total_Assets	2409	2.253171	20.98859	0.2050772	0.22286022	0.228438	-14.3261	282.4411	296.7672	11.77587	144.69343	0.427627
Borrowings	2409	706.5354	1910.227	126.8	306.811374	177.1707	0.1	15257.11	15257.01	5.632405	35.881304	38.91945
Borrowings.Total_Assets	2409	5.597716	30.01849	0.3389328	0.3513161	0.243269	0.000148	270.4954	270.4953	7.321082	56.799715	0.611604
Current.liabilities...provisions	2409	472.4819	1396.979	74.2	170.174058	100.8168	0.1	11422.88	11422.78	5.817996	38.045678	28.46239
Current.liabilities...provisions.Total_assets	2409	2.18417	17.65586	0.2138611	0.23168136	0.159526	0.000833	227.4373	227.4365	10.91299	126.13533	0.359725
Deferred.tax.liability	2409	141.3165	273.1294	44.8	100.495796	65.08614	0.1	2329.492	2329.392	5.719236	39.624754	5.564806
Deferred.tax.liability.Total_Assets	2409	2.218659	7.536574	0.0503	0.42135931	0.057377	0	54.3206	54.3206	5.221732	29.566827	0.153552
Shareholders.funds	2409	587.7235	1352.148	105.6	235.800518	136.1027	0.5	7678.624	7678.124	3.756571	14.800103	27.54899
Shareholders.funds.Total_assets	2409	0.411836	0.231838	0.3667588	0.39360111	0.223781	0.004377	1	0.995623	0.644248	-0.2441386	0.004724
Cumulative.retained.profits	2409	438.5677	1555.111	39.3	125.504364	70.4235	-498.696	14292.71	14791.41	6.707551	51.82714	31.68422
Cumulative.retained.profits.Total.Income	2409	0.192612	41.95372	0.1077543	0.14093272	0.160235	-371.4	1984.372	2355.772	43.53579	2076.7143	0.854776
Capital.employed	2409	1120.052	2589.092	224.4	451.286366	285.2522	2.658	14541.12	14538.46	3.790389	14.877804	52.75081
Capital.employed.Total.assets	2409	0.706766	0.17993	0.7330037	0.72201821	0.168761	0.021784	1	0.978216	-0.79742	0.5420494	0.003666
TOL.TNW	2409	3.16368	7.038054	1.45	1.75284085	1.4826	0	54.839	54.839	5.658101	35.411015	0.143395
Total.term.liabilities...tangible.net.worth	2409	1.206792	3.369405	0.35	0.53925868	0.51891	0	27.3566	27.3566	6.252709	42.905139	0.068649
Contingent.liabilities...Net.worth....	2409	36.04985	94.638	5.42	15.2019181	8.035692	0	756.4454	756.4454	5.496266	35.096281	1.928178
Contingent.liabilities	2409	529.576	816.9075	182.1	411.363648	268.3506	0.1	6230.28	6230.18	4.078528	23.248401	16.64388
Contingent.liabilities.Total.Assets	2409	9.915309	30.71641	0.1421602	2.47337128	0.197524	6.45E-05	222.97	222.9699	5.000733	27.781497	0.625824
Net.fixed.assets	2409	670.5572	2098.892	101.5	232.692536	135.0649	0	17098.72	17098.72	6.042367	40.278649	42.76334
Net.fixed.assets.Total.Assets	2409	3.672843	27.53844	0.322328	0.33482304	0.216708	0	287.4348	287.4348	9.424433	90.880463	0.561075
Investments	2409	417.3314	652.8862	153.9	333.720538	228.0239	0	5078.45	5078.45	4.420278	26.622729	13.30207
Investments.Total.Income	2409	21.65869	141.1762	0.1507306	0.99312146	0.222019	0	1530.598	1530.598	9.231533	89.852032	2.87636
Current.assets	2409	768.9078	2108.907	150.3	304.249829	202.0784	0.1	17899.09	17898.99	5.676719	37.306505	42.9674
Current.assets.Total_Assets	2409	1.908562	17.40683	0.4466101	0.4490853	0.234014	0.000696	313.7851	313.7845	14.44528	224.19947	0.354651
Net.working.capital	2409	112.5616	552.2414	17.6	44.2507056	48.48102	-1745.76	3754.11	5499.874	3.650746	23.584052	11.2515
Net.working.capital.Total.Capital	2409	2.895982	18.53484	0.5598291	0.96257045	1.231405	-92.4	322.5208	414.9208	10.61017	147.27918	0.377634
Quick.ratio..times.	2409	1.066165	1.688265	0.68	0.75401976	0.474432	0	13.7216	13.7216	5.550827	35.677439	0.034397
Current.ratio..times.	2409	1.743051	2.220313	1.24	1.35350946	0.563388	0.01	18.5622	18.5522	5.57675	36.332756	0.045237
Debt.to.equity.ratio..times.	2409	1.887745	4.311201	0.8	1.01630378	1.022994	0	34.694	34.694	5.686216	36.638121	0.087837
Cash.to.current.liabilities..times.	2409	0.300054	0.753436	0.07	0.12712583	0.088956	0	5.4234	5.4234	4.989996	27.648239	0.015351
Cash.to.average.cost.of.sales.per.day	2409	51.36765	161.8392	8.66	16.5604044	10.27442	0	1277.5	1277.5	5.998273	39.31829	3.297351
Creditors.turnover	2409	12.61624	19.35384	6.8	8.59999548	5.767314	0	131.4468	131.4468	4.189645	19.974647	0.39432
Debtors.turnover	2409	13.72802	25.07508	7.08	8.8070876	6.15279	0	191.2038	191.2038	5.199424	30.438049	0.510886
Finished.goods.turnover	2409	61.79011	118.2198	25.65	39.9469953	29.74096	0.8958	867.8504	866.9546	5.11095	29.156653	2.408639
WIP.turnover	2409	22.36031	31.40382	13.1	15.9538524	13.52131	0.25	223.8926	223.6426	4.121279	20.820564	0.639829
Raw.material.turnover	2409	11.2871	13.98821	7.19	8.91213878	7.04235	0	98.7758	98.7758	3.600489	17.461269	0.284999
Shares.outstanding	2409	17480159	36790657	6050000	9970589.83	8232774	19600	2.85E+08	2.85E+08	5.13244	30.229539	749582
Equity.face.value	2409	-244.374	540.7848	10	-148.79321	0	-1348.44	163	1511.445	-1.54444	0.4040376	11.01808
EPS	2409	20.09334	66.39023	1.5	5.17780715	3.054156	-13.7332	424.2444	437.9776	4.93527	25.334042	1.352651
Adjusted.EPS	2409	18.54535	62.87059	1.28	4.30638673	2.564898	-13.9632	394.2376	408.2008	4.862273	24.371183	1.280941
Total.liabilities	2409	2003.926	6285.073	319.3	657.538777	408.7528	4.2	54287.5	54283.3	6.176064	42.829089	128.0536
PE.on.BSE	2409	45.54078	31.46938	64.078012	48.5880496	0	-55.5462	158.462	214.0082	-0.4007	1.2725115	0.641165

Annex 2: Clean Data Visualization

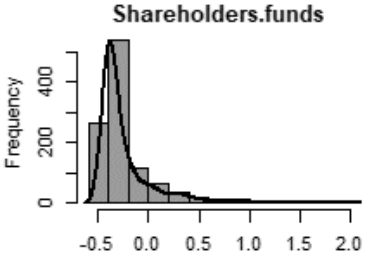


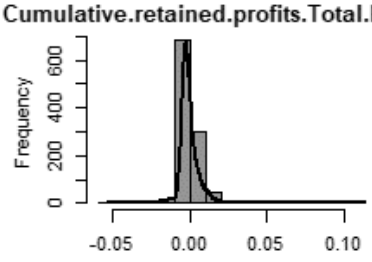

PAT.as...of.total.income
Sales
ıcome.from.financial.services.Total.Inc
Cash.profit.as...of.total.income
Sales.Total.assets
Other.income
PAT.as...of.net.worth
Income.from.financial.services
Other.income.Total.Income
Total.capital
Reserves.and.funds.Total_Assets
Current.liabilities...provisions
Total.capital.Total_Assets
Borrowings
Current.liabilities...provisions.Total_as
Reserves.and.funds
Borrowings.Total_Assets
Deferred.tax.liability
Frequency

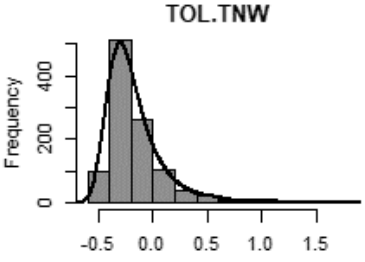

Deferred.tax.liability.Total_Assets
Frequency
0
200
600
-0.5
0.0
0.5
1.0
1.5
2.0

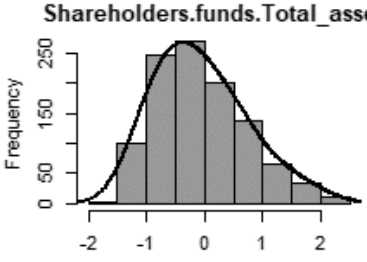

Cumulative.retained.profits
Frequency
0
200
600
-0.5
0.0
0.5
1.0
1.5

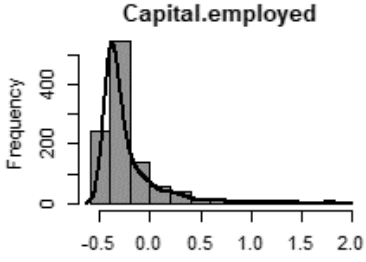

Capital.employed.Total.assets
Frequency
0
50
150
250
-2
-1
0
1
2

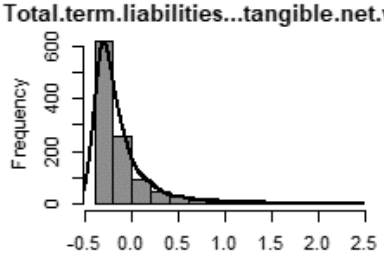

Shareholders.funds
Frequency
0
200
400
-0.5
0.0
0.5
1.0
1.5
2.0

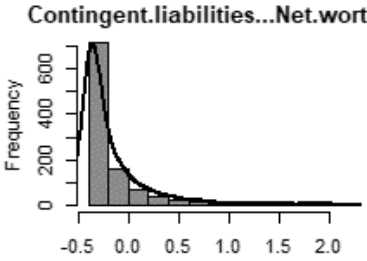

Cumulative.retained.profits.Total.Inco
Frequency
0
200
400
600
-0.05
0.00
0.05
0.10

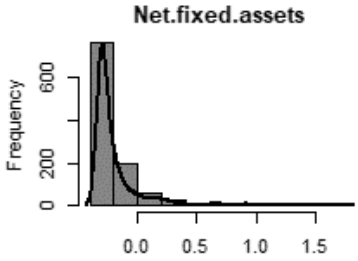

TOL.TNW
Frequency
0
200
400
-0.5
0.0
0.5
1.0
1.5

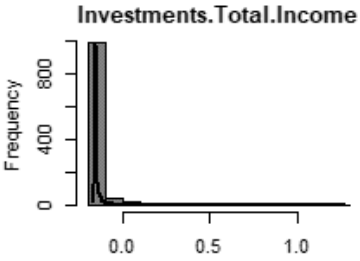

Shareholders.funds.Total_assets
Frequency
0
50
150
250
-2
-1
0
1
2

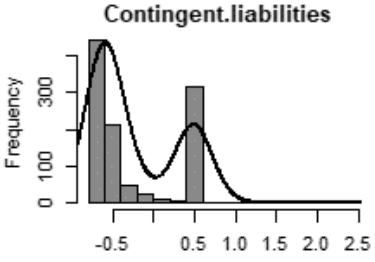

Capital.employed
Frequency
0
200
400
-0.5
0.0
0.5
1.0
1.5
2.0

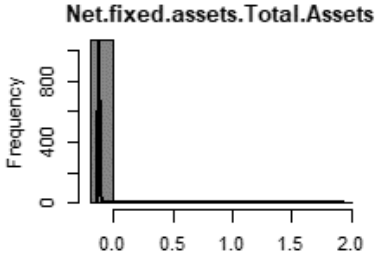

Total.term.liabilities...tangible.net.wo
Frequency
0
200
400
600
-0.5
0.0
0.5
1.0
1.5
2.0
2.5

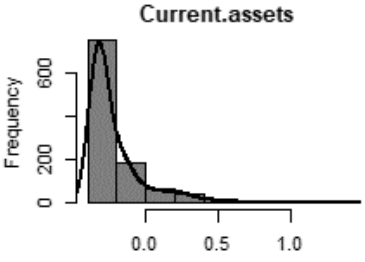

Contingent.liabilities...Net.worth....
Frequency
0
200
400
600
-0.5
0.0
0.5
1.0
1.5
2.0

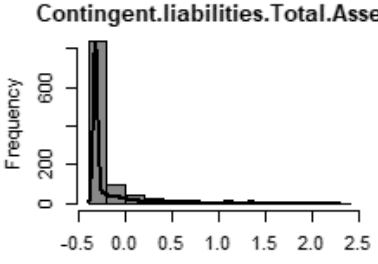

Net.fixed.assets
Frequency
0
200
600
0.0
0.5
1.0
1.5

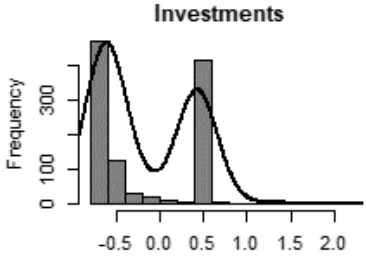

Investments.Total.Income
Frequency
0
400
800
0.0
0.5
1.0

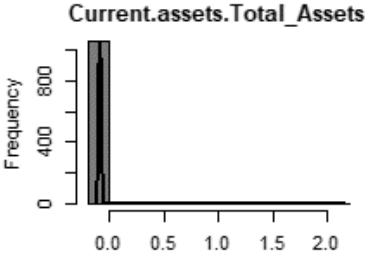

Contingent.liabilities
Frequency
0
100
300
-0.5
0.5
1.0
1.5
2.0
2.5

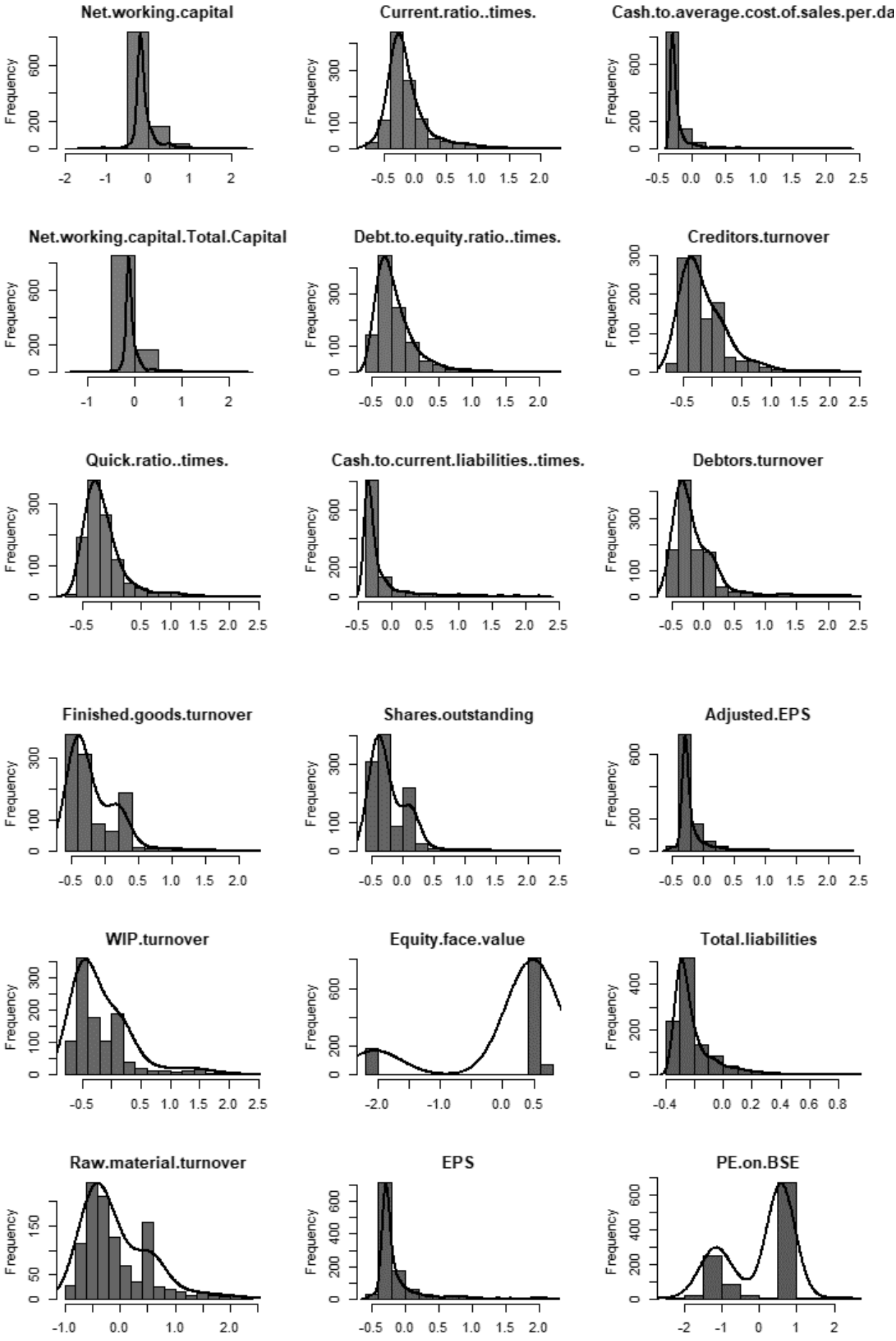

Net.fixed.assets.Total.Assets
Frequency
0
400
800
0.0
0.5
1.0
1.5
2.0


Current.assets
Frequency
0
200
600
0.0
0.5
1.0


Contingent.liabilities.Total.Assets
Frequency
0
200
600
-0.5
0.0
0.5
1.0
1.5
2.0
2.5


Investments
Frequency
0
100
300
-0.5
0.0
0.5
1.0
1.5
2.0


Current.assets.Total_Assets
Frequency
0
400
800
0.0
0.5
1.0
1.5
2.0


Net.working.capital
Current.ratio..times.
Cash.to.average.cost.of.sales.per.da
Net.working.capital.Total.Capital
Debt.to.equity.ratio..times.
Creditors.turnover
Quick.ratio..times.
Cash.to.current.liabilities..times.
Debtors.turnover
Finished.goods.turnover
Shares.outstanding
Adjusted.EPS
WIP.turnover
Equity.face.value
Total.liabilities
Raw.material.turnover
EPS
PE.on.BSE
Frequency

Annex 3: Pre-processing

```
dataset <- read.csv('corporate_credit_risk.csv')

Y <- dataset[,3]
X <- dataset[,-c(1:3,ncol(dataset))]

#normalization
X_scaled <- scale(X)

scales <- attr(X_scaled,"scaled:scale")
centers <- attr(X_scaled,"scaled:center")

#split to historical input and evaluating input for the purpose of research
historical <- sort(sample(seq(nrow(X_scaled)),nrow(X_scaled)*0.8))
X_input <- X_scaled[historical,]
Y_input <- Y[historical]

X_eval <- X_scaled[-historical,]
Y_eval <- Y[-historical]

#outlier cleaning
qntl <- 0.999
outlier <- apply(X_input, 1, function(x) any(abs(x) > qnorm(qntl)))

X_cleaned <- X_input[!outlier,]
Y_cleaned <- Y_input[!outlier]

layout(matrix(seq(9),3,3))
```

Annex 4: Factor determination using random forest

```
#define variables and amount of trees

trees <- 1000
select <- 10

data <- data.frame(Y_cleaned = as.factor(Y_cleaned),X_cleaned)


#drop least important variable until limit was selected
while(ncol(data) > select){

  importance_tester <- randomForest(Y_cleaned  ~ ., data = data, ntree = trees)

  importance <- data.frame(
    Variable = row.names(importance_tester$importance),
    Importance = rescale(importance_tester$importance[, 1])
  )

  data <- data[,-(which(importance$Importance == 0)+1)]
}

important <- which(colnames(X_cleaned) %in% importance$Variable)
colnames(X_cleaned)[important]
```

Annex 5: Neural network structure

```
#networks fit
structure <- c(20,20)
patience <- 100
model <- builder(neuron_list = structure, input_shape = ncol(X_train), rate = 0
.1)
#setup technical learning parameters
model %>% compile(
  loss = 'binary_crossentropy',
  optimizer = 'adam'
)
early_stopping <- callback_early_stopping(monitor = 'val_loss',

patience = patience, restore_best_weights = T)
history <- model %>% fit(
  x = X_train,
  y = Y_train,
  epochs = 1000,
  batch_size = 64,
  validation_split = 0.2,
  callbacks = list(early_stopping),
  verbose = 1
)
print(model)
```

Output:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 20)	220
dropout (Dropout)	(None, 20)	0
dense_1 (Dense)	(None, 20)	420
dropout (Dropout)	(None, 20)	0
dense_2 (Dense)	(None, 1)	21

Total params: 661 (516.00 Byte)

Trainable params: 661 (516.00 Byte)

Non-trainable params: 0 (0.00 Byte)

Annex 6: Logistic regression structure

```
#setup critical percentila
crit_p <- 0.05

data <- data.frame(Y_train, X_train)
not_signif <- 1

#reiterate till significance is achieved
while(length(not_signif) > 0){
  lr <- glm(Y_train~., data = data, family = binomial(link = 'logit'))
  sums <- summary(lr)
  not_signif <- which(sums$coefficients[,4]>crit_p)
  data <- data[,-(not_signif+1)]
}

print(sums)
```

Output:

```
Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -7.9110     1.2478  -6.340 2.30e-10 ***
Profit.after.tax.Total.assets      197.2362    60.0435   3.285 0.001020 **
PBT.Total.assets                  -195.2422    60.0222  -3.253 0.001143 **
PAT.as...of.net.worth               -2.1541     0.2123 -10.145  < 2e-16 ***
Reserves.and.funds.Total_Assets     -55.8055    11.7739  -4.740 2.14e-06 ***
Shareholders.funds.Total_assets      -0.9725     0.2334  -4.167 3.09e-05 ***
TOL.TNW                              -1.6701     0.6598  -2.531 0.011368 *
Contingent.liabilities.Total.Assets  -1.2533     0.3485  -3.596 0.000323 ***
Current.ratio..times.                -3.8133     0.5433  -7.019 2.23e-12 ***
Debt.to.equity.ratio..times.          2.0913     0.6350   3.294 0.000989 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1647.60  on 1188  degrees of freedom
Residual deviance:  768.96  on 1179  degrees of freedom
AIC: 788.96

Number of Fisher Scoring iterations: 7
```

Annex 7: K-means algorithm

```
#define limitations in cluster search
max_k <- 10
insure <- 1000

#initialize empty objects
wcss <- vector()
elbow_points <- matrix(NA, insure, 1)
iterations <- matrix(NA, insure, max_k)
clustering_list <- list()

rws <- nrow(portfolio)

#loop through all possible k multiple times
for(j in seq(2,insure)){
  clusters <- matrix(NA,rws,max_k)
  for (i in 1:max_k) {
    kmeans_model <- kmeans(portfolio, centers = i)
    wcss[i] <- kmeans_model$tot.withinss
    clusters[,i] <- kmeans_model$cluster
  }

  derivatives <- diff(wcss)
  elbow_index <- which(derivatives == max(derivatives))
  iterations[j,] <- wcss
  elbow_points[j,] <- round(mean(elbow_index))
  clustering_list <- c(clustering_list,list(clusters))

}

#define average optimal elbow points
optimal_k <- mean(elbow_points, na.rm = T)

optimal_clusters <-
  sapply(clustering_list, function(x){
    x[,optimal_k]})

lookups <- apply(optimal_clusters,2,unique)
```

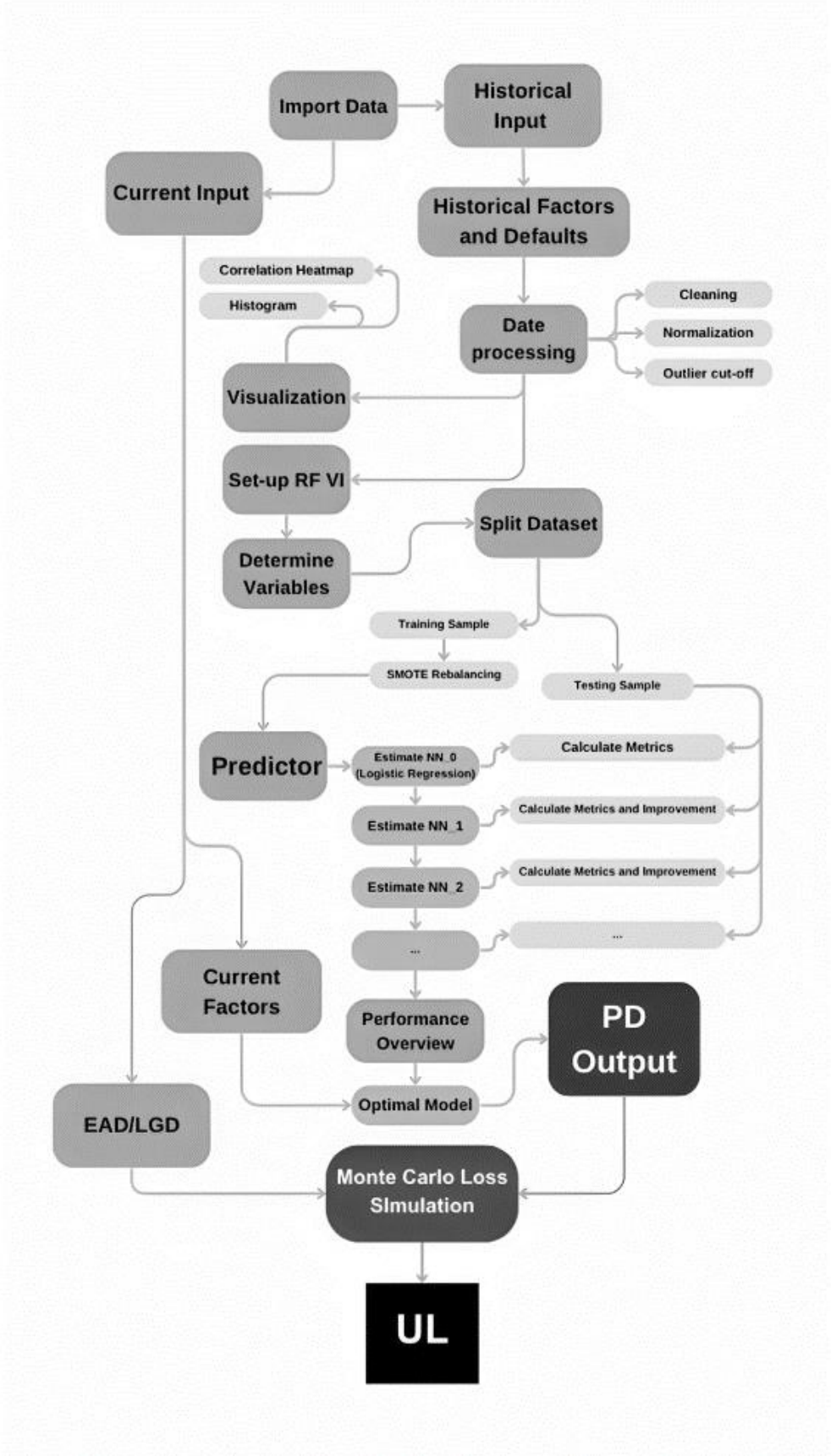
```r
#rematch values so all clusters are named same
for(i in seq(ncol(lookups))){
  lookup <- lookups[,i]
  for (j in seq(nrow(optimal_clusters))){
    optimal_clusters[j,i] <- match(optimal_clusters[j,i],lookup)
  }
}

#define cluster as one present most in each iteration
clusters <- as.numeric(apply(optimal_clusters, 1, function(x){ tbl <- table(x)
                       rownames(tbl)[which.max(tbl)]}))
```

Annex 8: Monte-Carlo simulation

```
#parameters for calculation
alpha <- 0.99

#parameters for convergence
allowed <- 100
min_n <- 1e5
track <- 1000
scale <- 1.1
patience <- 5000
plotlim  <- 1e4
verbose <- T
report  <- 100

#not changeable
delta_s <- Inf

portfolio <- read.csv('portfolio.csv')
EAD <- portfolio$EAD
PD <- portfolio$
LGD <- portfolio$LGD
N <- length(PD)
EL <- sum(PD*LGD*EAD)

#initiate recording lists

LD <- list()
errors <- list()
errors_s <- list()
errors_up <- list()

#perform minimum requirements
for (i in seq(min_n)){
  ELF <- sum((PD > runif(N))*EAD*LGD)
  LD <- c(LD, list(ELF))
}

#converge distribution
while(any(delta_s > allowed) | (i - min_n < patience)){
```

```
  UL_0 <- quantile(unlist(LD),alpha)
  ELF <- sum((PD > runif(N))*EAD*LGD)
  LD <- c(LD, list(ELF))
  i <- length(LD)
  UL_1 <- quantile(unlist(LD),alpha)
  delta <- abs(UL_0 - UL_1)
  errors <- c(errors,list(delta))
  error_s <- mean(tail(unlist(errors),min(track,length(errors))))
  errors_s <- c(errors_s ,list(error_s))
  delta_s <-  tail(unlist(errors_s),patience)
}
```

Annex 9: Summarized flow-chart of the framework suggested



Annex 10: References

1. Noriega, J. P., Rivera, L. A., Herrera, J. A. Machine Learning for Credit Risk Prediction: A Systematic Literature Review.
2. Долінський Л.Б. Оцінка та управління кредитними ризиками боргових зобов'язань: монографія. – Київ: КНЕУ, 2017. – 552 с.
3. Desabathula, A. S. D. Credit Card Approval Prediction Using Machine Learning. – 2023.
4. Shan, Q., Nilsson, M. Credit Risk Analysis with Machine Learning Techniques in Peer-to-Peer Lending Market. – 2008.
5. Tong, X. Modeling Banks' Probability of Default // Applied Economics and Finance. – 2013.
6. Chevalier, A., Redor, E. Handbook of Financial Engineering: Mergers, Acquisitions, and Credit Risk Ratings. – 2007.
7. Vidovic, L., Yue, L. Machine Learning and Credit Risk Modelling // Analytical Development Group, S&P Global Market Intelligence.
8. IFRS 9 and Expected Loss Provisioning // Financial Stability Institute. – 2014.
9. Hosmer, D. W., Lemeshow, S., Sturdivant, R. X. An Introduction to Logistic Regression Analysis and Reporting. – 2013.
10. Rao, M. B. Bayesian Networks // Handbook of Statistics. – Volume 32, 2014. – Pages 357-385.
11. Yang, X.-S. Introduction to Algorithms for Data Mining and Machine Learning. – 2019.
12. James, G., Witten, D., Hastie, T., Tibshirani, R. An Introduction to Statistical Learning with Applications in R. – 2013.
13. Dasarathy, B. V. Nearest Neighbor (NN) Classifiers. – 1991.
14. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. – 1958.
15. Krose, B. J. A. An Introduction to Neural Networks. – 1993.
16. Mair, P. Simulation-Based Methods for Credit Risk Management in Payment Service Provider Portfolios. – 2010.
17. Breiman, L., Cutler, A. Random Forests. – 2011.

18. Dahlström, K., Forssbeck, C. Simulation-Based Methods for Credit Risk Management in Payment Service Provider Portfolios. – 2023.
19. Brereton, T. J., Kroese, D. P., Chan, J. C. Monte Carlo Methods for Portfolio Credit Risk. – 2012.
20. Andersson, T., Mentes, M. Probability of Default Machine Learning Modeling: A Stress Testing Approach. – 2023.
21. ECB Guide to the Internal Capital Adequacy Assessment Process (ICAAP) // European Central Bank, Banking Supervision. – 2018.
22. Joseph, C. Advanced Credit Risk Analysis and Management. – 2013.
23. Corporate Credit Risk Dataset // Saxena, Y. – https://www.kaggle.com/datasets/yashsaxena005/corporate-credit-risk.

Glossary

ML – machine learning

Algorithms:

- LR – logistic regression
- BN – bayesian networks
- SVM – support vector machine
- KNN – k-nearest neighbors
- RF – random forest
- NN – neural network
- PAM – partitioning around medoids
- SMOTE – synthetic minority oversampling technique

Metrics:

- ROC – receiver operating characteristic
- TP – true positive
- FP – false positive
- TN – true negative
- FN – false negative
- TPR – true positive rate
- FPR – false positive rate
- CE – cross-entropy

Financial abbreviations:

- IFRS – international financial reporting standards
- LLP – loan loss provisions
- ECL – expected credit loss
- UL – unexpected loss
  ICAAP – internal capital adequacy assessment process