

MASTER THESIS

Credit Risk and Credit Derivatives

Author:
Kostas Botsi

Supervisors:
Athanasios
Gianakopoulos

A thesis submitted in fulfilment of the requirements
for the degree of Master in statistics



Δίπλωματικη εργασία

Πιστωτικός κίνδυνος και πιστωτικά παράγωγα

Συγγραφέας:
Κώστας Μπότσι

Επιβλέποντες:
Αθανασίος Γιαννακοπούλος

Διατριβή που υποβάλλεται προς εκπλήρωση των απαιτήσεων για την απόκτηση του μεταπτυχιακού τίτλου στη στατιστική

ABSTRACT

Credit risk refers the possibility the borrower can't afford to pay his loans back,which causes financial loss to the lender.The risk is inherent in varius financial products such as loans,bonds.The losses that the investor may be face are lost funds,interest not received and reduced cash flow.

In this project will focus on fundamental and statistical machine learning models for the measurement,modelling and management of credit risk as well as the study of derivative contracts so to alleviate and manage the credit risk.

Some important properties which make their quantitative modeling difficult are defaults events are rare, they may occur unexpectedly, default events involve significant losses, size of the losses are unknown before default.

There are several machine learning models and each case is different based on the variables that we have.Algorithm selection depends on various factors such as data type,features transparency and interoperability.
The statistical machine learning algorithms that we selected for analysis are the **Logistic regression**,which is a statistical model uses logit function to model probabilities [0,1], in some causes uses regularization techniques to avoid over fitting(lasso,ridge,elastic net).The next model is the **SVM** uses a hyperplane in a multidimensional surface to separate two classes in the data set.Using kernel functions,allowing to model non linearity classification problems. **Naive Bayes** classification is based on Bayes theorem that need to hold a strong assumption about the independence,in practice the assumption is often violated.This technique is easy to implement but due to assumption in independence has poor performance. **Decision Tree** model prediction is obtained through a sequence of nodes and branches.While are very flexible tool,the result often drive to a poor performnace due to overfitting.To avoid overfiting usually we use the **Random forest**.The **Random forest** model uses a number of decision tree,each tree using a random subset from the data set, to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

Περίληψη

Ο πιστωτικός κίνδυνος αναφέρεται στην πιθανότητα ο δανειολήπτης να μην έχει την οικονομική δυνατότητα να αποπληρώσει το δάνειά του, γεγονός που προκαλεί οικονομική ζημία στον δανειστή.Ο κίνδυνος είναι εγγενής σε διάφορα χρηματοοικονομικά προϊόντα όπως δάνεια, ομόλογα.Οι απώλειες που μπορεί να αντιμετωπίσει ο επενδυτής είναι τα χαμένα κεφάλαια, οι τόκοι που δεν εισπράττονται και οι μειωμένες ταμειακές ροές.

Σε αυτό το έργο θα επικεντρωθούμε σε θεμελιώδη και στατιστικά μοντέλα μηχανικής μάθησης για τη μέτρηση, μοντελοποίηση και διαχείριση του πιστωτικού κινδύνου, καθώς και στη μελέτη των συμβάσεων παραγώγων για την άμβλυνση και διαχείριση του πιστωτικού κινδύνου.

Ορισμένες σημαντικές ιδιότητες που καθιστούν δύσκολη την ποσοτική μοντελοποίησή τους είναι οπως τα γεγονότα αθέτησης πληρωμών είναι σπάνια, μπορεί να εμφανιστούν απροσδόκητα, τα γεγονότα αθέτησης περιλαμβάνουν σημαντικές απώλειες, το μέγεθος των ζημιών είναι άγνωστο πριν από την αθέτηση.

Υπάρχουν διάφορα μοντέλα μηχανικής μάθησης και κάθε περίπτωση είναι διαφορετική με βάση τις μεταβλητές Η επιλογή του αλγορίθμου εξαρτάται από διάφορους παράγοντες, όπως ο τύπος των δεδομένων, η διαφάνεια των χαρακτηριστικών και η διαλειτουργικότητα. Οι αλγόριθμοι μηχανικής μάθησης που επιλέξαμε για την ανάλυση είναι οι **Logistic regression**,ο οποίος είναι ένα στατιστικό μοντέλο που χρησιμοποιεί συνάρτηση logit για τη μοντελοποίηση πιθανοτήτων [0,1], σε ορισμένες περιπτώσεις χρησιμοποιεί τεχνικές κανονικοποίησης για την αποφυγή υπερβολικής προσαρμογής(lasso,ridge,elastic net). Το επόμενο μοντέλο είναι το **SVM** χρησιμοποιεί ένα υπερεπίπεδο σε μια πολυδιάστατη επιφάνεια για να διαχωρίσει δύο κλάσεις στο σύνολο δεδομένων. Χρησιμοποιώντας συναρτήσεις πυρήνα, επιτρέπει τη μοντελοποίηση προβλημάτων ταξινόμησης μη γραμμικότητας. Η ταξινόμηση **Naive Bayes** βασίζεται στο θεώρημα Bayes που χρειάζεται να κρατήσει μια ισχυρή υπόθεση σχετικά με την ανεξαρτησία,στην πράξη η υπόθεση συχνά παραβιάζεται.Αυτή η τεχνική είναι εύκολη στην εφαρμογή αλλά λόγω της υπόθεσης της ανεξαρτησίας έχει κακή απόδοση. Μοντέλο Δέντρο αποφάσεων Η πρόβλεψη λαμβάνεται μέσω μιας ακολουθίας κόμβων και κλάδων.Ενώ είναι πολύ ευέλικτο εργαλείο, το αποτέλεσμα συχνά οδηγεί σε κακή απόδοση λόγω υπερβολικής προσαρμογής.Για να αποφύγουμε την υπερβολική προσαρμογή συνήθως χρησιμοποιούμε

το μοντέλο **Random forest**.Το μοντέλο **Random forest** χρησιμοποιεί έναν αριθμό δέντρων απόφασης, κάθε δέντρο χρησιμοποιεί ένα τυχαίο υποσύνολο από το σύνολο δεδομένων, για να μετρήσει ένα τυχαίο υποσύνολο χαρακτηριστικών σε κάθε διαμέρισμα. Αυτή η τυχαιότητα εισάγει μεταβλητότητα μεταξύ των μεμονωμένων δέντρων, μειώνοντας τον κίνδυνο υπερπροσαρμογής και βελτιώνοντας τη συνολική απόδοση πρόβλεψης.

ACKNOWLEDGMENTS

I would like to express my thanks to my supervisor professor Athanasios giannakopoulos, whose precious advices lead me to the completion of this Thesis. Her guidance was constant. In addition, I would like to thank all of my professors in the MSc in Statistics program, who managed to provide us high quality knowledge. Last but not least, I owe much to my family and my friends whose support was non-negotiable.

Contents

I Introduction **9**
1.1 Expected loss . . . 10
1.1.1 Ratings . . . 12
1.1.2 The exposure at default (EAD) . . . 13
1.1.3 The loss given default . . . 16
1.2 Unexpected Loss . . . 17
1.2.1 The loss Distribution . . . 19
1.2.2 Monte Carlo Simulation of Losses . . . 19
1.2.3 Analytical Approximation . . . 20
1.2.4 Modeling Correlations by Means of Factor Models . . . 23

II The CreditRisk+ Model **29**
2.1 The Modeling Framework of CreditRisk+ . . . 29
2.2 Construction Step 1: Independent Obligors . . . 32
2.3 Construction Step 2: Sector Model . . . 33

III Machine Learning Models For Credit Risk **37**
3.1 Logistic regression . . . 38
3.2 Decision Tree . . . 40
3.3 Random Forest . . . 42
3.4 Support Vector Machine . . . 43
3.4.1 Linear SVM . . . 43
3.4.2 Non-linear SVM . . . 46
3.5 Neural Networks . . . 47
3.6 Naive Bayes Algorithm . . . 51
3.7 Linear Discriminant Analysis LDA . . . 53
3.7.1 A practical application of LDA . . . 53
3.7.2 Properties and assumptions of LDA . . . 54
3.7.3 Applying LDA with an example in credit risk . . . 55

IV Model performance evaluation **57**

4.1 Cross Validation in Machine Learning 57
4.2 Types of Cross-Validation 58
4.2.1 K-fold cross-validation 58
4.2.2 Holdout cross-validation 59
4.2.3 Stratified k-fold cross-validation 60
4.2.4 Leave-p-out cross-validation 61
4.2.5 Leave-one-out cross-validation 61
4.2.6 Monte Carlo cross-validation 62
4.2.7 Time series (rolling cross-validation / forward chaining method) 63
4.3 Confusion Matrix 64
4.4 AUC-ROC Curve 66

V Credit Derivatives **67**
5.1 Total Return Swaps 69
5.2 Credit Default Products 71
5.3 Basket Credit Derivatives 75
5.4 Credit Spread Products 79
5.5 Credit-Linked Notes 82

VI Application of credit risk **85**
6.1 Introduction 85
6.2 Distribution of Variables 86

Chapter I

Introduction

In today's world, with the rapid growth of technology, credit risk analysis can be effectively approached through quantitative methods, as well as the study of derivative contracts, to manage and mitigate credit risk. This project aims to introduce and apply machine learning models to predict whether a borrower will be able to repay their loans. We will evaluate the performance of these models using metrics such as Receiver Operating Characteristics (ROC), accuracy, and more.

Additionally, we will explore the implementation credit risk+ models

Following this, we will analyze credit derivatives, which are financial instruments used to manage or transfer credit risk between parties. Credit derivatives are crucial tools for risk management, enabling institutions to hedge against potential losses or speculate on changes in credit risk. Some common types of credit derivatives include Credit Default Swaps (CDS), Total Return Swaps (TRS), Credit Linked Notes (CLN), and Collateralized Debt Obligations (CDOs).

By integrating machine learning models with traditional credit risk assessment methods and exploring the role of credit derivatives, this project aims to provide a comprehensive approach to managing and mitigating credit risk.

1.1 Expected loss

The need of a loss protection in terms of an insurance, as one knows it from car or health insurances. Moreover, history shows that even good customers have a potential to default on their financial obligations, such that an insurance for not only the critical but all loans in the bank's credit portfolio makes much sense. The basic idea behind insurance is always the same. For example, in health insurance the costs of a few sick customers are covered by the total sum of revenues from the fees paid to the insurance company by all customers. Therefore, the fee that a man at the age of thirty has to pay for health insurance protection somehow reflects the insurance company's experience regarding expected costs arising from this particular group of clients. For bank loans one can argue exactly the same way: Charging an appropriate risk premium for every loan and collecting these risk premiums in an internal bank account called expected loss reserve will create a capital cushion for covering losses arising from defaulted loans.
The basic idea is as follows: The bank assigns to every customer a probability of default (PD), a loss fraction called the loss given default (LGD), describing the fraction of the loan's exposure expected to be lost in case of default, and the exposure at default (EAD) subject to be lost in the considered time period. The loss of any obligor is then defined by a loss variable.

$$\tilde{L} = EAD \times LGD \times L \quad \text{with} \quad L = \mathbb{1}_D, \quad P(D) = PD \tag{1.1}$$

Note that the quantities PD, LGD, EAD and all quantities derived from those three are measured w.r.t. a specified time horizon. We drop the time aspect for now but will come back to it later in the text.
In the setup we just described it is very natural to define the expected loss (EL) of any customer or, more general, credit-risky asset as follows

Definition

Given a loss variable $\tilde{L}$ as in (1.1), its expectation

$$\mathbb{E}L = \mathbb{E}[\tilde{L}]$$

is called the expected loss of the underlying credit-risky asset.

Proposition

If the constituents of $\tilde{L}$ in (1.1) are independent, the expected loss can be written as

$$\mathbb{E}L = \mathbb{E}[EAD] \times \mathbb{E}[LGD] \times PD. \tag{1.2}$$

Moreover, if EAD and LGD are constant values, the formula reads as

$$\mathbb{E}L = EAD \times LGD \times PD.$$

Note that making the assumption that EAD and LGD are constant values can be a good starting point for a back-of-the-envelope calculation to assign fixed values to EAD and LGD. However, in realistic situations EAD has to be modeled as a random variable due to uncertainties in payment profiles like, for instance, amortization, usage, and other drivers of EAD up to the chosen planning horizon.

1.1.1 Ratings

Originally ratings were not developed for the derivation of PDs but only for the discrimination of credit quality on an ordinal scale. And in case of rating agencies which we will introduce later it still is the case that they do not assign PDs directly to rated clients.So one has to be careful to put ratings and PDs in one bucket without keeping in mind that they are in fact different objects, as we will point out in a moment. However, because PDs are assigned to ratings and PDs are a main driver of the portfolio loss as well as all kinds of important ratios in banking, including regulatory capital related quantities, it is a common pattern that ratings and PDs are associated. Having said that, we continue our presentation from the viewpoint of the practitioner who uses ratings in the sense explained in the sequel. The assignment of default probabilities to clients typically functions via so-called rating systems. A rating system can be thought of as a discretization of PDs on an ordinal scale which is called the rating scale. Discretization of a continuous metric quantity like a PD to an ordinal scale makes life in large organizations easier although one could argue that discretization seems a bit artificial and in the context of pricing introduces unnecessary jumps in pricing grids. Well-known discretizations of PDs are the rating scales by the rating agencies Moody's, Standard & Poor's, and Fitch.
Rating scales of rating agencies look as follows. Standard & Poor's and Fitch use AAA, AA, A, BBB, BB, B, CCC, CC, C as a rating scale for rating best credit quality (AAA), 2nd-best credit quality (AA), and so on, until worst credit quality (C). The default state indicating that a company already failed in some payment obligation is denoted by D. Moody's uses Aaa, Aa, A, Baa, Ba, B, Caa, etc. to denote a comparable rating scale, again in decreasing order of credit quality. Each of the rating agencies has a finer rating scale in place to allow for a finer distinction of credit quality among obligors. Standard & Poor's and Fitch.

1.1.2 The exposure at default (EAD)

EAD is the quantity in specifying the exposure the bank does have to its borrower. In practice, banks grant to obligors so called credit lines which function like a credit limit for the single-obligor exposure For the sake of a better understanding let us introduce a working example which will accompany us through this whole section on EAD. Let us assume that a credit analyst assigns to a borrower, say, a medium sized firm, a credit line with a total limit of EUR 20m. Let us assume that the credit line is structured in the following way:

• Total credit line is EUR 20m.

• The borrower can draw EUR 12m as cash and can use the remaining EUR 8m of the credit line for so-called contingent liabilities, e.g., guarantees or comparable credit constructs but not for cash.

Now let us assume the borrower has drawn EUR 10m already. This part of the credit line is then called the outstandings of the client's exposure. The remaining open EUR 10m of the credit line are called commitments. In other words, the outstandings refer to the portion of the overall client exposure the obligor is already using. There is no randomness involved, drawn is drawn, and if the obligor defaults then the outstandings are subject to recovery and in a worst case situation could potentially be lost in total.

Of course, there is some time dynamics involved in outstandings. For instance, if the obligor pays back borrowed amounts over time then it makes a big difference whether an obligor defaults today or sometime in the future. Especially in mortgages where one often finds pre-determined amortization schemes the timing of default has a direct impact on the EAD. In our example one would need to accurately evaluate incoming cash from repayments versus newly opened parts of the credit line of the obligor which are subject to be drawn again, depending on the lending contract framework the bank and the obligor agreed to and signed.

The commitments, i.e., the remaining open EUR 10m of the borrower's credit line, are rather tricky to take into account. There is no other way than considering the exposure arising from the open part of the credit line as a random variable. So in our particular example we have EUR 10m open in the credit line but only EUR 2m can be drawn as cash. The other 8m can only be used for contingent liabilities. The two parts of the open line address different random effects:

• The EUR 2m which can be drawn as cash are driven by the likelihood that the borrower draws on them as well as by the fraction

Quantifying how much of the 2m she or he draws. Describing the situation by a simple equation, we could write

$$EAD_{\text{cash}} = \mathbb{1}_D \times X \times [2\,\text{m}]\,(\text{EUR}) \tag{1.5}$$

for the random exposure adding to current outstandings. Here, D describes the event (in the σ-field $\mathcal{F}$) that the obligor draws on the open cash credit line, and X is a random variable defined on the underlying probability space $(\Omega, \mathcal{F}, P)$, with $X(\omega) \in [0,1]$ for each $\omega \in \Omega$, quantifying the random fraction describing how much of the open 2m line is drawn.

Altogether, we are dealing with two random variables here. The equation could be made significantly more complex if one wants to take a stepwise drawing behavior into account, say, the obligor draws a partial amount in the future and another amount even later, and so on.
The remaining EUR 8m, which can be used for contingent liabilities, is also subject to various random effects. First of all, there are again one or more indicator variables reflecting the optionality of usage of free parts of the credit line. Second, there is randomness in the fact that contingent liabilities do not necessarily lead to cash exposure. A guarantee has no real exposure as of today but might convert into exposure in the future. Such random effects are typically treated by so-called conversion factors.

Let us put the pieces together for the EAD calculation. We assume that the bank has a huge loss database useful for the calibration of exposure parameters. One common exposure parameter is the so-called drawdown factor (DDF). In our example, it could be the case that the bank is able to say that the given type of obligor tends to draw on the free part of the credit line (EUR 2m) in 80% of the cases and on average uses 60% of the available cash. In other words, based on historic experience, the bank obtains parameters in (1.5) like

$$P(D) = 80\% \quad \text{and} \quad \mathbb{E}[X] = 60\%.$$

Assuming independence of $\mathbb{1}_D$ and X, this leads to an expected cash exposure for the unused part of the cash credit line of

$$\mathbb{E}[EAD_{\text{cash}}] = P(D) \times \mathbb{E}[X] \times [2\,\text{m}]\,(\text{EUR}) = 48\% \times [2\,\text{m}]\,(\text{EUR}).$$

The 48% would then be used as the DDF for this particular situation. Note that the DDF is one particular common example of conversion factors. For the contingent liability part of the credit line, we assume again the existence of a rich database that allows for the calibration of a DDF of, say, 40% for the contingent liability part and a so-called cash equivalent exposure factor (CEEF) of 80%, which is another conversion factor quantifying the conversion of the specific contingent liability, say, a guarantee, into a cash exposure. Altogether we obtain (assuming independence) the following representation for the EAD in our example:

$$\begin{aligned} \mathbb{E}[EAD] &= [10\,\mathrm{m}] + 48\% \times [2\,\mathrm{m}] + 32\% \times [8\,\mathrm{m}]\,(\mathrm{EUR}) \\ &= [10\,\mathrm{m} + 0.96\,\mathrm{m} + 2.56\,\mathrm{m}]\,(\mathrm{EUR}) \\ &= [13.52\,\mathrm{m}]\,(\mathrm{EUR}) \end{aligned} \tag{1.6}$$

where 32% =40% times 80%. So altogether, our (expected) EAD is between the already utilized 10m and the overall committed 20m but higher than the committed cash line of 12m.

Our example provided some flavor of how complicated EAD calculations can be, and in real life, it is even more complex. For example, commitments of banks to clients often include various so-called covenants, which are embedded options that may force an obligor, in times of financial distress, to provide more collateral or renegotiate the terms of the loan.

A problem is that often the obligor has some informational advantage in that the bank recognizes the financial distress of its borrowers with some delay. In case of covenants allowing the bank to close committed lines triggered by some early default indication, it really is a matter of timing whether the bank picks up such indications early enough to react before the customer has drawn on their committed lines. Bankers often speak of a *race to default*, which addresses the problem that distressed clients tend to exhaust their lines just before they default as much as possible.

The Basel Committee on Banking Supervision provides conversion factors for banks that are unable or not allowed by their regulator to use their own internal models.

1.1.3 The loss given default

A first distinction we need to make when it comes to LGDs is that of LGD as an amount of money and LGD as a percentage quote. The former is often denoted as \$LGD, which means loss given default in monetary units. The concept of LGD is best demonstrated by means of an example similar to how we proceeded for EAD.

Let us assume that a client has m credit products with the bank and pledged n collateral securities to the bank, which can, in case of default, be used for recovery purposes in order to mitigate the realized loss arising from the client's default. Each credit product is assigned an Exposure at Default (EAD), such that for m credit products we get $\text{EAD}_1, \text{EAD}_2, \ldots, \text{EAD}_m$, as well as expected recovery proceeds from the n collateral securities. We denote such recovery proceeds by $\$\text{REC}_1, \$\text{REC}_2, \ldots, \$\text{REC}_n$. This constellation, having m credit products and n collateral securities, is called an m-to-n situation.

It can be difficult to get the interdependence and relation between products and collateral right, especially in cases where we have to deal with dedicated collateral which can be used for certain purposes under certain circumstances only. Here we assume that we can simply collect "good cash" (recovery proceeds) and "bad cash" (loss exposure) together in two separate buckets which we then compare to obtain our net balance with the defaulted client. What we get from that approach is the following:

$$\$\text{LGD} = \max\left(0, (\text{EAD}_1 + \text{EAD}_2 + \cdots + \text{EAD}_m) - (\$\text{REC}_1 + \$\text{REC}_2 + \cdots + \$\text{REC}_n)\right)$$

which leads to a percentage LGD of

$$\text{LGD}\% = \frac{\$\text{LGD}}{\text{EAD}_1 + \text{EAD}_2 + \cdots + \text{EAD}_m}$$

Note that we easily wrote down the quantities $\$\text{REC}_i$, but in fact, their derivation can be quite complex and needs a rich database storing historic proceeds from collateral security categories. sufficient granularity. A typical discussion point in such calculations is, for instance, the time value of money. Recovery proceeds coming in later in time should be discounted in order to reflect the time value of money. The determination of an appropriate discount rate is just one out of many questions one has to solve in this context.

Summarizing one can say that LGD calibration is a long story and far from being trivial. The current regulatory framework forces banks with approval to use their internal PD, EAD and LGD calibrations to come up with good ideas on LGD calibration but we believe there is still a lot of ground to cover.

1.2 Unexpected Loss

At the beginning of this chapter, we introduced the Expected Loss (EL) of a transaction and imagined it as an insurance or loss reserve to cover losses the bank expects from historical default experience. However, focusing solely on expected losses is not enough. In addition to the expected loss, the bank should also ensure they have a good understanding of how much money would be necessary for covering unexpected losses, where the attribute 'unexpected' addresses losses exceeding the historic average observed in the past.

As a measure for the magnitude of the deviation of losses from the EL, the standard deviation of the loss variable $\tilde{L}$ as defined in (1.1) is a natural first choice.

1.2.1 Definition

The standard deviation

$$UL = \sqrt{V[\tilde{L}]}$$

where

$$V[\tilde{L}] = V(EAD \times LGD \times L),$$

of the loss variable $\tilde{L}$ from (1.1) is called the unexpected loss of the underlying loan or asset. One can prove the following representation formula for the UL of a loan.

$$UL = \text{EAD} \times \sqrt{V[\text{LGD}] \times \text{PD} + E[\text{LGD}]^2 \times \text{PD}(1 - \text{PD})} \tag{1.2}$$

where:

- EAD is the Exposure at Default.

- $V[\text{LGD}]$ is the variance of the Loss Given Default.
- $E[\text{LGD}]$ is the expected Loss Given Default.
- PD is the Probability of Default.

So far, we have always looked at the credit risk of a single facility, although banks have to manage large portfolios consisting of many different products with different risk characteristics. We will now indicate how one can model the total loss of a credit portfolio.

For this purpose, we consider a family of m loans:

$$\tilde{L}_i = \text{EAD}_i \times \text{LGD}_i \times L_i \tag{1.3}$$

where

$$L_i = 1_{D_i}, \quad P(D_i) = \text{PD}_i. \tag{1.4}$$

This family of loans is referred to as a portfolio from now on.
A portfolio is a collection of loss variables $\tilde{L}_i$. The portfolio loss is then defined as the random variable $\tilde{L}_P$, which is the sum of the individual loan losses:

$$\tilde{L}_P = \sum_{i=1}^{m} \tilde{L}_i = \sum_{i=1}^{m} \text{EAD}_i \times \text{LGD}_i \times L_i \tag{1.5}$$

where

$$\tilde{L}_i = \text{EAD}_i \times \text{LGD}_i \times L_i \tag{1.6}$$

with $L_i = 1_{D_i}$ and $P(D_i) = \text{PD}_i$.

1.2.6 Proposition

Given a portfolio of m loss variables as in equation (1.9) with deterministic EADs, the portfolio unexpected loss (UL) is given by

$$UL_{PF} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m} \text{EAD}_i \times \text{EAD}_j \times \text{Cov}[\text{LGD}_i \times L_i, \text{LGD}_j \times L_j]} \tag{1.7}$$

1.2.1 The loss Distribution

All risk quantities on a portfolio level are based on the portfolio loss variable L~ P F . Therefore it does not come much as a surprise that the distribution of L~ P F , the so-called loss distribution of the portfolio, plays a central role in credit risk management.All risk quantities of the credit portfolio can be identified by means of the loss distribution of the portfolio. This is an important observation, because it shows that in cases where the distribution of the portfolio loss can only be determined in an empirical way one can use empirical statistical quantities as a proxy for the respective "true" risk quantities

1.2.2 Monte Carlo Simulation of Losses

In a Monte Carlo simulation, losses are simulated and tabulated in the form of a histogram in order to obtain an empirical loss distribution of the underlying portfolio. The empirical distribution function can be determined as follows:

Assume we have simulated n potential portfolio losses $\tilde{L}_{PF}^{(1)}, \tilde{L}_{PF}^{(2)}, \ldots, \tilde{L}_{PF}^{(n)}$, taking into account the driving distributions of the single loss variables and their correlations. Then the empirical loss distribution function is given by

$$F(x) = \frac{1}{n} \sum_{j=1}^{n} 1_{[0,x]}(\tilde{L}_{PF}^{(j)})$$

where $1_{[0,x]}(\cdot)$ is the indicator function that equals 1 if the argument is in the interval $[0, x]$ and 0 otherwise. From the empirical loss distribution, we can derive all of the portfolio risk quantities introduced in the previous paragraphs. For example, the α-quantile of the loss distribution can directly be obtained from our simulation results $\tilde{L}_{PF}^{(1)}, \ldots, \tilde{L}_{PF}^{(n)}$ as follows:

Starting with the order statistics of $\tilde{L}_{PF}^{(1)}, \ldots, \tilde{L}_{PF}^{(n)}$, say

$$\tilde{L}_{PF}^{(i_1)} \leq \tilde{L}_{PF}^{(i_2)} \leq \cdots \leq \tilde{L}_{PF}^{(i_n)},$$

the α-quantile $\hat{q}_\alpha$ of the empirical loss distribution for any confidence level α is given by

$$\hat{q}_\alpha = \begin{cases} \alpha \tilde{L}_{PF}^{(i_{[n\alpha]})} + (1-\alpha)\tilde{L}_{PF}^{(i_{[n\alpha]+1})} & \text{if } n\alpha \in \mathbb{N} \\ \tilde{L}_{PF}^{(i_{[n\alpha]})} & \text{if } n\alpha / \in \mathbb{N} \end{cases}$$

1.2.3 Analytical Approximation

Another approach to the portfolio loss distribution is by analytical approximation. Roughly speaking, the analytical approximation maps an actual portfolio with unknown loss distribution to an equivalent portfolio with known loss distribution.
In practice this is often done as follows. Choose a family of distributions characterized by its first and second moment, showing the typical shape (i.e., right-skewed with fat tails13) of loss distributions From the known characteristics of the original portfolio (e.g., rating distribution, exposure distribution, maturities, etc.) calculate the first moment (EL) and estimate the second (centered) moment (UL2).
Note that the EL of the original portfolio usually can be calculated based on the information from the rating, exposure, and LGD distributions of the portfolio.
Unfortunately the second moment can not be calculated without any assumptions regarding the default correlations in the portfolio; Therefore, one now has to make an assumption regarding an average default correlation ρ
However, applying by setting all default correlations ρij equal to ρ will provide an estimated value for the original portfolio's UL Obviously the most critical part of an analytical approximation is the determination of the average asset correlation. Here one has to rely on practical experience with portfolios where the average asset correlation is known.
For example, one could compare the original portfolio with a set of typical bank portfolios for which the average asset correlations are known. In some cases there is empirical evidence regarding a reasonable range in which one would expect the unknown correlation to be located. For example, if the original portfolio is a retail portfolio, then one would expect the average asset correlation of the portfolio to be a small number, maybe contained in the interval [1%, 5%]. If the original portfolio contains loans given to large firms, then one would expect the portfolio to have a high average asset correlation, maybe somewhere between 40% and 60%. Just to give another example, the new Basel Capital Accord assumes an average asset correlation of 20% for corporate loans.We estimate the average asset correlation in Moody's universe of rated corporate bonds to be around 25%. Summarizing, we can say that calibrating14 an average correlation is on one hand a typical source of model risk, but on the other hand nevertheless often supported by some practical experience.

We will introduce a typical family of two-parameter loss distributions used for analytical approximation. Here, we want to approximate the loss distribution of the original portfolio by a beta distribution, matching the first and second moments of the original portfolio. In other words, we are looking for a random variable $X \sim \beta(a, b)$, representing the percentage portfolio loss, such that the parameters a and b solve the following equations:

$$0.003 = E[X] = \frac{a}{a+b}$$

and

$$0.002252 = V[X] = \frac{ab}{(a+b)^2(a+b+1)}.$$

Recall that the probability density function ϕ_X of X is given by

$$\phi_X(x) = \beta_{a,b}(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}$$

for $x \in [0, 1]$, with first moment and second (centered) moment

$$E[X] = \frac{a}{a+b}$$

and

$$V[X] = \frac{ab}{(a+b)^2(a+b+1)}.$$

Equations represent the moment matching addressing the “correct” beta distribution matching the first and second moments of our original portfolio. It turns out that a = 1.76944 and b = 588.045 solve equations.
The analytical approximation takes the random variable X as a proxy for the unknown loss distribution of the portfolio we started with. Following this assumption, the risk quantities of the original portfolio can be approximated by the respective quantities of the random variable X.
For example, quantiles of the loss distribution of the portfolio are calculated as quantiles of the beta distribution. Because the “true” loss distribution is substituted by a closed-form, analytical, and wellknown distribution, all necessary calculations can be done in fractions of a second. The price we have to pay for such convenience is that all calculations are subject to significant model risk. Admittedly, the beta distribution has the shape of a loss distribution, but there are various two-parameter families of probability densities having the typical shape of a loss distribution. For example, some gamma distributions, the F-distribution, and also the distributions have such a shape. Unfortunately they all have different tails, such that in case one of them would approximate really well the unknown loss distribution of the portfolio, the others automatically would be the wrong choice. Therefore, the selection of an appropriate family of distributions for an analytical approximation is a remarkable source of model risk. Nevertheless there are some families of distributions that are established as best practice choices for particular cases.

1.2.4 Modeling Correlations by Means of Factor Models

Factor models are a well established technique from multivariate statistics, applied in credit risk models, for identifying underlying drivers of correlated defaults and for reducing the computational effort regarding the calculation of correlated losses. We start by discussing the basic meaning of a factor.
Assume we have two firms A and B which are positively correlated. For example, let A be DaimlerChrysler and B stand for BMW. Then, it is quite natural to explain the positive correlation between A and B by the correlation of A and B with an underlying factor;.In our example we could think of the automotive industry as an underlying factor having significant impact on the economic future of the companies A and B. Of course there are probably some more underlying factors driving the riskiness of A and B. For example, DaimlerChrysler is to a certain extent also influenced by a factor for Germany, the United States, and eventually by some factors incorporating Aero Space and Financial Companies. BMW is certainly correlated with a country factor for Germany and probably also with some other factors. However, the crucial point is that factor models provide a way to express the correlation between A and B exclusively by means of their correlation with common factors. As already mentioned in the previous section, we additionally wish underlying factors to be interpretable in order to identify the reasons why two companies experience a down- or upturn at about the same time. For example, assume that the automotive industry gets under pressure. Then we can expect that companies A and B also get under pressure, because their fortune is related to the automotive industry. The part of the volatility of a company's financial success (e.g., incorporated by its asset value process) related to systematic factors like industries or countries is called the systematic risk of the firm. The part of the firm's asset volatility that can not be explained by systematic influences is called the specific or idiosyncratic risk of the firm.
Following Merton's model, the Global Correlation Model™ focuses on the asset value log-returns r_i of counterparties ($i = 1, \ldots, m$) at a certain planning horizon (typically 1 year), admitting a representation

$$r_i = \beta_i \Phi_i + \epsilon_i \quad (i = 1, \ldots, m).$$

Here, Φ_i is called the composite factor of firm i, because in multi-factor models Φ_i is typically a weighted sum of several factors. Equation (1.22) is a standard linear regression equation, where the sensitivity coefficient β_i captures the linear correlation of r_i and Φ_i.

In analogy to the Capital Asset Pricing Model (CAPM) β_i is called the beta of counterparty i. The variable ϵ_i represents the residual part of r_i, essentially meaning that ϵ_i is the error one makes in estimating r_i using Φ_i. When substituting r_i by $\beta_i\Phi_i$, Merton's model operates in a log-normal world, so that $\mathbf{r} = (r_1, \ldots, r_m) \sim N(\mu, \Gamma)$ is multivariate Gaussian with a correlation matrix Γ. The composite factors Φ_i and ϵ_i are accordingly also normally distributed. Another basic assumption is that ϵ_i is independent of the Φ_i's for every i. Additionally, the residuals ϵ_i are assumed to be uncorrelated. Therefore, the returns r_i are exclusively correlated by means of their composite factors. This is why Φ_i is considered the systematic part of r_i, whereas ϵ_i, due to its independence from all other involved variables, can be seen as a random effect relevant only for counterparty i.

In regression theory, one usually decomposes the variance of a variable into a systematic and a specific part. Taking variances on both sides of Equation (1.22) yields

$$V[r_i] = \beta_i^2 V[\Phi_i] \quad \text{(systematic)} + V[\epsilon_i] \quad \text{(specific)}$$

for $i = 1, \ldots, m$.

Because the variance of r_i captures the risk of unexpected movements of the asset value of counterparty i, the decomposition above can be seen as a splitting of the total risk of firm i into systematic and specific risk. The former captures the variability of r_i coming from the variability of the composite factor, which is $\beta_i^2 V[\Phi_i]$; the latter arises from the variability of the residual variable, $V[\epsilon_i]$. Note that some people refer to this as idiosyncratic risk instead of specific risk.

Alternatively to the beta of a firm, one could also look at the coefficient of determination of the regression . The coefficient of determination quantifies how much of the variability of r_i can be explained by Φ_i. This quantity is usually called the R^2 of counterparty i and constitutes an important input parameter in all credit risk models based on asset values. It is usually defined

as the systematic part of the variance of the standardized returns $\tilde{r}_i = \frac{r_i - E[r_i]}{\sqrt{V[r_i]}}$, namely

$$R_i^2 = \frac{\beta_i^2 V[\Phi_i]}{V[r_i]}$$

for $i = 1, \ldots, m$. The residual part of the total variance of the standardized returns $\tilde{r}_i$ is then given by $1 - R_i^2$, thereby quantifying the percentage value of the specific risk of counterparty i. Now we will look more carefully at the composite factors. The decomposition of a firm's variance into a systematic and a specific part is the first of three levels in the Global Correlation Model™;

The subsequent level is the decomposition of the firm's composite factor Φ_i into industry and country indices. This allows us to further break down the systematic component into contributions from different sources, reflecting how the firm's performance is influenced by broader economic factors.
For the second level, the Global Correlation Model™ decomposes every Φ_i with respect to an industry and country breakdown as follows:

$$\Phi_i = \sum_{k=1}^{K} w_{i,k} \Psi_k \quad (i = 1, \ldots, m),$$

where $\Psi_1, \ldots, \Psi_{K_0}$ are industry indices and $\Psi_{K_0+1}, \ldots, \Psi_K$ are country indices. The coefficients $w_{i,1}, \ldots, w_{i,K_0}$ are called the industry weights, and the coefficients $w_{i,K_0+1}, \ldots, w_{i,K}$ are called the country weights of counterparty i.

It is assumed that $w_{i,k} \geq 0$ for all i and k, and that

$$\sum_{k=1}^{K_0} w_{i,k} = \sum_{k=K_0+1}^{K} w_{i,k} = 1 \quad (i = 1, \ldots, m).$$

At the third and last level, a representation by a weighted sum of independent global factors is constructed for representing industry and country indices as follows:

$$\Psi_k = \sum_{n=1}^{N} b_{k,n} \Gamma_n + \delta_k \quad (k = 1, \ldots, K),$$

where δ_k denotes the Ψ_k-specific residual. Such a decomposition is typically performed by a principal components analysis (PCA) of the industry and country indices.

In vector notation, this becomes

$$\Psi = B\Gamma + \delta$$

where $B = (b_{k,n})_{k=1,\ldots,K;\, n=1,\ldots,N}$ denotes the matrix of industry and country betas, $\Gamma^T = (\Gamma_1, \ldots, \Gamma_N)$ is the global factor vector, and $\delta^T = (\delta_1, \ldots, \delta_K)$ is the vector of industry and country residuals.

Combining the previous decomposition with this representation, we finally obtain

$$r = \beta W(B\Gamma + \delta) + \epsilon.$$

In the *Global Correlation Model*™, the vector of the portfolio's returns $r^T = (r_1, \ldots, r_m)$ can conveniently be written by means of underlying factors. Note that for computational purposes, Equation (1.30) is the most convenient one, because the underlying factors are independent.

In contrast, for an economic interpretation and for scenario analysis, because the industry and country indices are easier to interpret than the global factors constructed by PCA. In fact, the industry and country indices have a clear economic meaning, whereas the global factors arising from a PCA are of synthetic type. Although they admit some vague interpretation, their meaning is not as clear as is the case for the industry and country indices.
The calculation of asset returns in the model as introduced above is now straightforward. First of all, we standardize the asset value log-returns,

$$\tilde{r}_i = \frac{r_i - E[r_i]}{\sigma_i} \quad (i = 1, \ldots, m)$$

where σ_i denotes the volatility of the asset value log-return of counterparty i.We then obtain a representation of standardized log-returns,

$$\tilde{r}_i = \frac{\beta_i}{\sigma_i}\tilde{\Phi}_i + \frac{\tilde{\epsilon}_i}{\sigma_i}$$

where $E[\tilde{\Phi}_i] = E[\tilde{\epsilon}_i] = 0$. (1.31)

Now, the asset correlation between two counterparties is given by

$$\text{Corr}[\tilde{r}_i, \tilde{r}_j] = E[\tilde{r}_i \tilde{r}_j] = \frac{\beta_i}{\sigma_i} \frac{\beta_j}{\sigma_j} E[\tilde{\Phi}_i \tilde{\Phi}_j] \tag{1.32}$$

because the *Global Correlation Model*™ assumes the residuals $\tilde{\epsilon}_i$ to be uncorrelated and independent of the composite factors. For calculation purposes, it is convenient to get rid of the volatilities σ_i and the betas β_i . This can be achieved by replacing the betas with the R-squared parameters of the involved firms.

$$R_i^2 = \frac{\beta_i^2}{\sigma_i^2} V[\Phi_i] \quad (i = 1, \ldots, m). \tag{1.33}$$

$$\text{Corr}[\tilde{r}_i, \tilde{r}_j] = \sqrt{R_i V[\Phi_i]} \sqrt{R_j V[\Phi_j]} E[\tilde{\Phi}_i \tilde{\Phi}_j].$$

because by construction we have $V[\Phi_i] = V[\tilde{\Phi}_i]$.

Chapter II

The CreditRisk+ Model

The popularity of CreditRisk+ has two major reasons:

- It seems easier to calibrate data to the model than is the case for multi-factor asset value models. Here we intentionally said "it seems" because from our point of view the calibration of bank-internal credit data to a multi-sector model is in general neither easier nor more difficult than the calibration of a multi-factor model on which an asset value model can be based.
- The second and maybe most important reason for the popularity of CreditRisk+ is its closed-form loss distribution. Using probability generating functions, the CreditRisk+ model offers (even in case of more than one sector) a full analytic description of the portfolio loss of any given credit portfolio. This enables users of CreditRisk+ to compute loss distributions in a quick and still "exact" manner. For many applications of credit risk models, this is a "nice-to-have" feature, e.g., in pricing or ABS structuring.

2.1 The Modeling Framework of CreditRisk+

Crucial in CreditRisk+ is the use of probability-generating functions. Recall that the generating function of a Poisson random variable L' with intensity λ is given by

$$G(z) = \sum_{k=0}^{\infty} P[L' = k] z^k = e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} z^k = e^{\lambda(z-1)}. \tag{4.1}$$

In order to reduce the computational effort, CreditRisk+ groups the individual exposures of the obligors in a considered portfolio into exposure bands. This is done as follows:

Choose an exposure unit amount E.Denote for any obligor i its Expected Loss by EL_i, its Exposure At Default by EAD_i, and its Loss Given Default by LGD_i. The exposure that is subject to be lost after an obligor's default is then

$$E_i = \mathrm{EAD}_i \times \mathrm{LGD}_i, \tag{4.2}$$

assuming a nonrandom LGD. The exposure ν_i respectively the Expected Loss ϵ_i of obligor i in multiples of the exposure unit E is given by

$$\nu_i = \frac{E_i}{E}, \quad \epsilon_i = \frac{\mathrm{EL}_i}{E}.$$

From this point on, CreditRisk+ "forgets" the exact exposures from the original portfolio and uses an approximation by means of exposure bands by rounding the exposures ν_i to the nearest integer number. In other words, every exposure E_i is replaced by the closest integer multiple of the unit exposure E.

Already one can see that an appropriate choice of E is essential in order to end up at an approximation that is, on one hand, "close" enough to the original exposure distribution of the portfolio in order to obtain a loss distribution applicable to the original portfolio, and on the other hand, efficient enough to really partition the portfolio into m_E exposure bands, such that m_E is significantly smaller than the original number of obligors m.

An important "rule-of-thumb" for making sure that not too much precision is lost is to at least take care that the width of exposure bands is "small" compared to the average exposure size in the portfolio. Under this rule, large portfolios (containing many loans) should admit a good approximation by exposure bands in the described manner.
In the sequel, we write $i \in [j]$ whenever obligor i is placed in the exposure band j. After the exposure grouping process, we have a partition of the portfolio into m_E exposure bands, such that obligors in a common band $[j]$ have the common exposure $\nu_{[j]}E$, where $\nu_{[j]} \in \mathbb{N}_0$ is the integer multiple of E representing all obligors i with

$$\min\{|\nu_i - n| : n \in \mathbb{N}_0\} = |\nu_i - \nu_{[j]}|,$$

where $i = 1, \ldots, m; i \in [j]; j = 1, \ldots, m_E$.
In cases where ν_i is an odd-integer multiple of 0.5, the above minimum is not uniquely defined. In such cases (which are obviously not very likely), one has to make a decision whether up- or down-rounding would be appropriate. In the sequel, we only consider $\nu_{[j]} \in \mathbb{N}$, excluding 0.

Now let us discuss how to assign a default intensity to a given exposure band. Because CreditRisk+ operates in a Poissonian framework, every obligor in the portfolio has its individual (one-year) default intensity λ_i, which can be calibrated from the obligor's one-year default probability PD_i by application of (2.12),

$$\lambda_i = -\log(1 - \mathrm{PD}_i), \quad (i = 1, \ldots, m).$$

Because the expectation of $L_i' \sim \mathrm{Pois}(\lambda_i)$ is $\mathbb{E}[L_i'] = \lambda_i$, the expected number of defaults in exposure band $[j]$ (using the additivity of expectations) is given by

$$\lambda_{[j]} = \sum_{i \in [j]} \lambda_i. \tag{4.4}$$

The Expected Loss in band $[j]$ will be denoted by $\epsilon_{[j]}$ and is calculated by multiplying the expected number of defaults in band $[j]$ with the band's exposure,

$$\epsilon_{[j]} = \lambda_{[j]} \nu_{[j]}. \tag{4.5}$$

Here, the CreditRisk+ Technical Document suggests making an adjustment of the default intensities λ_i (which so far have not been affected by the exposure band approximation process) in order to preserve the original value of the obligor's Expected Losses. This could be done by defining an adjustment factor γ_i for every obligor i by

$$\gamma_i = \frac{E_i}{\nu_{[j]} E} \quad (i \in [j], j = 1, \ldots, m_E). \tag{4.6}$$

Replacing for every obligor i the original default intensity λ_i by $\gamma_i \lambda_i$ with γ_i as defined in (4.6) preserves the original Expected Losses (ELs) after approximating the portfolio's exposure distribution by a partition into exposure bands. In the following, we assume without loss of generality that the default intensities λ_i already include the adjustment (4.6). From (4.4) and (4.5) it is straightforward to write down the portfolio's expected number of

default events (respectively the portfolio's overall default intensity), namely

$$\lambda_{\mathrm{PF}} = \sum_{j=1}^{m_E} \lambda_{[j]} = \sum_{j=1}^{m_E} \frac{\epsilon_{[j]}}{\nu_{[j]}}. \tag{4.7}$$

After these preparations, we are now ready to describe the construction of the CreditRisk+ loss distribution. We will proceed in two steps, starting with a portfolio of independent obligors and then mixing the involved Poisson distributions by means of a sector model.

2.2 Construction Step 1: Independent Obligors

We begin with a portfolio of m independent obligors whose default risk is modeled by Poisson variables L_i'. As already mentioned in Section 2.2.1, Poisson models allow for multiple defaults of a single obligor. This is an unpleasant, but due to the small occurrence probability, mostly ignored feature of all Poisson approaches to default risk.

Involving the (nonrandom) exposures E_i as defined in (4.2), we obtain loss variables $E_i L_i'$, where

$$L_1' \sim \mathrm{Pois}(\lambda_1), \ldots, L_m' \sim \mathrm{Pois}(\lambda_m) \tag{4.8}$$

are independent Poisson random variables. Grouping the individual exposures E_i into exposure bands $[j]$ and assuming the intensities λ_i incorporate the adjustments by the factors γ_i as described earlier, we obtain new loss variables $\nu_{[j]} L_i'$, where losses are measured in multiples of the exposure unit E. Because obligors are assumed to be independent, the number of defaults L' in the portfolio, respectively $L_{[j]}'$ in exposure band j, also follow a Poisson distribution, since the convolution of independent Poisson variables yields a Poisson distribution.

We obtain

$$L_{[j]}' = \sum_{i \in [j]} L_i' \sim \mathrm{Pois}(\lambda_{[j]}), \quad \lambda_{[j]} = \sum_{i \in [j]} \lambda_i, \tag{4.9}$$

for the number of defaults in exposure band $[j]$, $j = 1, \ldots, m_E$, and

$$L' = \sum_{j=1}^{m_E} \sum_{i \in [j]} L_i' \sim \mathrm{Pois}\left(\sum_{j=1}^{m_E} \lambda_{[j]} \right) = \mathrm{Pois}(\lambda_{\mathrm{PF}}) \tag{4.10}$$

(see (4.7)), for the portfolio's number of defaults. The corresponding losses (counted in multiples of the exposure unit E) are given by

$$\tilde{L}'_{[j]} = \nu_{[j]} L'_{[j]},$$

respectively

$$\tilde{L}' = \sum_{j=1}^{m_E} \nu_{[j]} L'_{[j]} = \sum_{j=1}^{m_E} \tilde{L}'_{[j]}. \tag{4.11}$$

Due to grouping the exposures $\nu_{[j]} \in \mathbb{N}$ together, we can now conveniently describe the portfolio loss by the probability-generating function of the random variable $\tilde{L}'$ defined in (4.11), applying the convolution theorem for generating functions. The probability-generating function is given by

$$G_{\tilde{L}'}(z) = \prod_{j=1}^{m_E} G_{\tilde{L}'_{[j]}}(z) = \prod_{j=1}^{m_E} \sum_{k=0}^{\infty} P[\tilde{L}'_{[j]} = \nu_{[j]} k] z^{\nu_{[j]} k} \tag{4.12}$$

$$= \prod_{j=1}^{m_E} \sum_{k=0}^{\infty} P[L'_{[j]} = k] z^{\nu_{[j]} k} = \prod_{j=1}^{m_E} \sum_{k=0}^{\infty} e^{-\lambda_{[j]}} \frac{\lambda_{[j]}^k}{k!} z^{\nu_{[j]} k}$$

$$= \prod_{j=1}^{m_E} e^{-\lambda_{[j]} + \lambda_{[j]} z^{\nu_{[j]}}} = \exp\left(\sum_{j=1}^{m_E} \lambda_{[j]} (z^{\nu_{[j]}} - 1) \right).$$

So far we assumed independence among obligors and were rewarded by the nice closed formula (4.12) for the generating function of the portfolio loss. In the next section we drop the independence assumption, but the nice feature of CreditRisk+ is that, nevertheless, it yields a closed-form loss distribution, even in the case of correlated defaults.

2.3 Construction Step 2: Sector Model

A key concept of CreditRisk+ is sector analysis. The rationale underlying sector analysis is that the volatility of the default intensity of obligors can be related to the volatility of certain underlying factors incorporating a common systematic source of credit risk.

Associated with every such background factor is a so-called sector, such that every obligor i admits a breakdown into sector weights $w_{is} \geq 0$, where

$$\sum_{s=1}^{m_S} w_{is} = 1,$$

expressing for every $s = 1, \ldots, m_S$ that sector s contributes with a fraction w_{is} to the default intensity of obligor i. Here m_S denotes the number of involved sectors. Obviously, the calibration of sectors and sector weights is the crucial challenge in CreditRisk+. For example, sectors could be constructed with respect to industries, countries, or rating classes.
In order to approach the sector model of CreditRisk+, we rewrite Equation (4.12):

$$G_{\tilde{L}'}(z) = \exp\left(\sum_{j=1}^{m_E} \lambda_{[j]}(z^{\nu_{[j]}} - 1)\right) \tag{4.13}$$

$$= \exp\left(\lambda_{\mathrm{PF}}\left(\sum_{j=1}^{m_E} \frac{\lambda_{[j]}}{\lambda_{\mathrm{PF}}} z^{\nu_{[j]}} - 1\right)\right),$$

where λ_{PF} is defined as in (4.7). Defining functions

$$G_{L'}(z) = e^{\lambda_{\mathrm{PF}}(z-1)} \quad \text{and} \quad G_N(z) = \sum_{j=1}^{m_E} \frac{\lambda_{[j]}}{\lambda_{\mathrm{PF}}} z^{\nu_{[j]}}, \tag{4.14}$$

we see that the generating function of the portfolio loss variable $\tilde{L}'$ can be written as

$$G_{\tilde{L}'}(z) = G_{L'} \circ G_N(z) = e^{\lambda_{\mathrm{PF}}(G_N(z)-1)}. \tag{4.15}$$

Therefore, the portfolio loss $\tilde{L}'$ has a so-called compound distribution, essentially meaning that the randomness inherent in the portfolio loss is due to the compound effect of two independent sources of randomness. The first source of randomness arises from the uncertainty regarding the number of defaults in the portfolio, captured by the Poisson random variable L' with intensity λ_{PF} defined in (4.10). The function $G_{L'}(z)$ is the generating function of L'; recall (4.1).

The second source of randomness is due to the uncertainty about the exposure bands affected by the L' defaults. The function $G_N(z)$ is the generating

function of a random variable N taking values in $\{\nu_{[1]}, \ldots, \nu_{[m_E]}\}$ with distribution

$$P[N = \nu_{[j]}] = \frac{\lambda_{[j]}}{\lambda_{\text{PF}}} \quad (j = 1, \ldots, m_E). \tag{4.16}$$

For some more background on compound distributions, refer to the literature. For example, in [86] the reader will find theory as well as Some interesting examples will be provided later, where we will obtain the generating function of sector losses in a form that, conditional on the sector's default rate, replicates Equation (4.15).

Let us assume that we have parameterized our portfolio by means of m_S sectors. CreditRisk+ assumes that a gamma-distributed random variable

$$\Lambda^{(s)} \sim \Gamma(\alpha_s, \beta_s) \quad (s = 1, \ldots, m_S)$$

is assigned to every sector; see Figure 2.2 for an illustration of gamma densities. The number of defaults in any sector s follows a gamma-mixed Poisson distribution with random intensity $\Lambda^{(s)}$; see also Section 2.2.2. Hereby, it is always assumed that the sector variables $\Lambda^{(1)}, \ldots, \Lambda^{(m_S)}$ are independent.

For the calibration of $\Lambda^{(s)}$, recall from (2.38) that the first and second moments of $\Lambda^{(s)}$ are

$$E[\Lambda^{(s)}] = \alpha_s \beta_s, \quad V[\Lambda^{(s)}] = \alpha_s \beta_s^2. \tag{4.17}$$

We denote the expectation of the random intensity $\Lambda^{(s)}$ by $\lambda^{(s)}$. The volatility of $\Lambda^{(s)}$ is denoted by $\sigma^{(s)}$. Altogether, we have from (4.17)

$$\lambda^{(s)} = \alpha_s \beta_s, \quad \sigma^{(s)} = \sqrt{\alpha_s}\beta_s. \tag{4.18}$$

Knowing the values of $\lambda^{(s)}$ and $\sigma^{(s)}$ determines the parameters α_s and β_s of the sector variable $\Lambda^{(s)}$.

For every sector, we now follow the approach that has taken us to Equation (4.15). More explicitly, we first find the generating function of the number of defaults in sector s, then obtain the generating function for the distribution of

default events among the exposures in sector s, and finally get the portfolio-loss-generating function as the product of the compound sector-generating functions.

Chapter III

Machine Learning Models For Credit Risk

In this section, we introduce how machine learning models can play an effective role in credit risk analysis. These models were discovered many decades ago but have resurfaced in recent years due to the evolution of computers and the availability of vast amounts of data. With the large datasets available today, these models can be trained to learn from the data, becoming more effective and intelligent in their decision-making processes.

3.1 Logistic regression

Despite being a traditional statistical model, logistic regression is often used as a baseline in credit risk modeling. It estimates the probability that a borrower will default the loan.Models the log odds of the event as a linear combination of one or more variables.There is a single binary dependent variable and where are labeled as "0" not to pay of the dept and "1" to pay the dept.The logistic regression models the probability of outputs in term of inputs,it is not a statistical classification model,but we can use it as a classifier if we put a threshold and say above this probability classify to class 1 or below this probability classify to class 0.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{3.1}$$

where $\beta_0 = -\frac{\mu}{s}$ is the intercept (also known as the vertical or y-intercept of the line $y = \beta_0 + \beta_1 x$), and $\beta_1 = \frac{1}{s}$ is the inverse scale parameter or rate parameter. These parameters represent the y-intercept and slope of the log-odds as a function of x. Conversely, μ and s can be expressed in terms of the parameters β_0 and β_1 as follows:

$$\mu = -\frac{\beta_0}{\beta_1} \tag{3.2}$$

$$s = \frac{1}{\beta_1} \tag{3.3}$$

The use the measurement of the goodness of fit we use the log loss

$$\ell_k = \begin{cases} -\ln p_k & \text{if } y_k = 1, \\ -\ln(1 - p_k) & \text{if } y_k = 0. \end{cases} \tag{3.4}$$

The loss function can be interpreted as the surpisonal of the actual value yk relative to the probability of this value.The loss function is often greater from zero and in perfect prediction it is zero,it aprroaches infinity as the prediction gets worst.This can be written as

$$\ell_k = -y_k \ln p_k - (1 - y_k) \ln(1 - p_k). \tag{3.5}$$

The (2.5) expression it is called and cross entropy between the actual value (yk,1-yk) and the predicted value (pk,1-pk).The purpose is minimaze the

cross entropy. Alternatively we we can maximaze the inverse.

$$\ell = \sum_{k:y_k=1} \ln(p_k) + \sum_{k:y_k=0} \ln(1-p_k) = \sum_{k=1}^{K} (y_k \ln(p_k) + (1-y_k) \ln(1-p_k)) \tag{3.6}$$

In the case that the model we have multiple explanatory variables we can write it as follow

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m \tag{3.7}$$

and

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m)}} \tag{3.8}$$

The dependent variable can be describe as a binomial distribution variable with two outcome "0" for the not pay the loan and "1" pay the loan.

$$\begin{aligned} Y_i \mid x_{1,i}, \ldots, x_{m,i} &\sim \text{Bernoulli}(p_i) \\ \mathbb{E}[Y_i \mid x_{1,i}, \ldots, x_{m,i}] &= p_i \\ \Pr(Y_i = y \mid x_{1,i}, \ldots, x_{m,i}) &= \begin{cases} p_i & \text{if } y = 1 \\ 1 - p_i & \text{if } y = 0 \end{cases} \\ \Pr(Y_i = y \mid x_{1,i}, \ldots, x_{m,i}) &= p_i^y (1-p_i)^{(1-y)} \end{aligned} \tag{3.9}$$

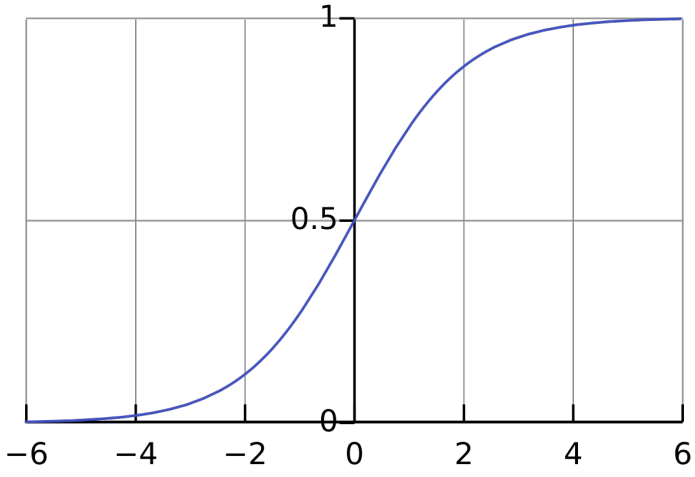


FIGURE 3.1: Logistic-curve

3.2 Decision Tree

Decision tree is a non-parametric algorithm,which can be used for classification and regression method.In our cause to conclude if a borrower will pay or not his loan we will use it for classification purpose.It is a hierarchical tree structure which consists root nodes,branches,internal nodes and leaf nodes.The decision tree start with root node,which does not consist any branches.The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogeneous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset. In this mini example we can see how

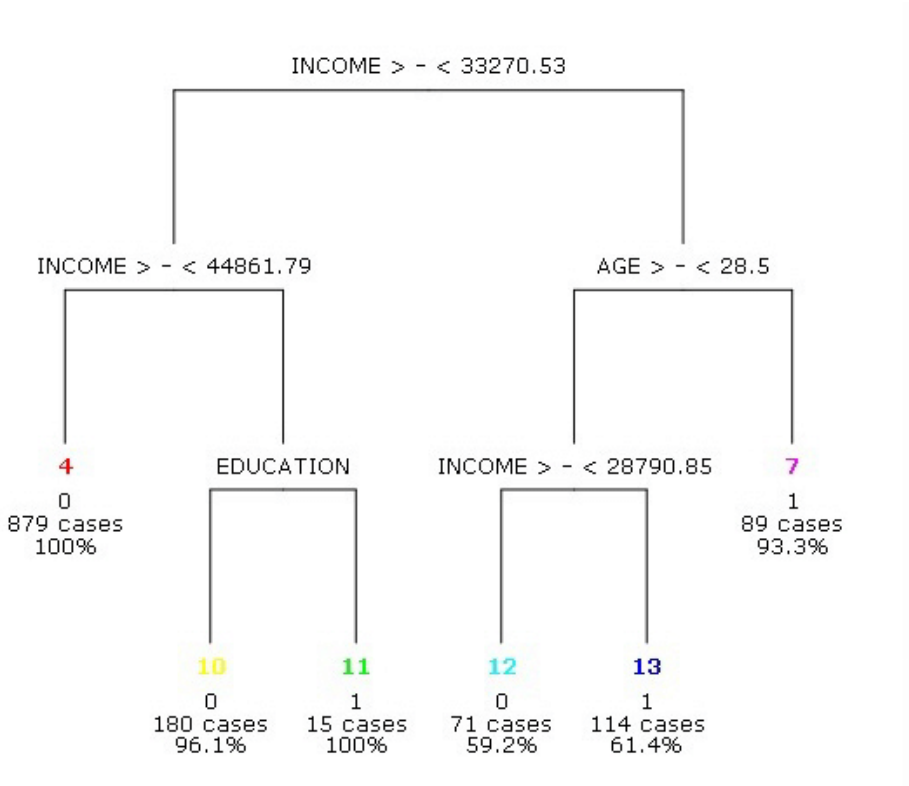


FIGURE 3.2: Example of Decision Tree in Credit Scoring

the decision tree can work in credit scoring.We see base on variable income education and age how we can give a score about the borrower to give his loan back.Now the question that arises is how slip the nodes and which variable should be first.For this question can be answered where gini index and information gain can help to splitting criterion for decision tree models. First we need to discuss entropy,which is a measure,which measures the impurity

of the sample values. It is defined with by the following formula, where

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{3.10}$$

Entropy takes values between 0 and 1,if all samples in dataset belong to one class then takes the value 0,if in splits fifty fifty then in takes the value 1.In order to find the best feature to split on and find the optimal decision tree,should have the smallest amount of entropy.Information gain is the difference in entropy before and after split on a given attribute.The highest the information gain the best split it is doing resulting in the best classification.

$$IG(T, A) = H(T) - \sum_{v \in \text{Values}(A)} \frac{|T_v|}{|T|} H(T_v) \tag{3.11}$$

- $H(T)$ is the entropy of the original dataset T.
- T_v is the subset of T for which attribute A has value v.
- $\frac{|T_v|}{|T|}$ is the proportion of the subset T_v to the entire dataset T.
- $H(T_v)$ is the entropy of the subset T_v.
- Values(A) is the set of all possible values of attribute A.

Gini impurity is the probability of incorrectly classifying random data point in the dataset if it were labeled based on the class distribution of the dataset. Similar to entropy, if set, S, is pure—i.e. belonging to one class) then, its impurity is zero

$$G = 1 - \sum_{i=1}^{n} p_i^2 \tag{3.12}$$

- p_i is the probability of an element being classified into class i.

The advantages are easily easy to interpret,little data preparation required and more flexible.The disadvantages are that are prone to overffiting it means that has good result in training data and bad result at test data,high variance estimators and is more costly.

3.3 Random Forest

Random Forest it works by creating numbers of Decision Trees,each tree is constructed used random sample from the data.This randomness introduces variability among trees where it helps to avoid over fitting and improving prediction performance.

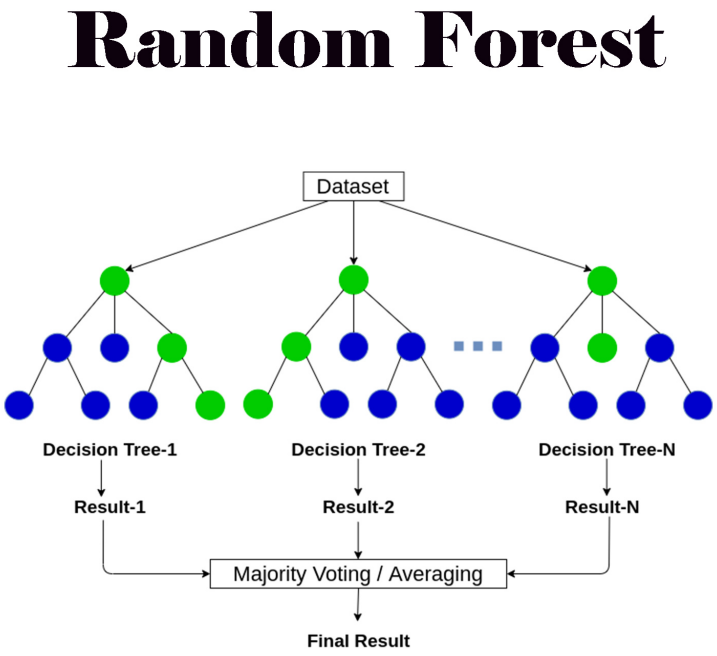


FIGURE 3.3: Example of Random Forest

The random Forest has three main hyper-parameters that need to specify,which needs to specify before set.This includes nodes,the number of trees and the number of feature to sample.The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample. Of that training sample, one-third of it is set aside as test data, known as the out-of-bag (oob) sample.To summarize the step need to follow are the following.

1. Select random K data points from the training set.
2. Build the decision trees associated with the selected data points (Subsets).
3. Choose the number N for decision trees that you want to build.

4. Repeat Steps 1 and 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The benefits are that Reduced risk of overfitting,Provides flexibility and Easy to determine feature importance.Challenges need to face time-consuming process,Requires more resources and More complex

3.4 Support Vector Machine

SVM is a supervised Machine Learning Algorithm that classify data,finds an optimal hyperline or line that maximizes the distance between each class in N-dimensional space.The number of features in the input data, determine if the hyperline is a plane in 2D space,or a plane in N-dimensional space.It can handle non-linear and linear task.When the data are non-linear spreadable,transform the data into higher space with kernel functions

No.	Kernel Type	Formula
1	Linear	$K(x_i, x_j) = x_j^T x_i$
2	Polynomial	$K(x_i, x_j) = (x_j^T x_i + 1)^p$
3	Gaussian/RBF (Radial Basis Function)	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
4	Sigmoid	$K(x_i, x_j) = \tanh(\alpha x_j^T x_i + \theta)$

TABLE 3.1: SVM Kernels

3.4.1 Linear SVM

The equation for the linear hyperplane can be written as:

$$w^T x + b = 0 \tag{3.13}$$

The vector **w** represents the normal vector to the hyperplane, i.e., the direction perpendicular to the hyperplane. The parameter b in the equation represents the offset or distance of the hyperplane from the origin along the normal vector **w**.

The distance between a data point x_i and the decision boundary can be calculated as:

$$d_i = \frac{w^T x_i + b}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\|$ represents the Euclidean norm of the weight vector $\mathbf{w}$. For a Linear SVM classifier:

$$\hat{y} = \begin{cases} 1 & : w^T x + b \geq 0 \\ 0 & : w^T x + b < 0 \end{cases} \tag{3.14}$$

There are two approaches to calculate the **margin** or the **maximum distance between classes** which are soft margin classification and hard margin classification.If we use hard margin classification the data points will be perfect separable outside of the support vectors.
For a Hard Margin Linear SVM classifier:

$$\min_{w,b} \frac{1}{2} w^T w = \min_{W,b} \frac{1}{2} \|w\|^2 \tag{3.15}$$

subject to

$$y_i(w^T x_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, 3, \ldots, m \tag{3.16}$$

The target variable or label for the i-th training instance is denoted by the symbol t_i in this statement. And $t_i = -1$ for negative occurrences (when $y_i = 0$) and $t_i = 1$ for positive instances (when $y_i = 1$) respectively. Because we require the decision boundary that satisfies the constraint:

$$t_i(w^T x_i + b) \geq 1 \tag{3.17}$$

For a Soft Margin Linear SVM classifier:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \zeta_i \tag{3.18}$$

subject to

$$y_i(w^T x_i + b) \geq 1 - \zeta_i \quad \text{and} \quad \zeta_i \geq 0 \quad \text{for} \quad i = 1, 2, 3, \ldots, m \tag{3.19}$$

Dual Problem: A dual problem of the optimization problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The optimal Lagrange multipliers α_i that maximize the following dual objective function:

$$\max_{\alpha} \left(\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j t_i t_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i \right) \tag{3.20}$$

where,

- α_i is the Lagrange multiplier associated with the i-th training sample. - $K(x_i, x_j)$ is the kernel function that computes the similarity between two samples x_i and x_j. It allows SVM to handle nonlinear classification problems by implicitly mapping the samples into a higher-dimensional feature space. - The term $\sum \alpha_i$ represents the sum of all Lagrange multipliers.

The SVM decision boundary can be described in terms of these optimal Lagrange multipliers and the support vectors once the dual issue has been solved and the optimal Lagrange multipliers have been discovered. The training samples that have $\alpha_i > 0$ are the support vectors, while the decision boundary is supplied by:

$$w = \sum_{i=1}^{m} \alpha_i t_i K(x_i, x) + b \tag{3.21}$$

$$t_i(w^T x_i - b) = 1 \quad \Leftrightarrow \quad b = w^T x_i - t_i \tag{3.22}$$

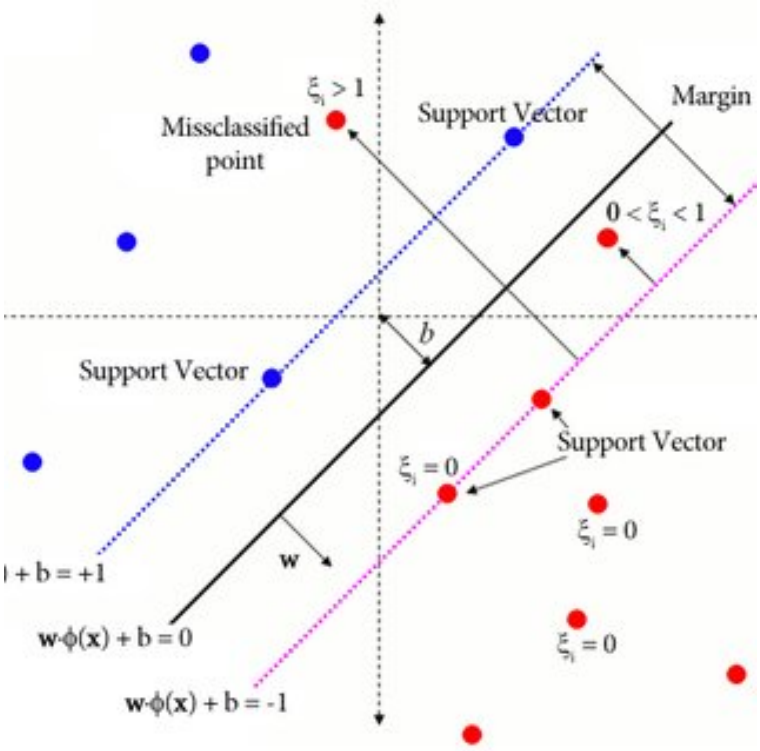


FIGURE 3.4: Expanation of SVM

3.4.2 Non-linear SVM

Much of the real word scenarios are that the data are non-linear.So we need non-linear svm .In order to make the data separable we transform the data into higher dimensional space,but this can cause the problem of the overfitting and computational complexity.The kernel trick comes to reduce some of this complexity,making the computation more efficient and it does this by replacing dot product calculations with an equivalent kernel function.Some of this kernel function are the polynomial,radial basis function RDF and sigmoid

3.5 Neural Networks

Neural network is a machine learning algorithm that takes decision manner similar to human brain.Every neural layer consists one or more input layer,one or more hidden layer and an ouput layer.Each node connects to others, and has its own associated weight and threshold.If the output of an individual node is above a specific threshold the node is activated, sending data to the next layer of the network.Neural also called and activation function.

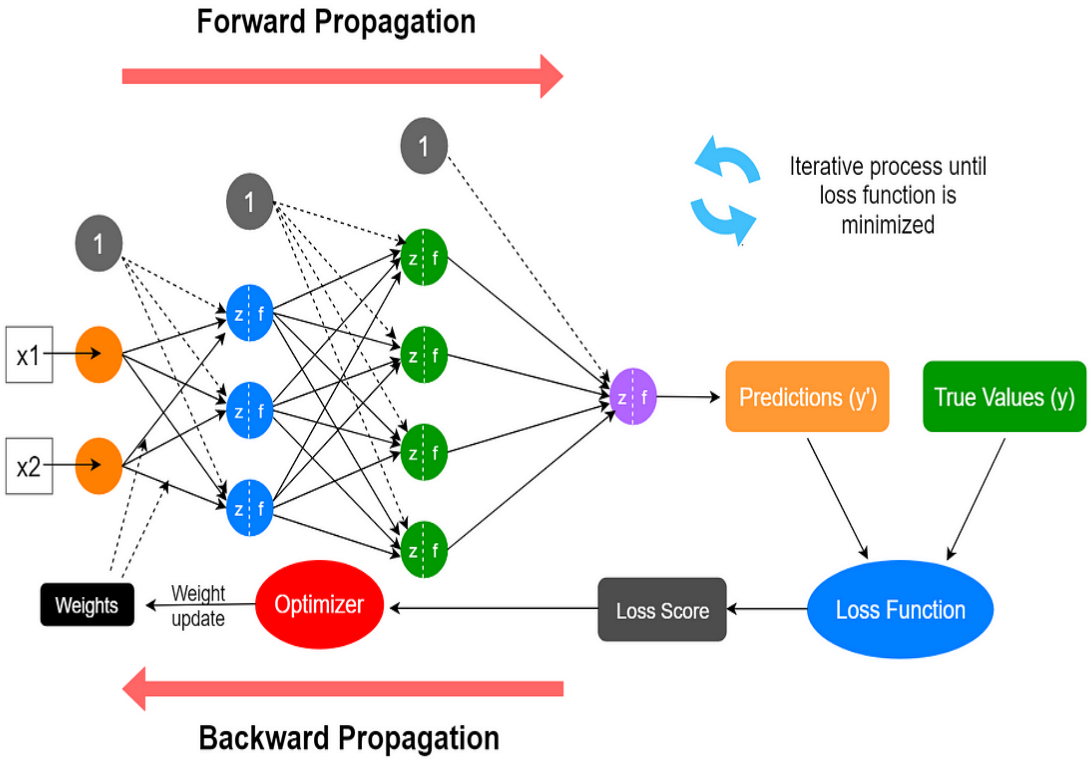


FIGURE 3.5: Neural network example

In this illustration it will be explained how the neural networks work.We have the

1. an Input layer whose nodes represent observation variables,
2. an Output layer whose nodes represent the prediction,
3. one or two Hidden layers used to capture the non-linearity of the data.

The number of input layers are equal to the number of features,the ouput layer holds a value between zero and one and base on a given threshold we

can classify whether the borrower will default or no the borrow amount.The hidden layers are responsible for holding the pattern between them.The connection of every pair of neurons are called layers,these weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs. all inputs are then multiplied by their respective weights and then summed,it is called weighted sum

$$\sum w_i x_i + \text{bias} = w_1 x_1 + w_2 x_2 + w_3 x_3 + \text{bias} \tag{3.23}$$

The bias variable determine how large the weighted sum will be.If the bias is negative the weighted sum will be low,or if it will be high then it the weighted sum will be high.
This weighted sum it is passed inside the activation function. One commonly used activation function is the Sigmoid function,

$$sigma(Z) = \frac{1}{1+\exp(-Z)}. \tag{3.24}$$

$$x_{i,j} = \sigma\left(\sum_k w_{k,j} x_{i,k} + b\right), \tag{3.25}$$

which is the prediction of $X^{[1]}$. The forward propagation it refers that it start from the input layer to the final output layer,the procedure is that we start from random weights,train the model and then put the trained weights backward.To be this more clearly we present the following graph.

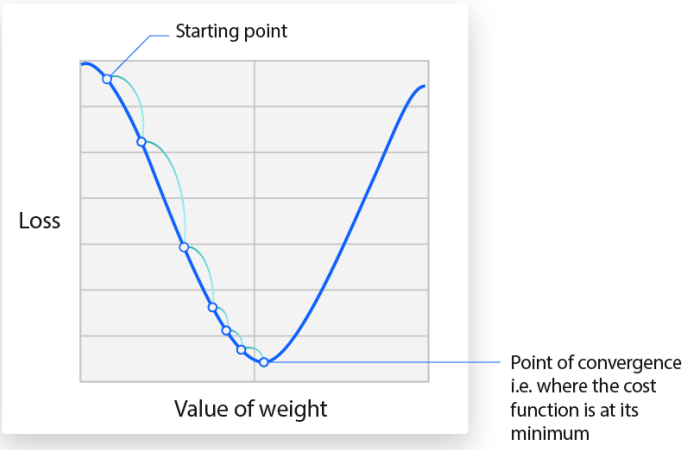


FIGURE 3.6: Caption

We try to minimize the cost function,which refer to the error representation between the predicted and the actual value.For example if we put randomly the weights the error will be high,so we need to find the heights that minimize the loss as we see in the figure 2.6.To achieve a minimal cost, the gradient descent method updates all parameters backward.
article amsmath

1. Initialize the weights.
2. Forward propagation:

$$
\begin{aligned}
z^{[1]} &= W^{[1]}a^{[0]} + b^{[1]}, \\
a^{[1]} &= f(z^{[1]}) \\
z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]}, \\
a^{[2]} &= f(z^{[2]})
\end{aligned}
$$

3. Find the cost value:

$$J = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, \hat{Y}_i)$$

4. Backward propagation (gradient descent procedure to update the weights):
   - Compute the error for the output layer:

$$\delta^{[2]} = a^{[2]} - Y$$

   - Compute the gradient with respect to $W^{[2]}$ and $b^{[2]}$:

$$dW^{[2]} = \frac{1}{n}\sum_{i=1}^{n}(a^{[1]})_i^T \delta_i^{[2]}, \quad db^{[2]} = \frac{1}{n}\sum_{i=1}^{n}\delta_i^{[2]}$$

   - Compute the error for the hidden layer:

$$\delta^{[1]} = (W^{[2]})^T\delta^{[2]} \cdot f'(z^{[1]})$$

   - Compute the gradient with respect to $W^{[1]}$ and $b^{[1]}$:

$$dW^{[1]} = \frac{1}{n}\sum_{i=1}^{n}(a^{[0]})_i^T \delta_i^{[1]}, \quad db^{[1]} = \frac{1}{n}\sum_{i=1}^{n}\delta_i^{[1]}$$

- Update the weights and biases:

$$W^{[2]} \leftarrow W^{[2]} - \eta dW^{[2]}, \quad b^{[2]} \leftarrow b^{[2]} - \eta db^{[2]}$$

$$W^{[1]} \leftarrow W^{[1]} - \eta dW^{[1]}, \quad b^{[1]} \leftarrow b^{[1]} - \eta db^{[1]}$$

5. Repeat steps 2, 3, and 4 until convergence.

3.6 Naive Bayes Algorithm

Naive Bayes is a probabilistic machine learning model based on bayes theorem where is used for classification.In the context of credit risk Naive Bayes can be employed to give the probability a borrower will default on a loan.Bayes theorem works as the probability of event based on the prior knowledge.

$$P(A \mid B) = \frac{P(B) \times P(B \mid A) \times P(A)}{P(B)} \tag{3.26}$$

- $P(A \mid B)$: The probability of event A occurring given that B is true (posterior probability).
- $P(B \mid A)$: The probability of event B occurring given that A is true (likelihood).
- $P(A)$: The probability of event A occurring (prior probability).
- $P(B)$: The probability of event B occurring (marginal likelihood).

The Naive Bayes algorithm to apply,it needs to hold a assumption that all the predictors are independent of each other given the class label.This assumption allows the model to handle high dimensional data,the predictors can be Borrower's income level,Employment status,Loan amount,Debt-to-income ratio.
To understand better how the algorithm works we have the classes and the predictors.
Classes: The target variable (class) in credit risk is typically binary: Default or Non-Default.

Features: The model takes various borrower attributes as input features to predict the probability of default. The model calculates

$$P(\text{Default} = \text{Yes} \mid \text{Features}) \quad \text{and} \quad P(\text{Default} = \text{No} \mid \text{Features})$$

The steps are the following.
Calculate Prior Probabilities: Determine the prior probability of a borrower defaulting ($P(\text{Default} = \text{Yes})$) and not defaulting ($P(\text{Default} = \text{No})$), based on the training data.
Calculate Likelihoods: For each feature, calculate the likelihood of observing that feature value given that the borrower defaults and given that the borrower does not default.

Apply Bayes' Theorem: Use Bayes' Theorem to calculate the posterior probability of default given the observed feature values.
Prediction: Compare the posterior probabilities for default and non-default. The class with the higher probability is chosen as the model's prediction.

3.7 Linear Discriminant Analysis LDA

To mitigate risk, financial institutions must identify and minimize credit default. LDA can help identify applicants who might be likely to default on loans from those who are creditworthy by sifting through financial factors and behavior data.
Linear discriminant analysis (LDA) is an approach used in supervised machine learning to solve multi-class classification problems. LDA separates multiple classes with multiple features through data dimensionality reduction. This technique is important in data science as it helps optimize machine learning models.
LDA works by identifying a linear combination of features that separates or characterizes two or more classes of objects or events. LDA does this by projecting data with two or more dimensions into one dimension so that it can be more easily classified. The technique is, therefore, sometimes referred to as dimensionality reduction.

3.7.1 A practical application of LDA

Suppose that a bank is deciding whether to approve or reject loan applications. The bank uses two features to make this decision: the applicant's credit score and annual income.

Here, the two features or classes are plotted on a 2-dimensional (2D) plane with an X-Y axis. If we tried to classify approvals using just one feature, we might observe overlap. By applying LDA, we can draw a straight line that completely separates these two class data points. LDA achieves this by using the X–Y axis to create a new axis, separating the different classes with a straight line and projecting data onto the new axis.

To create this new axis and reduce dimensionality, LDA follows these criteria:

Maximize the distance between the means of two classes.
Minimize the variance within individual classes.

3.7.2 Properties and assumptions of LDA

Linear Discriminant Analysis (LDA) operates by projecting a feature space, i.e., a dataset with n-dimensions, onto a smaller space k, where $k \leq n - 1$, without losing class information. An LDA model comprises the statistical properties that are calculated for the data in each class. When there are multiple features or variables, these properties are calculated over the multivariate Gaussian distribution.

The multivariates are:

- **Means**
- **Covariance matrix**, which measures how each variable or feature relates to others within the class

The statistical properties that are estimated from the data set are fed into the LDA function to make predictions and create the LDA model. There are some constraints to bear in mind, as the model assumes the following:

1. The input dataset has a Gaussian distribution, where plotting the data points gives a bell-shaped curve.
2. The data set is linearly separable, meaning LDA can draw a straight line or a decision boundary that separates the data points.
3. Each class has the same covariance matrix.

For these reasons, LDA may not perform well in high-dimensional feature spaces.

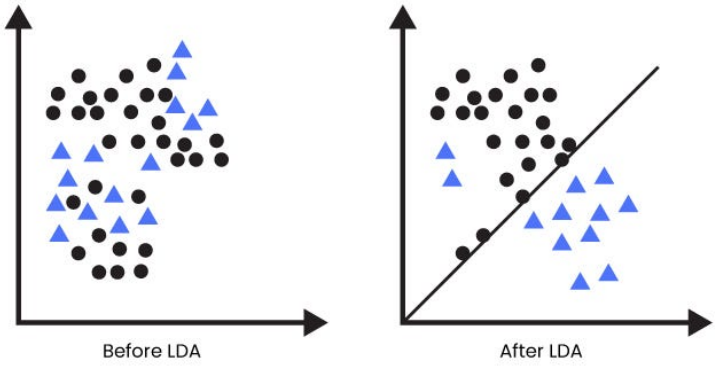


FIGURE 3.7: LDA

3.7.3 Applying LDA with an example in credit risk

Let's use the equation to work through a loan approval example. To recap, the bank is deciding whether to approve or reject loan applications. The bank uses two features to make this decision: the applicant's credit score (x) and annual income. The bank has collected historical data on previous loan applicants and whether the loans were approved.

Class ω_0 represents "Loan rejected."
Class ω_1 represents "Loan approved."

Using the linear discriminant function, the bank can calculate a score ($\delta(x)$) for each loan application.

The equation for the linear discriminant function might look like this:

$$\delta(x) = x \times \left(\sigma^2 \times (\mu_0 - \mu_1) - 2 \times \sigma^2 \times (\mu_0^2 - \mu_1^2) + \ln\left(\frac{P(\omega_0)}{P(\omega_1)}\right)\right)$$

where:

- x represents the applicant's credit score and annual income.
- μ_0 and μ_1 are the means of these features for the two classes: "Loan rejected" and "Loan approved."
- σ^2 is the common within-class variance.
- $P(\omega_0)$ is the prior probability of "Loan rejected".
- $P(\omega_1)$ is the prior probability of "Loan approved".

The bank computes the linear discriminant function for each loan application.

- If $\delta(x)$ is positive, it suggests that the loan application is more likely to be approved.
- If $\delta(x)$ is negative, it suggests that the loan application is more likely to be rejected.

The bank can thus automate its loan approval process, making quicker and more consistent decisions while minimizing human bias.

Chapter IV

Model performance evaluation

In this chapter, we will explore various techniques for evaluating model performance. Since different datasets may require different approaches, it's crucial to identify the model that offers the best accuracy and minimizes prediction errors. By applying these evaluation techniques, we can ensure that the most suitable model is selected for each dataset, ultimately improving prediction quality and reliability.

4.1 Cross Validation in Machine Learning

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance. Cross validation is an important step in the machine learning process and helps to ensure that the model selected for deployment is robust and generalizes well to new data.
The main purpose of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data.

4.2 Types of Cross-Validation

There are several types of cross validation techniques, including k-fold cross validation, leave-one-out cross validation, and Holdout validation, Stratified Cross-Validation. The choice of technique depends on the size and nature of the data, as well as the specific requirements of the modeling problem.

4.2.1 K-fold cross-validation

In this technique, the whole dataset is partitioned in k parts of equal size and each partition is called a fold. It's known as k-fold since there are k parts where k can be any integer - 3,4,5, etc.

One fold is used for validation and other K-1 folds are used for training the model. To use every fold as a validation set and other left-outs as a training set, this technique is repeated k times until each fold is used once. The image

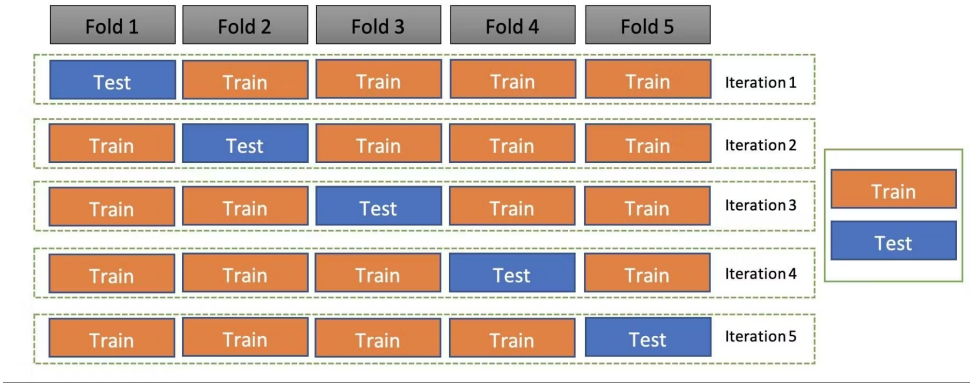


FIGURE 4.1: Enter Caption

above shows 5 folds and hence, 5 iterations. In each iteration, one fold is the test set/validation set and the other k-1 sets (4 sets) are the train set. To get the final accuracy, you need to take the accuracy of the k-models validation data.

This validation technique is not considered suitable for imbalanced datasets as the model will not get trained properly owing to the proper ratio of each class's data.

4.2.2 Holdout cross-validation

Also called a train-test split, holdout cross-validation has the entire dataset partitioned randomly into a training set and a validation set. A rule of thumb to partition data is that nearly 70% of the whole dataset will be used as a training set and the remaining 30% will be used as a validation set. Since the dataset is split into only two sets, the model is built just one time on the training set and executed faster.

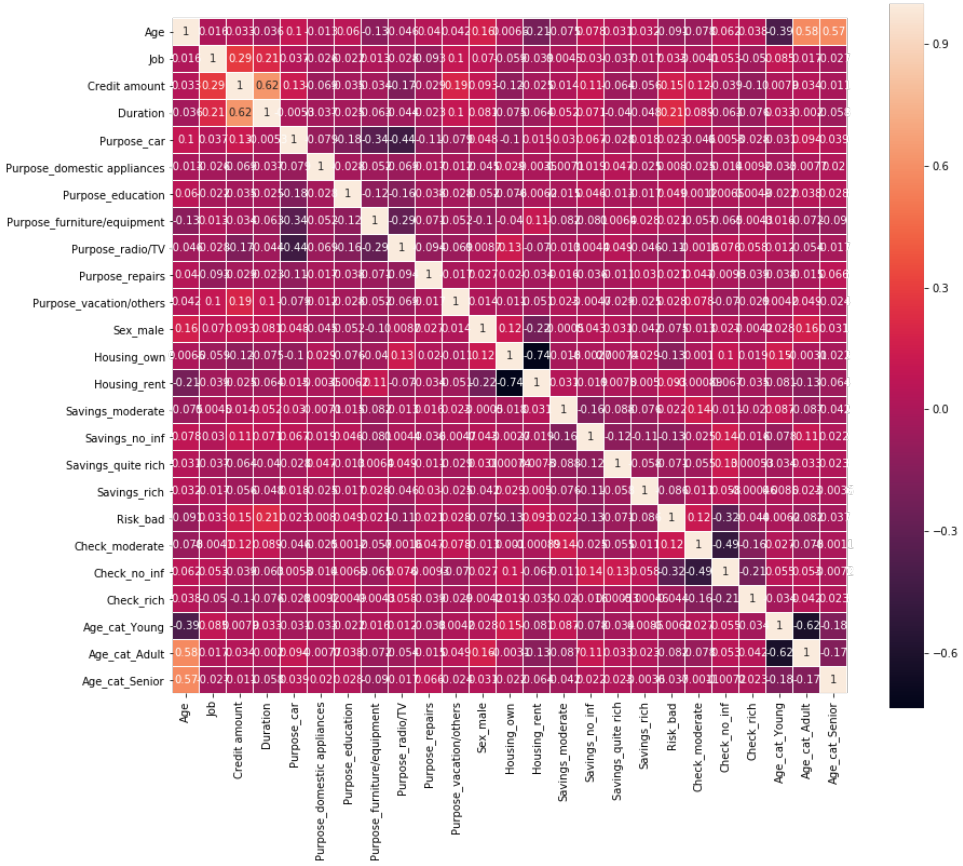


FIGURE 4.2: Holdout cross-validation

In the image above, the dataset is split into a training set and a test set. You can train the model on the training set and test it on the testing dataset. However, if you want to hyper-tune your parameters or want to select the best model, you can make a validation set like the one below.

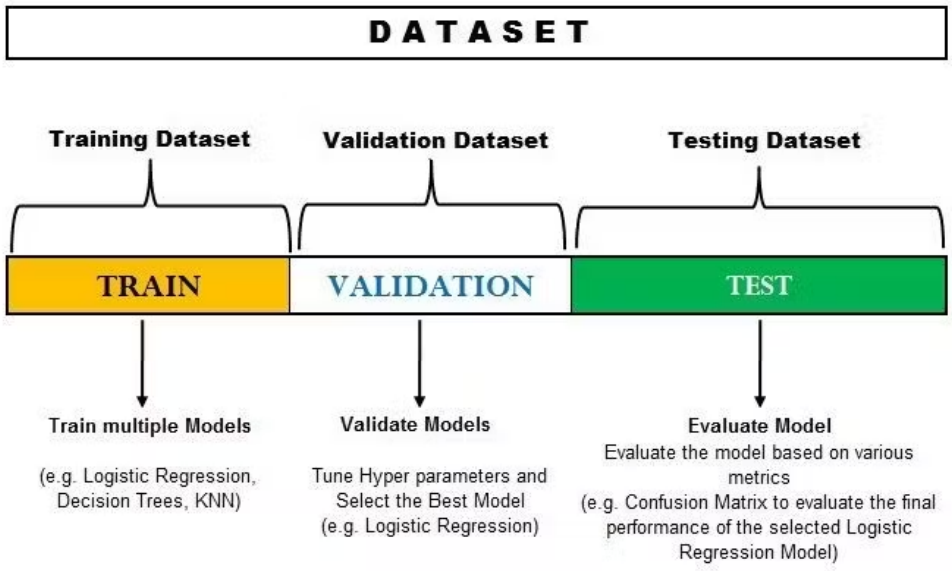


FIGURE 4.3: Holdount cross-validation

4.2.3 Stratified k-fold cross-validation

As seen above, k-fold validation can't be used for imbalanced datasets because data is split into k-folds with a uniform probability distribution. Not so with stratified k-fold, which is an enhanced version of the k-fold cross-validation technique. Although it too splits the dataset into k equal folds, each fold has the same ratio of instances of target variables that are in the complete dataset. This enables it to work perfectly for imbalanced datasets, but not for time-series data. In the example above, the original dataset contains females that are a lot less than males, so this target variable distribution is imbalanced. In the stratified k-fold cross-validation technique, this ratio of instances of the target variable is maintained in all the folds.

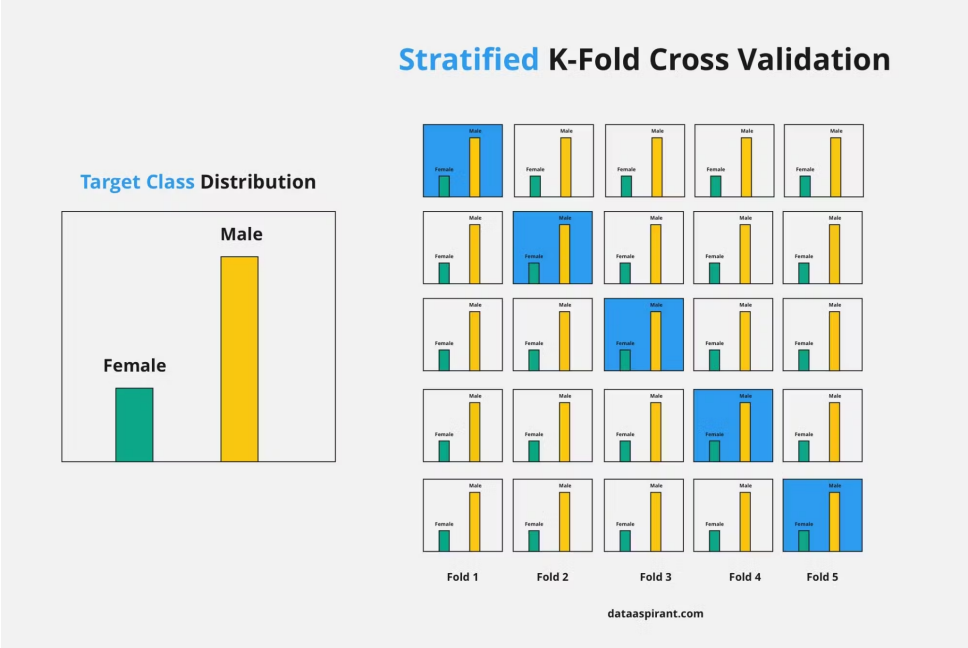


FIGURE 4.4: Stratified k-fold cross-validation

4.2.4 Leave-p-out cross-validation

An exhaustive cross-validation technique, p samples are used as the validation set and n-p samples are used as the training set if a dataset has n samples. The process is repeated until the entire dataset containing n samples gets divided on the validation set of p samples and the training set of n-p samples. This continues till all samples are used as a validation set.

The technique, which has a high computation time, produces good results. However, it's not considered ideal for an imbalanced dataset and is deemed to be a computationally unfeasible method. This is because if the training set has all samples of one class, the model will not be able to properly generalize and will become biased to either of the classes.

4.2.5 Leave-one-out cross-validation

In this technique, only 1 sample point is used as a validation set and the remaining n-1 samples are used in the training set. Think of it as a more specific case of the leave-p-out cross-validation technique with P=1.

To understand this better, consider this example: There are 1000 instances in your dataset. In each iteration, 1 instance will be used for the validation set and the remaining 999 instances will be used as the training set. The process repeats itself until every instance from the dataset is used as a validation

sample. The leave-one-out cross-validation method is computationally ex-

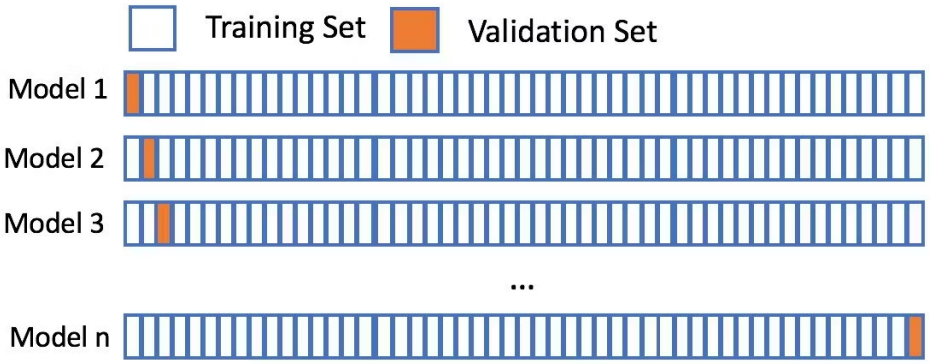


FIGURE 4.5: Leave-one-out cross-validation

pensive to perform and shoudnt be used with very large datasets. The good news is that the technique is very simple and requires no configuration to specify. It also provides a reliable and unbiased estimate for your model performance.

4.2.6 Monte Carlo cross-validation

Also known as shuffle split cross-validation and repeated random subsampling cross-validation, the Monte Carlo technique involves splitting the whole data into training data and test data. Splitting can be done in the percentage of 70-30% or 60-40% - or anything you prefer. The only condition for each iteration is to keep the train-test split percentage different.

The next step is to fit the model on the train data set in that iteration and calculate the accuracy of the fitted model on the test dataset. Repeat these iterations many times - 100,400,500 or even higher - and take the average of all the test errors to conclude how well your model performs.

For a 100-iteration run, the model training will look like this You can see that in each iteration, the split ratio of the training set and test set is different. The average has been taken to get the test errors.

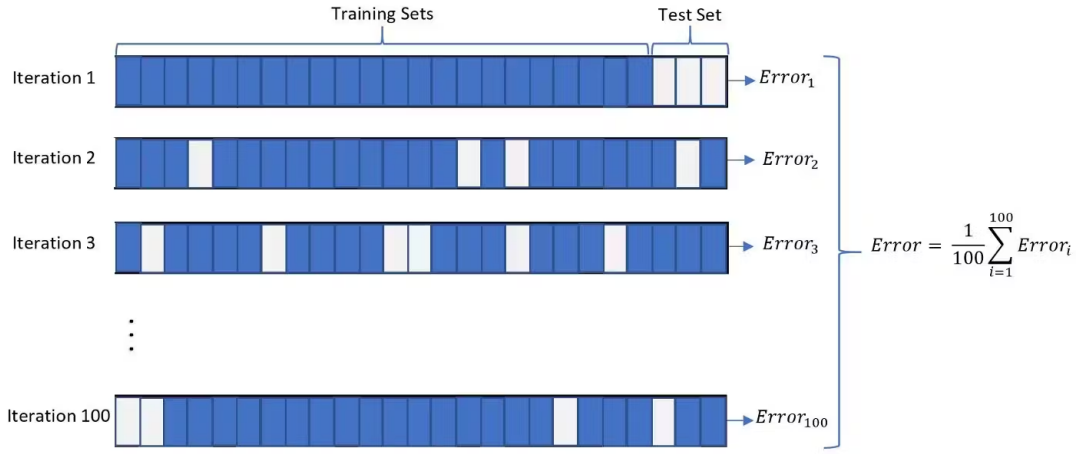


FIGURE 4.6: Monte carlo cross-validation

4.2.7 Time series (rolling cross-validation / forward chaining method)

Before going into the details of the rolling cross-validation technique, it's important to understand what time-series data is.

Time series is the type of data collected at different points in time. This kind of data allows one to understand what factors influence certain variables from period to period. Some examples of time series data are weather records, economic indicators, etc.

In the case of time series datasets, the cross-validation is not that trivial. You can't choose data instances randomly and assign them the test set or the train set. Hence, this technique is used to perform cross-validation on time series data with time as the important factor.

Since the order of data is very important for time series-related problems, the dataset is split into training and validation sets according to time. Therefore, it's also called the forward chaining method or rolling cross-validation.

To begin: Start the training with a small subset of data. Perform forecasting for the later data points and check their accuracy. The forecasted data points are then included as part of the next training dataset and the next data points are forecasted. The process goes on.

The image below shows the method.

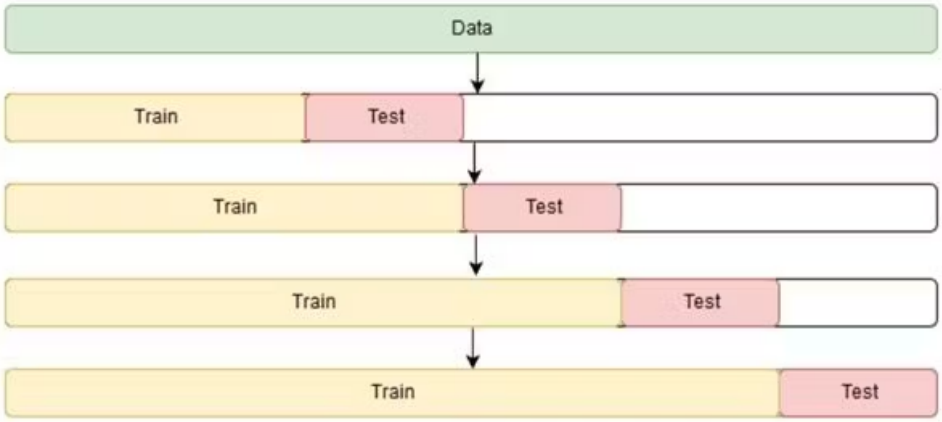


FIGURE 4.7: Time series cross validation

4.3 Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

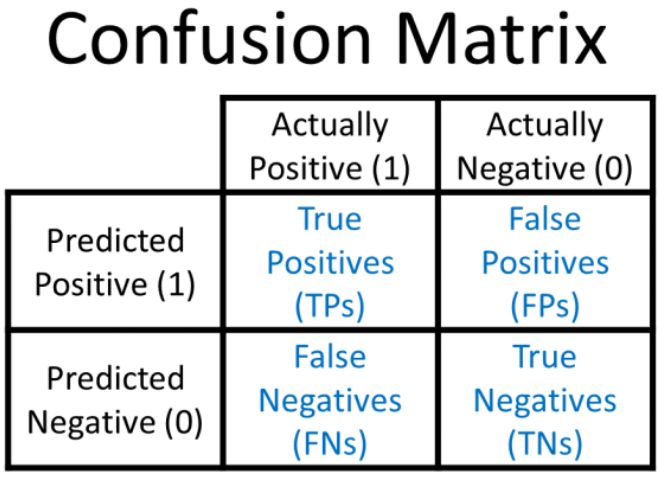


FIGURE 4.8: Confusion Matrix

The confusion matrix consists of four basic characteristics (numbers) that are used to define the measurement metrics of the classifier. These four numbers are:

1. TP (True Positive): TP represents the number of patients who have been properly classified to have malignant nodes, meaning they have the disease.

2. TN (True Negative): TN represents the number of correctly classified patients who are healthy.

3. FP (False Positive): FP represents the number of misclassified patients with the disease but actually they are healthy. FP is also known as a Type I error.

4. FN (False Negative): FN represents the number of patients misclassified as healthy but actually they are suffering from the disease. FN is also known as a Type II error.

Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated on the basis of the above-stated TP, TN, FP, and FN.

Accuracy of an algorithm is represented as the ratio of correctly classified patients (TP+TN) to the total number of patients (TP+TN+FP+FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

Precision of an algorithm is represented as the ratio of correctly classified patients with the disease (TP) to the total patients predicted to have the disease (TP+FP).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

Recall metric is defined as the ratio of correctly classified diseased patients (TP) divided by total number of patients who have actually the disease.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

F1 score is also known as the F Measure. The F1 score states the equilibrium between the precision and the recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.4}$$

4.4 AUC-ROC Curve

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

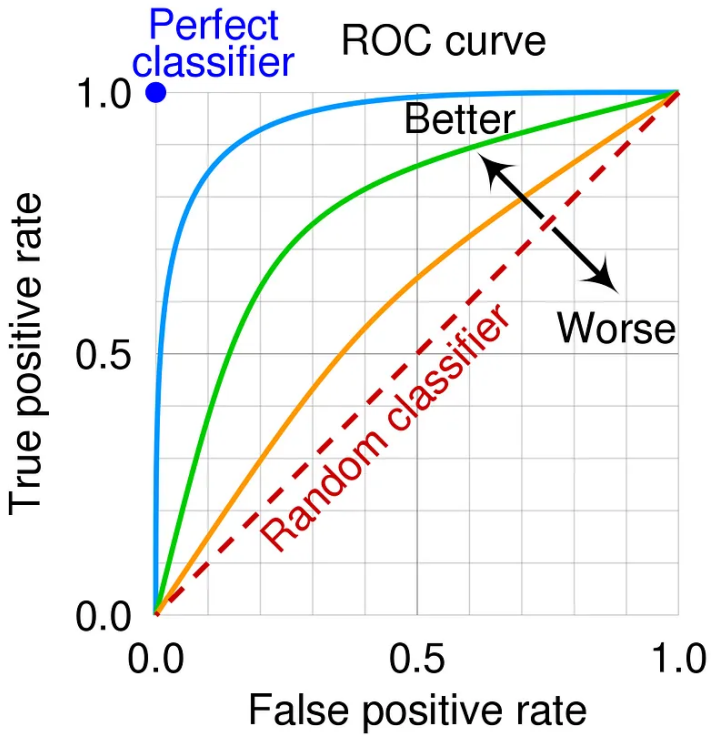


FIGURE 4.9: ROC Curve

Chapter V

Credit Derivatives

Credit derivatives are instruments that help banks, financial institutions, and debt security investors to manage their credit-sensitive investments. Credit derivatives insure and protect against adverse movements in the credit quality of the counterparty or borrower. For example, if a borrower defaults, the investor will suffer losses on the investment, but the losses can be offset by gains from the credit derivative transaction.

One might ask why both banks and investors do not utilize the well-established insurance market for their protection. The major reasons are that credit derivatives offer lower transaction costs, quicker payment, and more liquidity. Credit default swaps, for instance, often pay out very soon after the event of default; in contrast, insurances take much longer to pay out, and the value of the protection bought may be hard to determine.

Finally, as with most financial derivatives initially invented for hedging, credit derivatives can now be traded speculatively. Like other over-the-counter derivative securities, credit derivatives are privately negotiated financial contracts. These contracts expose the user to operational, counterparty, liquidity, and legal risk. From the viewpoint of quantitative modeling, we are here only concerned with counterparty risk.

One can think of credit derivatives being placed somewhere between traditional credit insurance products and financial derivatives. Each of these areas has its own valuation methodology, but neither is wholly satisfactory for pricing credit derivatives. The insurance techniques make use of historical data, as, e.g., provided by rating agencies, as a basis for valuation. This approach assumes that the future will be like the past and does not take into account market information about credit quality.

In contrast, derivative technology employs market information as a basis for valuation. Derivative securities pricing is based on the assumption of risk-neutrality, which assumes arbitrage-free and complete markets, but it is not clear whether these conditions hold for the credit market or not. If a credit event is based on a freely observable property of market prices, such as credit spreads, then we believe that conventional derivative pricing methodology may be applicable.

Credit derivatives are bilateral financial contracts that isolate specific aspects of credit risk from an underlying instrument and transfer that risk between two counterparties. By allowing credit risk to be freely traded, risk management becomes far more flexible. There are many different types of credit derivatives, but we shall only treat the most commonly used ones.

They could be classified into two main categories according to valuation, namely the replication products and the default products. The former are priced off the capacity to replicate the transaction in the money market, such as credit spread options. The latter are priced as a function of the exposure underlying the security, the default probability of the reference asset, and the expected recovery rate, such as credit default swaps.

Another classification could be along their performance as protection-like products, such as credit default options, and exchange-like products, such as total return swaps. In the next sections, we describe the most commonly used credit derivatives and illustrate simple examples. For a more elaborate introduction to the different types of credit derivatives and their use for risk management

5.1 Total Return Swaps

A total return swap (TRS), is a mean of duplicating the cash flows of either selling or buying a reference asset, without necessarily possessing the asset itself. The TRS seller pays to the TRS buyer the total return of a specified asset and receives a floating rate payment plus a margin. The total return includes the sum of interest, fees, and any change in the value with respect to the reference asset, the latter being equal to any appreciation (positive) or depreciation (negative) in the market value of the reference security. Any net depreciation in value results in a payment to the TRS seller. The margin, paid by the TRS buyer, reflects the cost to the TRS seller of financing and servicing the reference asset on its own balance sheet. Such a transaction transfers the entire economic benefit and risk as well as the reference security to another counterparty.

A company may wish to sell an asset that it holds, but for tax or political reasons may be unable to do so. Likewise, it might hold a view that a specific asset is likely to depreciate in value in the near future, and wish to short it. However, not all assets in the market are easy to short in this way. Whatever the reason, the company would like to receive the cash flows which would result from selling the asset and investing the proceeds. This can be achieved exactly with a total return swap. Let us give an example: Bank A decides to get the economic effect of selling securities (bonds) issued by a German corporation, X. However, selling the bonds would have undesirable consequences, e.g., for tax reasons. Therefore, it agrees to swap with bank B the total return on one million 7.25% bonds maturing in December 2005 in return for a six-month payment of LIBOR plus 1.2% margin plus any decrease in the value of the bonds.

Total return swaps are popular for many reasons and attractive to different market segments .One of the most important features is the facility to obtain an almost unlimited amount of leverage. If there is no transfer of physical asset at all, then the notional amount on which the TRS is paid is unconstrained. Employing TRS, banks can diversify credit risk while maintaining confidentiality of their client's financial records. Moreover, total return swaps can also give investors access to previously unavailable market assets. For instance, if an investor can not be exposed to the Latin American market for various reasons, he or she is able to do so by doing a total return swap with a counterparty that has easy access to this market. Investors can also receive cash flows that duplicate the effect of holding an asset while keeping the actual assets

away from their balance sheet. Furthermore, an institution can take advantage of another institution's back-office and documentation experience, and get cash flows that would otherwise require infrastructure, which it does not possess.

5.2 Credit Default Products

Credit default swaps are bilateral contracts in which one counterparty pays a fee periodically, typically expressed in basis points on the notional amount, in return for a contingent payment by the protection seller following a credit event of a reference security. The credit event could be either default or downgrade; the credit event and the settlement mechanism used to determine the payment are flexible and negotiated between the counterparties. A TRS is importantly distinct from a CDS in that it exchanges the total economic performance of a specific asset for another cash flow. On the other hand, a credit default swap is triggered by a credit event. Another similar product is a credit default option. This is a binary put option that pays a fixed sum if and when a predetermined credit event (default/downgrade) happens in a given time. Let us assume that bank A holds securities (swaps) of a low-graded firm X, say BB, and is worried about the possibility of the firm defaulting. Bank A pays to firm X floating rate (Libor) and receives fixed (5.5%). For protection bank A therefore purchases a credit default swap from bank B which promises to make a payment in the event of default. The fee reflects the probability of default of the reference asset, here the low-graded firm. Credit default swaps are almost exclusively inter-professional transactions, and range in nominal size of reference assets from a few millions to billions of euros. Maturities usually run from one to ten years. The only true limitation is the willingness of the counterparties to act on a credit view. Credit default swaps allow users to reduce credit exposure without physically removing an asset from their balance sheet. Purchasing default protection via a CDS can hedge the credit exposure of such a position without selling for either tax or accounting purposes. When an investor holds a credit-risky security, the return for assuming that risk is only the net spread earned after deducting the cost of funding. Since there is no up-front principal outlay required for most protection sellers when assuming a CDS position, they take on credit exposure in off-balance sheet positions that do not need to be funded. On the other hand, financial institutions with low funding costs may fund risky assets on their balance sheets and buy default protection on those assets. The premium for buying protection on such securities may be less than the net spread earned over their funding costs.

For modeling purposes, let us reiterate some basic terminology.We consider a frictionless economy with a finite horizon $[0, T]$. We assume that there exists a unique martingale measure Q making all the default-free and risky security

prices martingales, after renormalization by the money market account. This assumption is equivalent to the statement that the markets for the riskless and credit-sensitive debt are complete and arbitrage-free [89].

A filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, Q)$ is given, and all processes are assumed to be defined on this space and adapted to the filtration $\mathcal{F}_t$ (where $\mathcal{F}_t$ describes the information observable until time t). We denote the conditional expectation and the probability with respect to the equivalent martingale measure by $E_t(\cdot)$ and $Q_t(\cdot)$, respectively, given information at time t.

Let $B(t,T)$ be the time t price of a default-free zero-coupon bond paying a sure currency unit at time T. We assume that forward rates of all maturities exist; they are defined in continuous time by

$$f(t,T) = -\frac{\partial}{\partial T} \log B(t,T).$$

The default-free spot rate is defined by

$$r(t) = \lim_{T \to t} f(t,T).$$

Spot rates can be modeled directly as by Cox et al. [34] or via forward rates as in Heath et al. [90]. The money market account that accumulates return at the spot rate is defined as

$$A(t) = e^{\int_0^t r(s)ds}.$$

Under the above assumptions, we can write default-free bond prices as the expected discount value of a sure currency unit received at time T, that is,

$$B(t,T) = E_t\left[\frac{A(t)}{A(T)}\right] = E_t\left[e^{-\int_T^t r(s)ds}\right].$$

Now, let $B_e(t,T)$ be the time t price of a credit risky zero-coupon bond promising to pay a currency unit at time T. This promised payment may not be made in full if the firm is bankrupt at time T, i.e., only a fraction of the outstanding will be recovered in the event of default. Here we assume that the event premium is the difference between par and the value of a specified reference asset after default. Let again τ represent the random time at which default occurs, with a distribution function $F(t) = P[\tau \leq t]$ and $\mathbb{1}_{\{\tau<T\}}$ as

the indicator function of the event. Then the price of the risky zero-coupon bond can be written in two ways:

$$B_e(t,T) = E_t\left[e^{-\int_T^t r(s)ds}\left(\mathbb{1}_{\{\tau>T\}} + \mathrm{REC}(T)\mathbb{1}_{\{\tau<T\}}\right)\right] \tag{7.1}$$

$$= E_t\left[e^{-\int_T^t r(s)ds}\mathbb{1}_{\{\tau>T\}} + e^{-\int_\tau^t r(s)ds}\mathrm{REC}(\tau)\mathbb{1}_{\{\tau<T\}}\right]. \tag{7.2}$$

In the first expression, the recovery rate $\mathrm{REC}(T)$ is thought of as a payout received at maturity, whereas in the second expression, we think of $\mathrm{REC}(\tau)$ as the payment made at the time of default. Given the existence of the money market account, we can easily translate from one representation of the recovery to the other by

$$\mathrm{REC}(T) = \mathrm{REC}(\tau)e^{\int_T^\tau r(s)ds}.$$

A credit default swap now has a default leg and a premium leg. The present value of the contingent payment $1 - \mathrm{REC}(\tau)$ is then given by

$$A_{\mathrm{def},t} = E_t\left[e^{-\int_t^\tau r(u)du}(1-\mathrm{REC}(\tau))\mathbb{1}_{\{\tau<T\}}\right].$$

The present value of the spread payments s is given by:

$$A_{\mathrm{fee},t} = sE_t\left[e^{-\int_T^t r(u)du}\mathbb{1}_{\{\tau>T\}}\right].$$

From arbitrage-free arguments, the value of the swap should be zero when it is initially negotiated. In the course of time, its present value from the protection buyer's point of view is $A_{\mathrm{def},t} - A_{\mathrm{fee},t}$. In order to calculate the value of the CDS, it is required to estimate the survival probability, $S(t) = 1 - F(t)$, and the recovery rates $\mathrm{REC}(t)$.

Swap premiums are typically due at prespecified dates, and the amount is accrued over the respective time interval. Let $0 \le T_0 \le T_1 \le \ldots \le T_n$ denote the accrual periods of the default swap, i.e., at time T_i, $i \ge 1$ the protection buyer pays $s\Delta_i$, where Δ_i is the day count fraction for the period $[T_{i-1}, T_i]$, provided that there is no default until time T_i.

Assuming furthermore a deterministic recovery rate at default, $\mathrm{REC}(\tau) = \mathrm{REC}$, and no correlation between default and interest rates, we arrive at

$$A_{\mathrm{def},t} = (1-\mathrm{REC})\int_{T_0}^{T_n} B(T_0,u)F(du), \tag{7.3}$$

$$A_{\text{fee},t} = \sum_{i=1}^{n} s\Delta_i B(T_0, T_i)(1 - F(T_i)). \tag{7.4}$$

The integral describes the present value of the payment $(1 - \text{REC})$ at the time of default. For a default "at" time u, we have to discount with $B(T_0, u)$ and multiply with the probability $F(du)$ that default happens "around" u.

In some markets, a plain default swap includes the feature of paying the accrued premium at default, i.e., if default happens in the period (T_{i-1}, T_i), the protection buyer is obliged to pay the already accrued part of the premium payment. In this case, the value of the premium leg changes to

$$A_{\text{fee},t} = \sum_{i=1}^{n} s \left[\Delta B(T_0, T_i)(1 - F(T_i)) + \int_{T_{i-1}}^{T_i} (u - T_{i-1}) B(T_0, u) F(du) \right], \tag{7.5}$$

where the difference $u - T_{i-1}$ is according to the given day count convention. Both reduced-form models (intensity models) and structural models can in principle be applied to price default swaps. In the reduced-form model framework, the relation between the intensity process h_t and the random survival probabilities at future times t provided $\tau > t$ is given by

$$q(t, T) = P[\tau > T | \mathcal{F}_t] = E_t \left[e^{-\int_T^t h(s) ds} \right].$$

If we assume a deterministic recovery rate REC and understand the recovery as a fraction of a corresponding riskless zero with the same maturity, we can write the price for a risky zero bond as (on $\{\tau > t\}$):

$$B_e(t, T) = \text{REC}\, E_t \left[e^{-\int_T^t r(s) ds} \right] + (1 - \text{REC})\, E_t \left[e^{-\int_T^t (r(s) + h(s)) ds} \right]. \tag{7.6}$$

In the case of zero correlation between the short rate and the intensity process, both processes in the exponent would factorize when taking the expectation value. However, a truly sophisticated default swap model would call for correlated default and interest rates, which leads us beyond the scope of this presentation. Instead, we turn in the following section back to correlated defaults and their application to basket swaps.

5.3 Basket Credit Derivatives

Basket default swaps are more sophisticated credit derivatives that are linked to several underlying credits. The standard product is an insurance contract that offers protection against the event of the kth default on a basket of n underlying names, where $n \geq k$. It is similar to a plain default swap, but now the credit event to insure against is the occurrence of the kth default, not specified to a particular name in the basket.

Again, a premium, or spread, s is paid as an insurance fee until maturity or the event of the kth default. We denote by $s_{k\text{th}}$ the fair spread in a kth-to-default swap, i.e., the spread that makes the value of this swap equal to zero at inception.

If the n underlying credits in the basket default swap are independent, the fair spread $s_{1\text{st}}$ is expected to be close to the sum of the fair default probabilities of the underlying credits.
Swap spreads s_i over all underlyings $i = 1, \ldots, n$ can be summarized as follows: if the underlying credits are in some sense "totally" dependent, the first default will be the one with the worst spread; therefore,

$$s_{1\text{st}} = \max_i(s_i).$$

The question now is how to introduce dependencies between the underlying credits into our model. The concept of copulas, as introduced in Section 2.6, can be used here.Was the first to apply copulas to valuing basket swaps by generating correlated default times as random variables via a correlation model and a credit curve. For more on copulas,the literature referenced there. Denote by τ_i, $i = 1, \ldots, n$ the random default times for the n credits in the basket, and let furthermore $(F_i(t))_{t\geq 0}$ be the curve of cumulative (risk-neutral) default probabilities for credit i:

$$F_i(t) = P[\tau_i \leq t], \quad t \geq 0,$$

with $S_i(t) = P[\tau_i > t] = 1 - F_i(t)$. It is assumed that $F(t)$ is a strictly increasing function of t with $F(0) = 0$ and $\lim_{t\to\infty} F(t) = 1$. This implies the existence of the quantile function $F^{-1}(x)$ for all $0 \leq x \leq 1$.

From elementary probability theory, we know that for any standard uniformly distributed U:

$$U \sim U(0,1) \implies F^{-1}(U) \sim F. \tag{7.7}$$

This gives a simple method for simulating random variates with distribution F, i.e., random default times in our case. The cash flows in a basket default swap are functions of the whole random vector $(\tau_1, \ldots, \tau_n)$, but in order to model and evaluate this basket swap, we need the joint distribution of the τ_i's:

$$F(t_1, \ldots, t_n) = P[\tau_1 \leq t_1, \ldots, \tau_n \leq t_n].$$

Similarly, we define the multivariate survival function S by

$$S(t_1, \ldots, t_n) = P[\tau_1 > t_1, \ldots, \tau_n > t_n].$$

Note that

$$S_i(t_i) = S(0, \ldots, 0, t_i, 0, \ldots, 0),$$

and

$$S(t_1, \ldots, t_n) \neq 1 - F(t_1, \ldots, t_n).$$

We exploit again the concept of copula function where, for uniform random variables $U_1, U_2, \ldots, U_n$,

$$C(u_1, u_2, \ldots, u_n) = P[U_1 \leq u_1, U_2 \leq u_2, \ldots, U_n \leq u_n]$$

defines a joint distribution with uniform marginals. The function $C(u_1, u_2, \ldots, u_n)$ is called a Copula function. Remember that $U_i = F_i(\tau_i)$ admits a uniform distribution on the interval $[0, 1]$; so, the joint distribution of $(\tau_1, \ldots, \tau_n)$ can be written as:

$$F(t_1, \ldots, t_n) = C(F_1(t_1), \ldots, F_n(t_n)). \tag{7.8}$$

Hence, the Copula function introduces a mutual correlation by linking univariate marginals to their full multivariate distribution, thereby separating the dependency structure C, i.e., the ingredients are some credit curve for each credit as marginal distribution functions for the default times and a suitably chosen copula function. Observe that by Sklar's theorem (Section 2.6), any joint distribution can be reduced to a copula and the marginal distributions, although it may be difficult to write down the copula explicitly.

One of the most elementary copula functions is the multivariate normal distribution:

$$C(u_1, u_2, \ldots, u_n) = N_n\left(N^{-1}(u_1), N^{-1}(u_2), \ldots, N^{-1}(u_n); \Gamma\right), \tag{7.9}$$

where N_n is the cumulative multivariate normal distribution with correlation matrix Γ and N^{-1} is the inverse of a univariate normal distribution. Clearly, there are various different copulas generating all kinds of dependencies, and the choice of the copula entails a significant amount of model risk [68, 70]. The advantage of the normal copula, however, is that, as we have seen in Chapter 2, it relates to the latent variable approach to model dependent defaults.

Assume that the default event of credit i up to time T is driven by a single random variable r_i (ability-to-pay variable) being below a certain threshold $c_i(T)$:

$$\tau_i < T \iff r_i < c_i(T).$$

If the Z_i's admit a multivariate standard normal distribution with correlation matrix Γ, then to be consistent with our given default curve, we set $c_i(T) = N^{-1}(F_i(T))$. The pairwise joint default probabilities are now given in both representations by:

$$P[\tau_i \leq T, \tau_j \leq T] = P[r_i \leq c_i(T), r_j \leq c_j(T)] = N_2[N^{-1}(F_i(T)), N^{-1}(F_j(T)); \Gamma_{e_{ij}}].$$

. We see that these probabilities (7.10) only coincide with those from the normal copula approach (7.8) and (7.9) if the asset correlation matrix Γ_e and the correlation matrix Γ in the normal copula are the same. However, since the asset value approach can only model defaults up to a single time horizon

T, calibration between the two models can only be conducted for one fixed horizon. Thus, we observe again that the factor model approach to generate correlated defaults based on standard normal asset returns is tantamount to a normal copula approach.

Remark: Analogously to the default distribution, we can apply Sklar's theorem to the survival function. When S is a multivariate survival function with margins $S_1, \ldots, S_n$, there exists a copula representation:

$$S(t_1, \ldots, t_n) = C^S(S_1(t_1), \ldots, S_n(t_n)). \tag{7.11}$$

There is an explicit, albeit rather complex, relation between the survival copula C^S and the distribution copula C [76]. In the two-dimensional case, we obtain:

$$C^S(u_1, u_2) = S(S_1^{-1}(u_1), S_2^{-1}(u_2)) = S(t_1, t_2) = 1 - F_1(t_1) - F_2(t_2) + F(t_1, t_2) = S_1(t_1) + S_2(t_2)$$

This simplifies to:

$$C^S(u_1, u_2) = u_1 + u_2 - 1 + C(1 - u_1, 1 - u_2).$$

It can be easily shown that C^S is indeed a copula function. At this point, we note that a copula is radially symmetric if and only if $C = C^S$ (proof [76]). The normal copula is radially symmetric. Thus, in two dimensions, we find:

$$C^S(u_1, u_2) = u_1 + u_2 - 1 + C(1 - u_1, 1 - u_2) = u_1 + u_2 - 1 + N_2\left(N^{-1}(1 - u_1), N^{-1}(1 - u_2); \Gamma\right)$$

This property is particularly interesting for computational purposes, as in the radially symmetric case, it is equivalent to work with either the distribution copula or the survival copula.

5.4 Credit Spread Products

Credit spread is the difference between the yield on a particular debt security and a benchmark yield, usually on a government bond. Credit spread options (CSO) can be based on various types of credit spreads, such as the asset-swap spread, the default-swap spread, and the yield spread (**?**). These options allow investors to express a directional view on credit spreads or to hedge risk.

In the case of options on CDS spreads, one speaks of a *credit default swaption*. The "put/call" terminology is sometimes confusing here. Credit default swaptions use the lingo of *payer* and *receiver* instead (similar to interest rate swaptions). A *payer option* is the right to buy credit default protection at a pre-specified strike level K on a future date, with a payoff equal to:

$$\max(S(T) - K, 0)$$

at maturity, where $S(T)$ denotes the credit spread at time T.
A payer option is both a put on credit quality – a bet that credit will deteriorate – and a call on spreads – a bet that spreads will widen. Likewise, a *receiver option* is the right to sell credit default protection at a pre-specified strike level K on a future date, with a payoff equal to:

$$\max(K - S(T), 0).$$

A receiver option is both a call on credit (the buyer profits when credit quality improves) and a put on spreads.

One of the key characteristics of these products is that the return is not dependent on a specific credit event. It merely depends on the value of one reference credit spread against another. If the credit rating of the reference asset owner declines (increasing the default probability), the credit spread widens, and vice versa.

A debt issuer can use payer options (a put on credit) to hedge against a rise in the average credit spread. On the other hand, a financial institution holding debt securities can purchase receiver options (a call on credit) to hedge against a fall in the credit spread.

Credit spread derivatives are priced using various models. One approach is to model the spread itself as an asset price, which offers simplicity. Longstaff and Schwartz (**?**) developed a simple framework for pricing credit spread derivatives, which we summarize below. Their model captures key empirical properties of observed credit spreads and provides closed-form solutions for call and put CSOs.

Let x denote the logarithm of the credit spread, i.e., $x_t = \log(S(t))$. The dynamics of x are described by the stochastic differential equation (SDE):

$$dx = (a - bx)dt + s\,dB_1,$$

where a, b, and s are parameters, and B_1 is a Wiener process. This implies that changes in x are mean-reverting and homoscedastic, which aligns with empirical observations.

We assume the default-free term structure is determined by a one-factor model (**?**), given by:

$$dr = (\alpha - \beta r)dt + \sigma dB_2,$$

where α, β, and σ^2 are parameters, and B_2 is a Wiener process. The correlation coefficient between dB_1 and dB_2 is $\hat{\rho}$. We assume that market prices of risk are incorporated into a and α, so both a and α are risk-adjusted parameters, consistent with Vasicek (**?**) and Longstaff and Schwartz.
The risk-adjusted process for x is given by Longstaff and Schwartz (**?**) as:

$$dx = \left(a - bx - \frac{\rho\sigma s}{\hat{\beta}}\left(1 - e^{-\beta(T-t)}\right)\right)dt + s\,dB_1. \tag{7.14}$$

This SDE can be solved by making a change of variables and then integrating. The resulting solution implies that x_T is conditionally normally distributed with respect to (7.14) with mean μ and variance η^2, where:

$$\mu = e^{-bT}x + \frac{1}{b}\left(a - \frac{\rho\sigma s}{\hat{\beta}}\right)\left[1 - e^{-bT}\right] + \frac{\rho\sigma s}{\hat{\beta}(b+\beta)}\left[1 - e^{-(b+\beta)T}\right],$$

$$\eta^2 = \frac{s^2\left(1 - e^{-2bT}\right)}{2b}.$$

Note that as $T \to \infty$, the values of μ and η^2 converge to fixed values, and the distribution of x_T converges to a steady-state stationary distribution.

With this framework, we can find the price of a European call credit spread option (CSO). Let $C(x, r, T)$ denote the value of the option. The payoff function for this option is simply $H(x) = \max(e^x - K, 0)$. The closed-form solution for the call CSO is given by:

$$C(x, r, T) = p(r, T) \left[e^{\mu + \eta^2/2} N(d_1) - K N(d_2) \right],$$

where $N(\cdot)$ is the cumulative standard normal distribution, $p(r, T)$ is a riskless discount bond, and:

$$d_1 = \frac{-\log(K) + \mu + \eta^2}{\eta}, \quad d_2 = d_1 - \eta.$$

The value of a European put CSO is:

$$P(x, r, T) = C(x, r, T) + p(r, T) \left[K - e^{\mu + \eta^2/2} \right].$$

The option formula has some similarities with the Black-Scholes option pricing formula. However, the value of a call option can be less than its intrinsic value, even when the call is slightly in the money. This surprising result is due to the mean reversion of the credit spreads. When the spread is above the long-run mean, it is expected to decline over time. This cannot happen in the Black-Scholes model because the underlying asset must appreciate at the riskless rate in the risk-neutral.The delta for a call is always positive, as in the Black-Scholes (B-S) framework, but the delta of a credit spread option (CSO) call decreases to zero as the time until expiration increases. A change in the current credit spread is heavily outweighed by the effects of mean reversion if the expiration date of the call is far in the future.

An investor may combine a payer and a receiver option to create a straddle, which is a bet on spread volatility. The buyer of the straddle makes money if spreads either widen or tighten by more than the breakeven level. Investors can also insure against rising credit spreads by buying a payer option and reduce the cost by selling a receiver option.

In a credit spread forward (CSF), counterparty A pays at time T a pre-agreed fixed payment and receives the credit spread of the reference asset at time T.

Conversely, counterparty B receives the fee and pays the credit spread. The fixed payment is chosen at time $t < T$ to set the initial value of the credit spread forward to zero. The credit spread forward can also be structured around the relative credit spread between two different defaultable bonds. Credit spread forwards can be combined to a credit spread swap, in which one counterparty pays periodically the relative credit spread, $S_1(t) - S_2(t)$, to the other.

5.5 Credit-Linked Notes

Credit-linked notes exist in various forms in the credit derivatives market;. In its most common form, a credit-linked note (CLN) is a synthetic bond with an embedded default swap.
CLNs are initiated in several ways. In the following we outline four examples of typical CLN structures. The first case we present is the situation of an (institutional) investor who wants to have access to a credit exposure (the reference asset) for which by policy, regulation, or other reasons he has no direct access. In such cases, a CLN issued by another institution (the issuer) which has access to this particular credit exposure offers a way to evade the problems hindering the investor to purchase the exposure he is interested in. The issuer sells a note to the investor with underlying exposure equal to the face value of the reference asset. He receives the face value of the reference asset as cash proceeds at the beginning of the transaction and in turn pays interest, including some premium for the default risk, to the investor. In case the reference asset experiences a credit event, the issuer pays to the investor the recovery proceeds of the reference asset. The spread between the face value and the recovery value of the reference asset is the investor's exposure at risk. In case no credit event occurred during the lifetime of the reference note, the issuer pays the full principal back to the investor. So in this example one could summarize a CLN as a synthetic bond with an embedded default swap. In our second example, an investor, who has no access to the credit derivatives market or is not allowed to do off-balance sheet transactions, wants to invest in a credit default swap, selling protection to the owner of some reference asset. This can be achieved by investing in a CLN in the same way as described in our first example. Note that from the investor's point of view the CLN deal differs from a default swap agreement by the cash payment made upfront. In a default swap, no principal payments are

exchanged at the beginning. Another common way to set up a CLN is protection buying. Assume that a bank is exposed to the default risk of some reference asset. This could be the case by means of an asset on the balance sheet of the bank or by means of a situation where the bank is the protection seller in a credit default swap. The bank can now issue a CLN to some investor who pays the exposure of the reference asset upfront in cash to the bank and receives interest, including some premium reflecting the riskiness of the reference asset, during the lifetime of the note. If the reference asset defaults, the bank suffers a loss for its balance sheet asset (funded case) or has to make a contingent payment.
For the default swap (unfunded case), the CLN then compensates the bank for the loss, such that the CLN functions as an insurance. In this example, the difference between a CLN and just another default swap arises from the cash proceeds the bank receives upfront from the CLN investor. As a consequence, the bank is not exposed to the counterparty risk of the protection selling investor. Therefore, the credit quality of the investor is of no relevance2 . The proceeds from the CLN can be kept as a cash collateral or be invested in high-quality collateral securities, so that losses on the reference asset will be covered with certainty.

Chapter VI

Application of credit risk

6.1 Introduction

The original dataset contains 1000 entries with 20 categorial/symbolic attributes prepared by Prof. Hofmann. In this dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of attributes. The link to the original dataset can be found below.
It is almost impossible to understand the original dataset due to its complicated system of categories and symbols. Thus, I wrote a small Python script to convert it into a readable CSV file. Several columns are simply ignored, because in my opinion either they are not important or their descriptions are obscure. The selected attributes are:
Age (numeric)
Sex (text: male, female)
Job (numeric: 0 - unskilled and non-resident, 1 - unskilled and resident, 2 - skilled, 3 - highly skilled)
Housing (text: own, rent, or free)
Saving accounts (text - little, moderate, quite rich, rich)
Checking account (numeric, in DM - Deutsch Mark)
Credit amount (numeric, in DM)
Duration (numeric, in month)
Purpose(text: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others
Risk (Value target - Good or Bad Risk)

6.2 Distribution of Variables

In this section i presend the distribution of each variable.

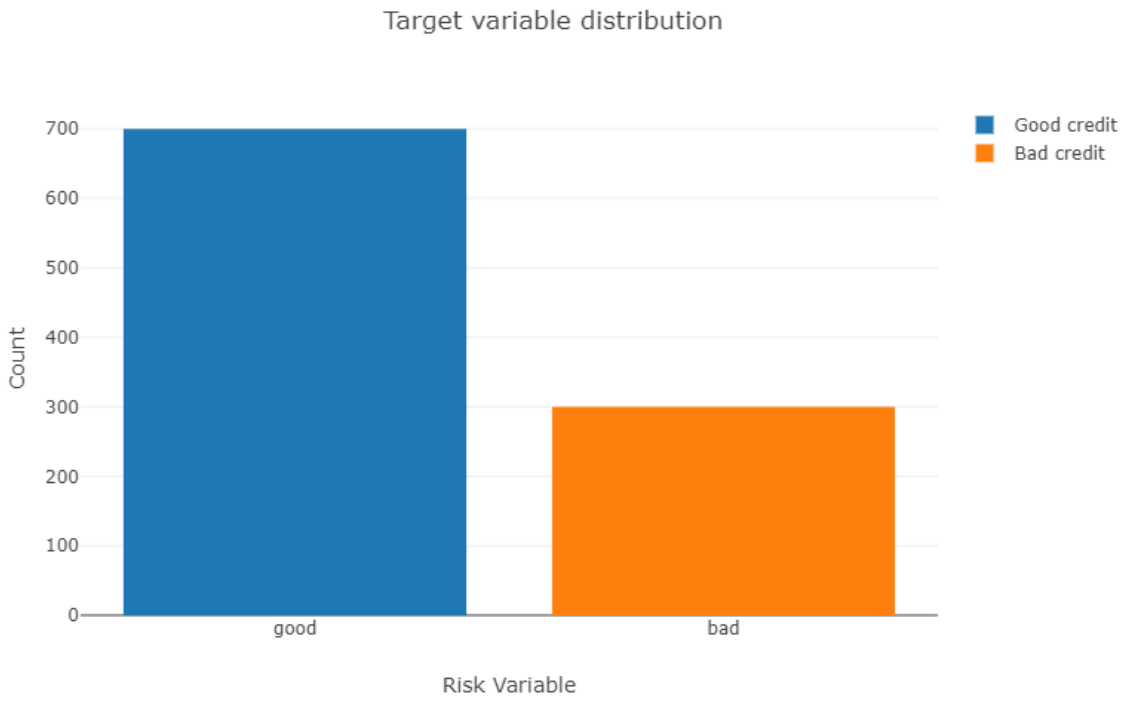


FIGURE 6.1: histogram 1

In this figure we see that most of persons,who takes a credit by bank have good credit
In this histogram we see that the age distributio for bad and good credit target.For the two categories we see that we have a left skew distribution and also overall

In the box plots we see the credit amount by age,we see that the student and young have smaller variability than adult who are bad credit. Also we see that in young and student we have a lot of points away,which means are cases about the usually credit amount that they have take.

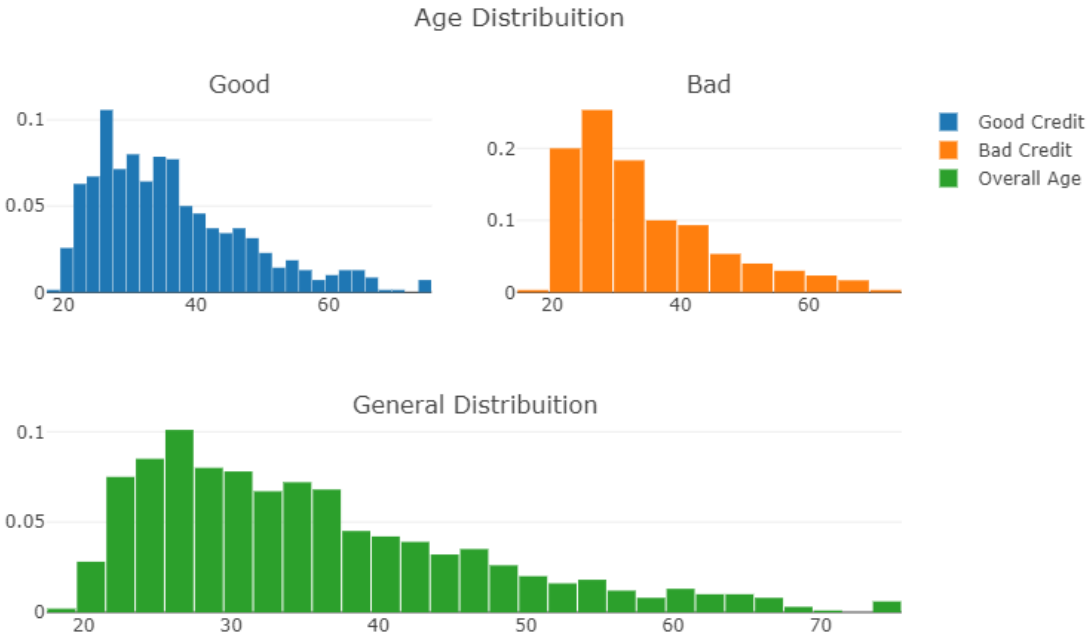


FIGURE 6.2: Histogram 2

TABLE 6.1: Model Performance Metrics

Model	Mean (SD)
LR	0.387574 (0.077970)
LDA	0.428815 (0.093162)
KNN	0.263130 (0.035560)
CART	0.547962 (0.114109)
NB	0.596300 (0.081365)
RF	0.301979 (0.050865)
svm	0.182230 (0.089)
XGB	0.42 (0.11)

Almost all models shows a low value to recall. We can observe that our best results was with CART, NB and XGBoost. I will implement some models and try to do a simple Tunning on them

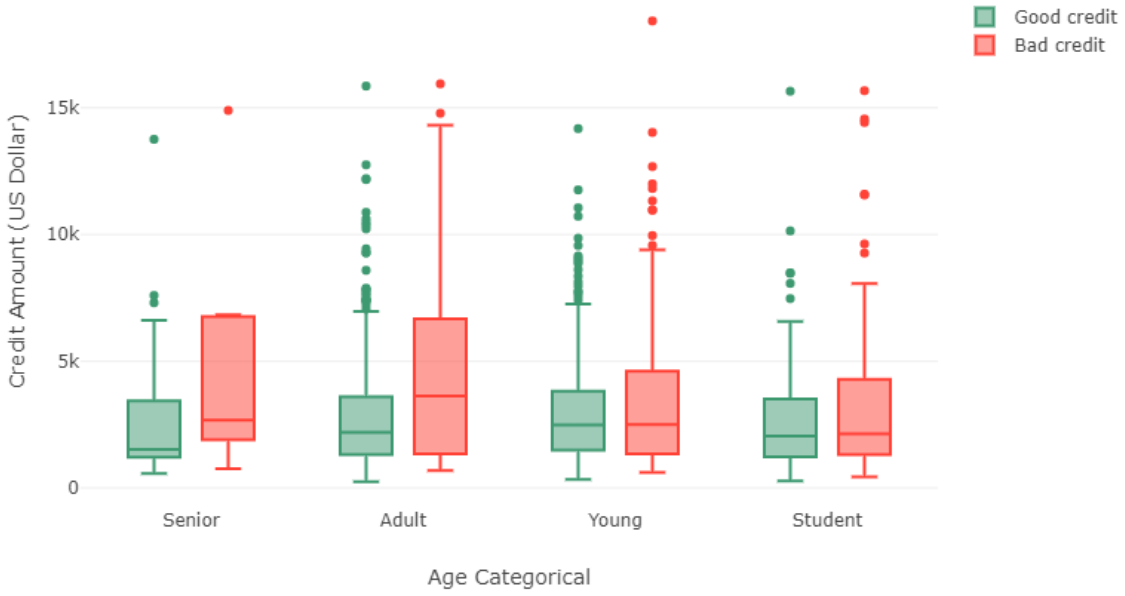


FIGURE 6.3: Box plots

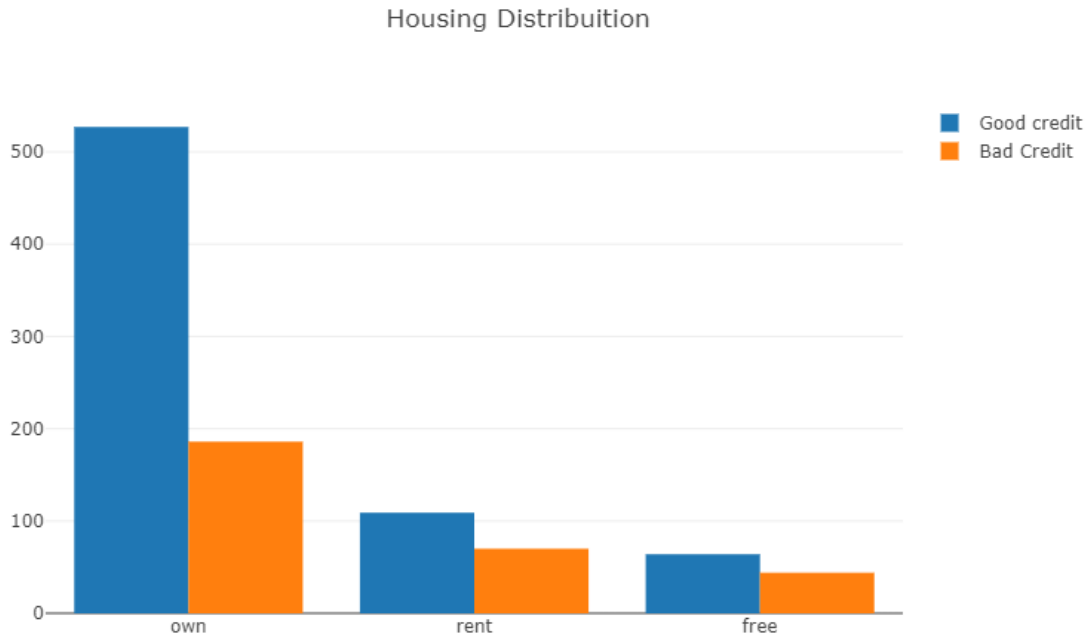


FIGURE 6.4

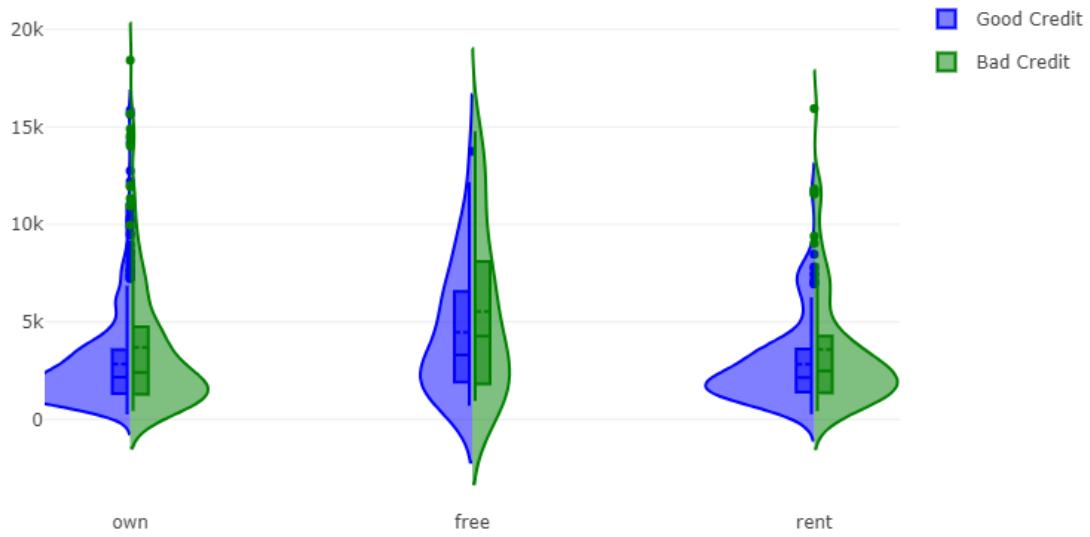


FIGURE 6.5



FIGURE 6.6

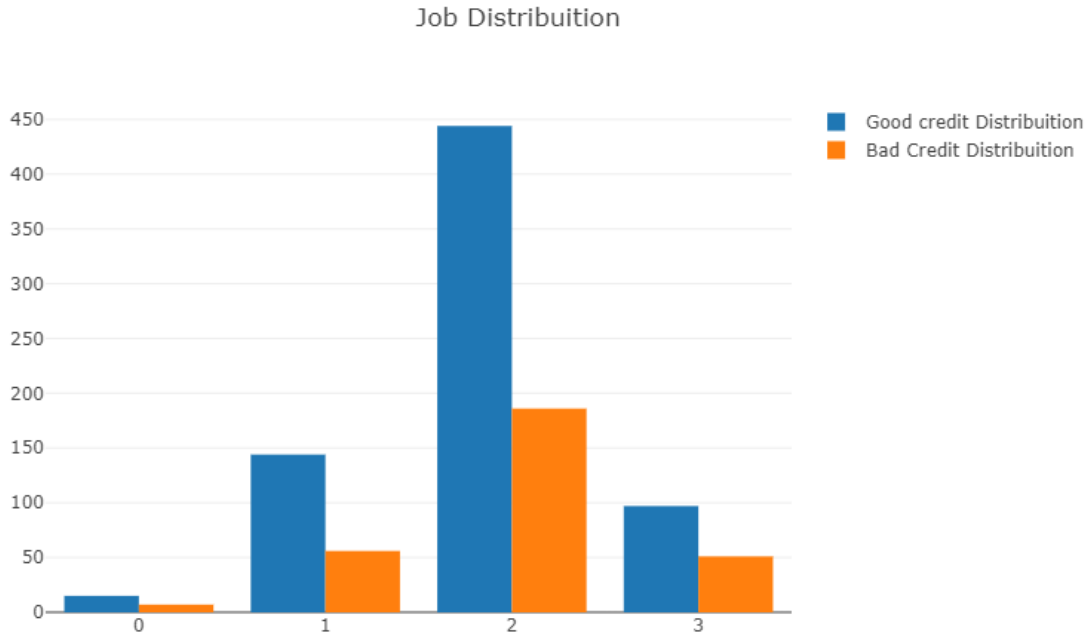


FIGURE 6.7

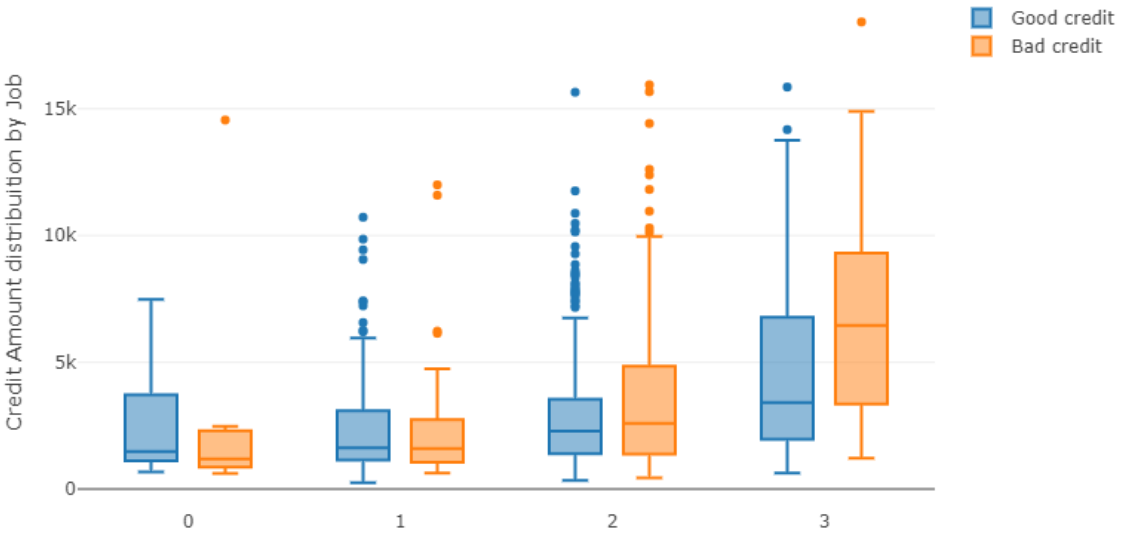


FIGURE 6.8

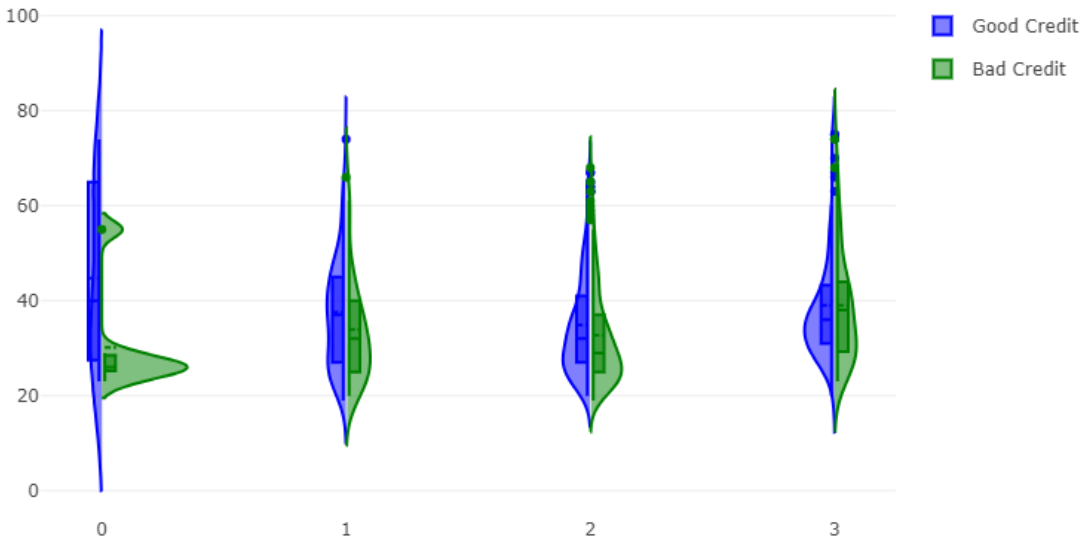


FIGURE 6.9

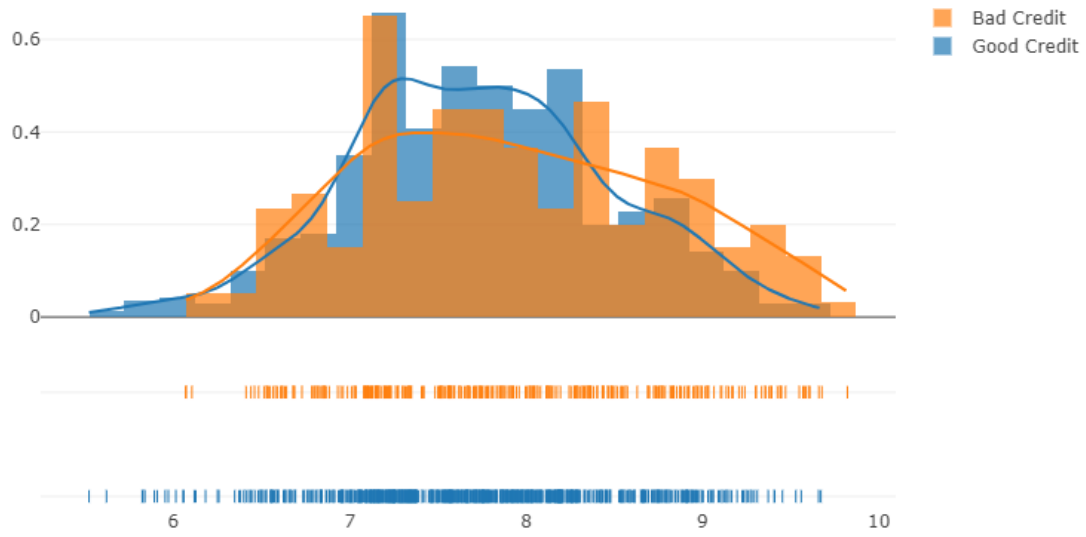


FIGURE 6.10



FIGURE 6.11: Enter Caption

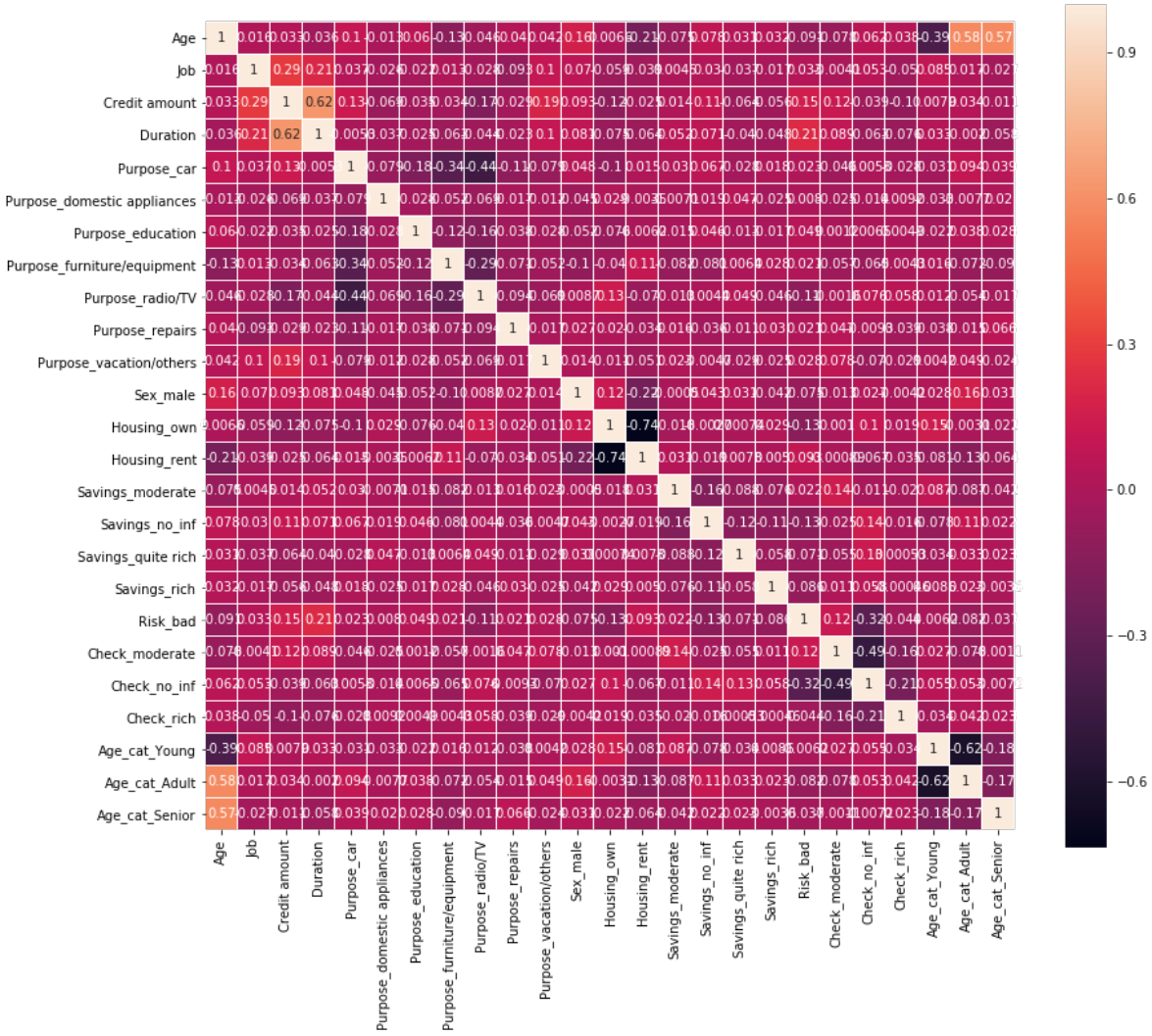


FIGURE 6.12: corrplot

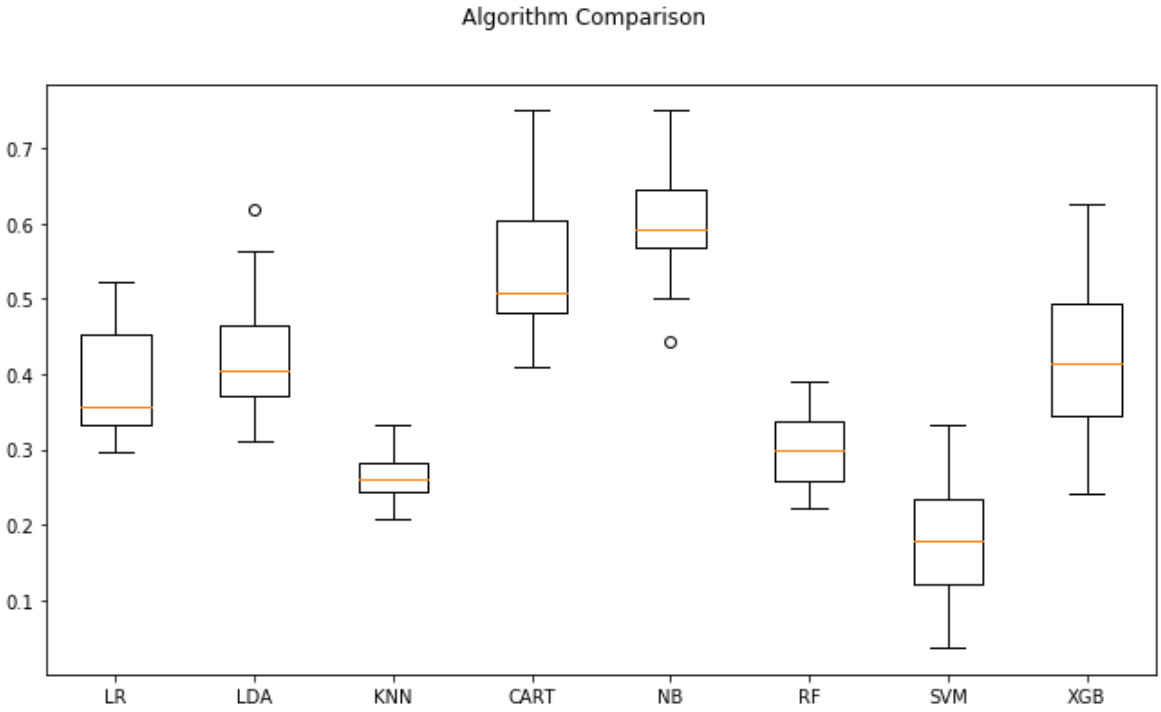


FIGURE 6.13: Enter Caption

Bibliography

[1] Jarrow, R.A. and Turnbull, S.M. (1995). *Pricing Derivatives on Financial Securities Subject to Credit Risk*. Journal of Finance, 50(1), pp. 53-85.

[2] Duffie, D. and Singleton, K.J. (1999). *Modeling Term Structures of Defaultable Bonds*. Review of Financial Studies, 12(4), pp. 687-720.

[3] Merton, R.C. (1974). *On the Pricing of Corporate Debt: The Risk Structure of Interest Rates*. Journal of Finance, 29(2), pp. 449-470.

[4] O'Kane, D. (2008). *Modelling Single-name and Multi-name Credit Derivatives*. Wiley.

[5] Bluhm, C., Overbeck, L., and Wagner, C. (2010). *An Introduction to Credit Risk Modeling*. Chapman and Hall/CRC.