DISTILLATION LEARNING

Enhancing IRB Models for PD Estimation

AUTHOR
Omar ELGHAFFOULI
DATA SCIENTIST

SUPERVISOR
Imen FOURATI
MRM EXPERTE LEADER

FOLLOWED-UP BY
Zineb BAROUDI and Nada AMINI
MRM ANALYSTS

RISQ/MRM
FROM 4/3/2023 To 9/29/2023

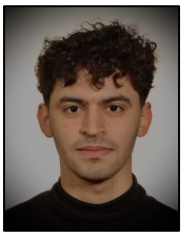
Société Générale : 17, Cours Valmy, 92000, Puteaux, France

THE FUTURE IS YOU SOCIETE GENERALE | Centrale CASABLANCA ENSAE

FILIERE: Data Science, Statistics and Machine Learning

OPTION : Mobilité 3A, M2 ENSAE Paris

PROMOTION : 2023



NOM : ELGHAFFOULI **PRENOM :** OMAR **SUJET DE STAGE :** DISTILLATION LEARNING

Résumé précis du rapport de PFE en Français *:*

Au cours de ce stage, nous avons exploré le concept de **l'apprentissage par distillation**, une puissante technique en *Machine Learning*. Elle possède de nombreuses applications, telles que **la compression des modèles** pour réduire leur complexité, les rendant ainsi adaptés à une inférence plus rapide et au déploiement sur des systèmes embarqués aux ressources limitées. De plus, l'apprentissage par distillation possède des applications dans **l'interprétabilité des modèles en *Machine Learning*** en transformant les modèles *« black-box »* en modèles interprétables, améliorant ainsi leur interprétabilité globale. Notre recherche a impliqué une vaste revue de la littérature sur les techniques de la distillation, englobant les concepts, les *Frameworks* d'entraînement, les outils existants et les applications pertinentes. Nous avons ensuite appliqué ces techniques pour évaluer leur pertinence dans le secteur bancaire, en particulier pour améliorer les modèles d'estimation de la probabilité de défaut dans le cadre de la réglementation bâloise II en mettant l'accent sur le risque de crédit. Grâce à cette application, nous avons observé des améliorations notables des performances, notamment grâce à des méthodes telles que la distillation « ***Response-based*** » ou la « ***X-Distillation*** ». Par la suite, nous avons élargi notre analyse en testant d'autres approches de la distillation comme la distillation *« **Feature-based** »* et la distillation *« **Multi-teacher** »* mettant en lumière leurs contributions et leurs limites en utilisant les données de *Lending Club*. Bien que les techniques de distillation trouvent des applications variées dans des domaines tels que la reconnaissance d'images et le traitement du langage naturel, notre principal objectif était d'élucider les principes fondamentaux qui restent applicables dans divers domaines.

MOTS-CLÉS: KNOWLEGDE DISTILLATION, MACHINE LEARNING, XAI, MODEL COMPRESSION, MODELS PERFORMANCE ENHANCING, CREDIT RISK

DATE DE SOUTENANCE PFE : 25 octobre 2023

SOCIETE GENERALE

Centrale CASABLANCA

CONFIDENTIALITY DISCLAIMER

This present document is confidential. It cannot be communicated outside in paper format nor distributed in electronic format.

AKNOWLEDGEMENT

This internship was a great opportunity to start a data science professional career. I had the chance to deepen my knowledge in many topics related to machine leaning such as mathematics, data science and statistics while working on one of the most advanced and recent topics in the research field which is distillation learning. I had also the opportunity to explore other topics related to risk management in the banking sector, specifically credit risk. Exchanges with the members of MRM department were very instructive and helped me to improve the understanding and the functioning of the risk department.

I would like to thank warmly my internship supervisors Imen FOURATI, Zineb BAROUDI and Nada AMINI, for their guidance and the constructive exchanges we had. I had enjoyed working together. They did the best they can to make my internship a rich learning journey as they helped me to focus on significant points and improve the quality of my work. Finally, I would like to thank the whole RISQ/MRM team for the kindness and for facilitating my integration in the work environment.

EXECUTIVE SUMMARY

In this internship, we delved into the concept of **distillation learning**, a potent machine learning technique. It serves multiple purposes, such as **compressing models** to reduce complexity, making them suitable for faster inference and deployment on resource-constrained edge devices. Furthermore, distillation learning plays a vital role in the realm of **explainable artificial intelligence** (XAI), aiding in the transformation of black-box models into interpretable student models, thus enhancing global interpretability. The research journey involved an extensive literature review on knowledge distillation techniques, encompassing concepts, frameworks, existing tools, and relevant applications. We then applied these techniques to assess their relevance in the banking sector applications focusing on credit risk management, aiming at improving the probability of default estimation models. Through this application, we observed notable performance enhancements, particularly through methods like **response-based distillation**, **adversarial knowledge distillation** and **X-Distillation**. Subsequently, we expanded the analysis by testing **feature-based** and **multi-teacher distillation** approaches on the *Lending Club* dataset, shedding light on their contributions and limitations. While distillation techniques find wide-ranging applications in domains like image and speech recognition, the primary in this work focus was on elucidating the foundational principles that remain applicable across various domains.

KEYWORDS: KNOWLEGDE DISTILLATION, MACHINE LEARNING, XAI, MODEL COMPRESSION, MODELS PERFORMANCE ENHANCING, CREDIT RISK

TABLE OF CONTENTS

INTRODUCTION

Context

Knowledge distillation is a machine learning concept introduced the first time in 2015 by Hinton and his collaborators in the article titled “Distilling the Knowledge in a Neural Network” (Hinton, et al., 2015). The key idea was originally introduced by Caruana and his collaborators in 2006 as a model compression technique that uses a fast and compact model to approximate the empirical function learned by a slower, larger and a better performing model (Caruana, et al., 2006).

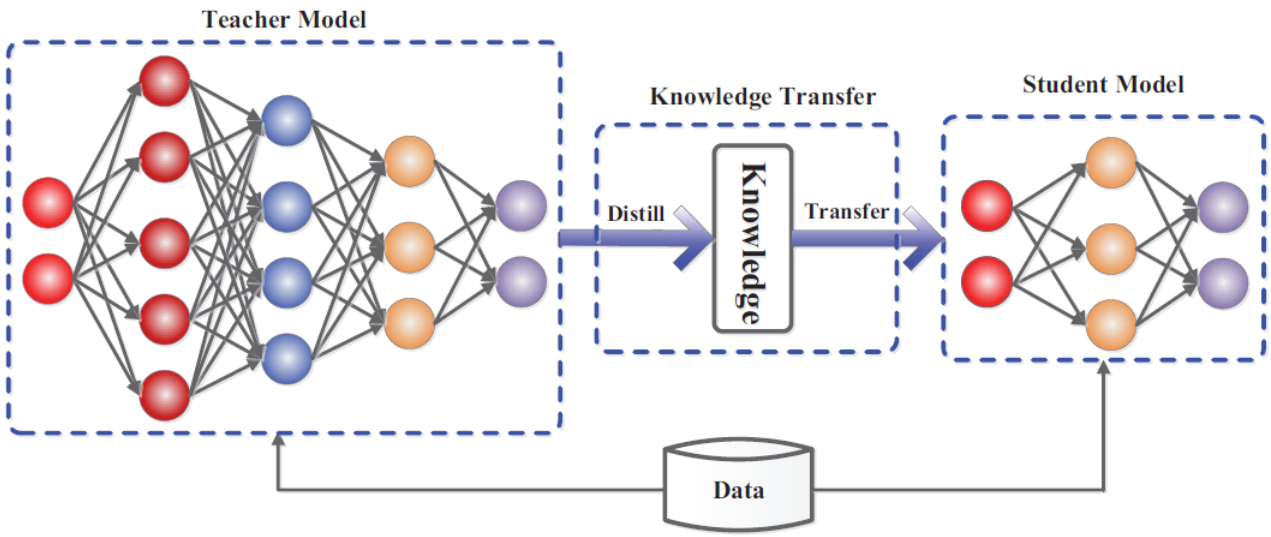


Figure 1. the generic teacher-student framework for knowledge distillation *(Gou, et al., 2021)*

In essence, knowledge distillation involves training a smaller model, often referred to as the student model, to replicate the behavior of a larger, more complex model known as the teacher model. This process aims to imbue the student model with the comprehensive insights and knowledge encapsulated within the teacher model.

Distillation learning has several useful applications. In **explainable machine learning**, Large complex models such as neural networks, bagging, boosting, or stacking models are distilled into simple explainable models such as decision-trees or regression models. In medical imaging, complex CNNs teachers’ models train interpretable decision-tree students’ models. Insights from the student model's transparent predictions help doctors in disease diagnosis. Knowledge distillation training is also used to improve models’ performance. A decision-tree model trained by distillation of a complex teacher generally perform better than a decision-tree with standard training.

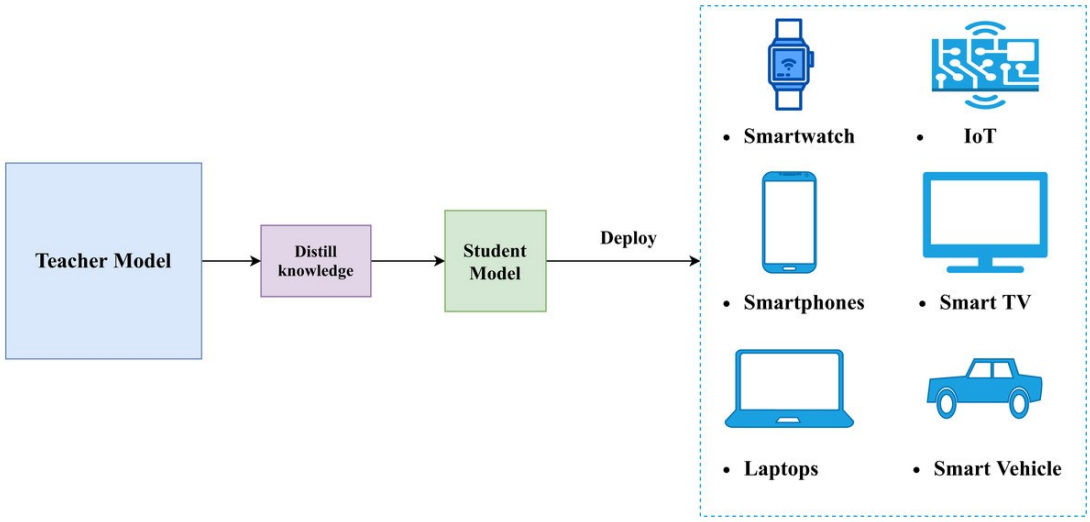


Figure 2. Knowledge Distillation for Device Model Embedding

Also, in scenarios where resource constraints pose challenges, knowledge distillation offers a pragmatic solution through model **compression** to deploy large models on resource-limited devices and accelerate inference in real-time applications and therefore reduce memory and computational requirements.

Mission

Considering the relevance of the powerful machine learning technique of distillation learning . The objective of the internship is to study some of its applications for model risk management (RISQ/MRM) at Société Générale[1].

This internship offers an exceptional opportunity to gain knowledge on distillation learning, specifically in the domain of model risk management not only theoretical insights but also hands-on involvement in real-world projects that contribute to the bank's resilience and compliance with regulatory standards.

In conclusion, the integration of distillation learning techniques within the model risk management landscape showcases SG's commitment to innovation and excellence. The internship promises a rich and dynamic learning experience, fostering a deeper understanding of both distillation learning and its role in fortifying the bank's risk management framework.

[1] *Please refer to the appendix for information about the RISQ/MRM department*

CHAPTER 1: DISTILLATION LEARNING LITERATURE REVIEW

Distillation learning has large applications and techniques in several fields. The goal of literature review was to identify useful applications in the bank context. As a first step, we explore a large scop of distillation learning techniques and applications without worrying about their applicability in order to gain maximum knowledge and insights.

In this chapter, we review almost **twenty-nine** scientific works, and we provide a general overview of relevant distillation learning techniques, frameworks[2], and applications. This review has tree aspects. In the first section, we study fundamental techniques necessary to perform distillation learning tasks. This can be regarded as the core foundation of distillation learning theory. In the second section, we explore some successful and efficient frameworks about reducing performance gap between the student model and its teacher. In the last two sections, we provide a general literature summary of relevant applications mainly in explainable machine learning and models' performance improvement and review some existing packages of distillation in *Python*.

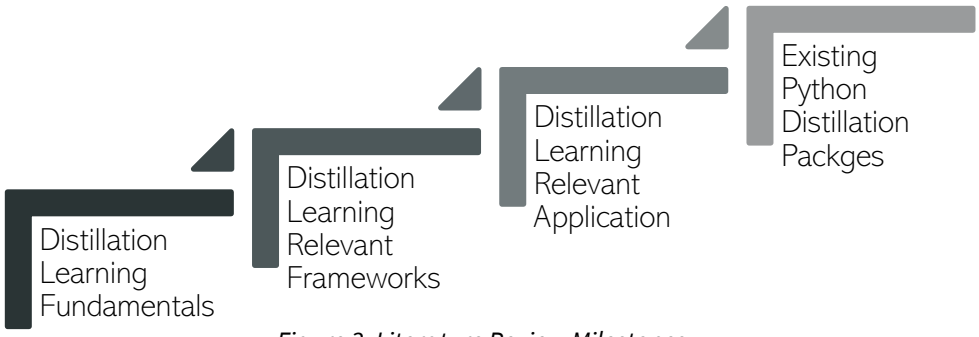


Figure 3. Literature Review Milestones

Distillation Learning Fundamentals

Knowledge

Knowledge refers to teacher's learned experience and consist of what we want to transfer to the student model. In a neural network, knowledge can be weights, feature maps[3], activation functions, feature representations, between layers relationship, or data distribution.

[2] *Specific distillation settings to make the performance of the student as close as possible to its teacher.*
[3] *Feature maps refer to layers output.*

Traditionally, distillation uses logits[4] as source of teacher knowledge. Other techniques focus on weights, activations of intermediate layers and relationship between parameters and hyperparameters.

There are three different categories of knowledge: **Response-based knowledge, Feature-based knowledge, and Relation-based knowledge**. Depending on the information we want to transfer to the student, we can use each or all these techniques to obtain the closest student performance to the teacher.

Response-based knowledge

This is the typical framework for distillation learning. The student tries to **mimic** accurately only the **output predictions** of the teacher model. Practically, we use logits from both teacher and student models and try to minimize their distance using a loss function. Using backpropagation[5], student's parameters are updated according to teacher logits' match.

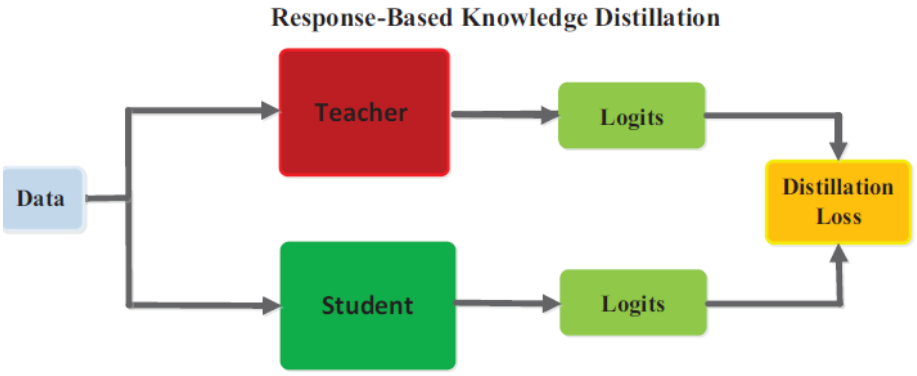


Figure 4. The specific architecture of response-based knowledge distillation. The student model learns to mimic teacher's predictions *(Gou, et al., 2021)*

Logits contains teacher's dark knowledge (Hinton, et al., 2015). Dark knowledge refers to information not directly encoded in the original training dataset and made explicit by the teacher model. In other words, dark knowledge is the relevant and essence information from the original dataset used by the teacher to make its predictions. Hard targets [6] have demonstrated worse performance comparing to logits. Hard classification introduces bias and lack of knowledge which is unhelpful for a student whilst distillation training. For instance, let's consider a binary classification problem. For a specific example, suppose that the teacher's soft labeling is:

[4] *Logits are the raw, unnormalized classification predictions generated by a model before being transformed into probabilities through a SoftMax.*
[5] *Explanations are illustrated on neural networks for the sake of simplicity.*
[6] *Classification labels or categories such as 0 and 1.*

$$\begin{cases} P(y = 1/x) = 0.59 \\ P(y = 0/x) = 0.41 \end{cases}$$

if we choose a classification cut-off equal to 0.5, teacher's hard labeling will be 1. However, hard labeling will ignore that the same example is also more likely to be labeled 0 as classes' probabilities are very close. Hard labeling ignores inter-class information. Therefore, training student on teacher's hard labeling provides poorer information during distillation training.

Instead of using logits, (Hinton, et al., 2015) demonstrated that using soft targets has better performance as it helps the student to capture well teacher's knowledge. For a class i, its probability distribution can be expressed as:

$$pi = \frac{exp(z_i)}{\sum_j exp\left(z_j\right)}$$

Formula 1. SoftMax Probability Distribution

Where Z_i the logit corresponding to the class 'i'. To control softness of probability distribution within classes, (Hinton, et al., 2015) proposed the following formula:

$$pi = \frac{exp\left(\frac{z_i}{T}\right)}{\sum_j exp\left(\frac{z_j}{T}\right)}$$

Formula 2. Hinton Soft-Targets for Response-Based Knowledge Distillation; very high T values correspond approximately to matching logits.

Where T is a temperature term. The $T \to +\infty$ case corresponds approximately to marching logits (Hinton, et al., 2015). The motivation behind using a temperature parameter T in soft targets during knowledge distillation is to control the level of confidence and smoothness of the teacher model's predicted probabilities before transferring them to the student model.

When a higher temperature value is used ($T > 1$), the teacher's predicted probabilities become softer and more evenly distributed across classes. This means that teacher assigns non-zero probabilities to multiple classes leading to low confidence. This increased entropy in the predictions allows the student model to learn from a richer source of information, potentially avoiding getting stuck in local minima during training.

On the other hand, when a lower temperature value is used ($T < 1$), the teacher's predicted probabilities become sharper and more peaked, leading to a more deterministic assignment of probabilities to the most likely class. This can help the student model focus on the most relevant information and reduce the noise in the learning process.

In response-based knowledge distillation, the student is in practice trained to mimic teacher outputs whilst being trained at the same time on ground truth labels. This helps achieve better performance during the test phase.

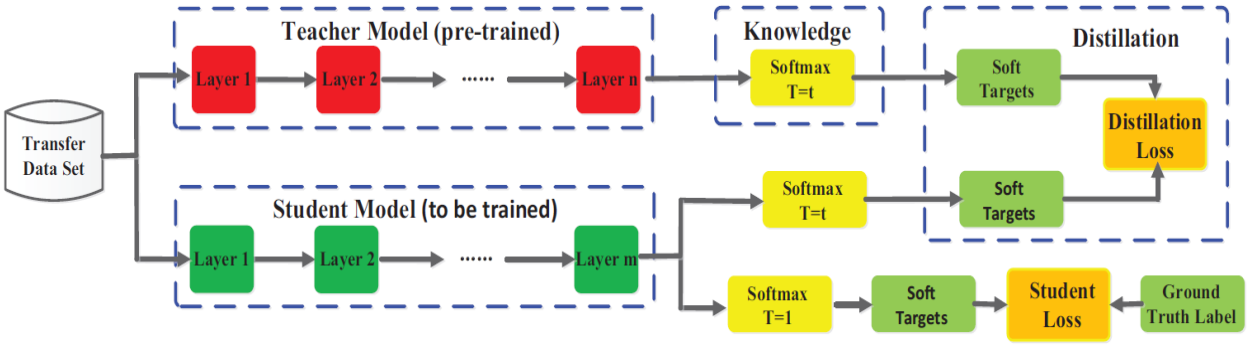


Figure 5. The student model learns to mimic teacher's predictions whilst being trained at the same on the ground truth labels *(Gou, et al., 2021)*.

In terms of loss, response-based distillation loss is expressed as a sum of two losses:

$$\mathcal{L}_{kD} = \sum_{(x_t, y_t)} [\alpha \mathcal{L}_{CE}(f_s, x_t, y_t) + \beta \mathcal{L}_{kL}(f_s, f_T, x_t)]$$

Formula 3. Response-Based Knowledge Distillation Loss

The first loss is the cross-entropy loss with respect to ground truth labels. The second loss is the kullback-Leibler divergence to measure the difference between teacher's and student's probability distributions.

Response-based distillation learning is a very efficient, fast, and straight-forward distillation framework. It focuses mainly to transfer teacher dark knowledge embedded in logits or in soft probabilities and at the same time training the student on ground truth labels. However, it is limited to supervised learning tasks and relies strongly on teacher's outputs. It also fails to address intermediate-level supervision.

Feature-based knowledge

Feature-based distillation learning captures knowledge in intermediate layers in addition to the final layer. It addresses the challenge of transferring not just prediction probabilities but also the underlying informative patterns from the teacher to the student. It is especially useful when the

teacher's nuanced knowledge, present in its feature representations, can enhance the student's learning. This approach helps the student model achieve improved generalization, even with limited labeled data. As shown in *Figure 6*, the distillation loss minimizes the difference between feature representation of teacher and the student models in each intermediate layer.

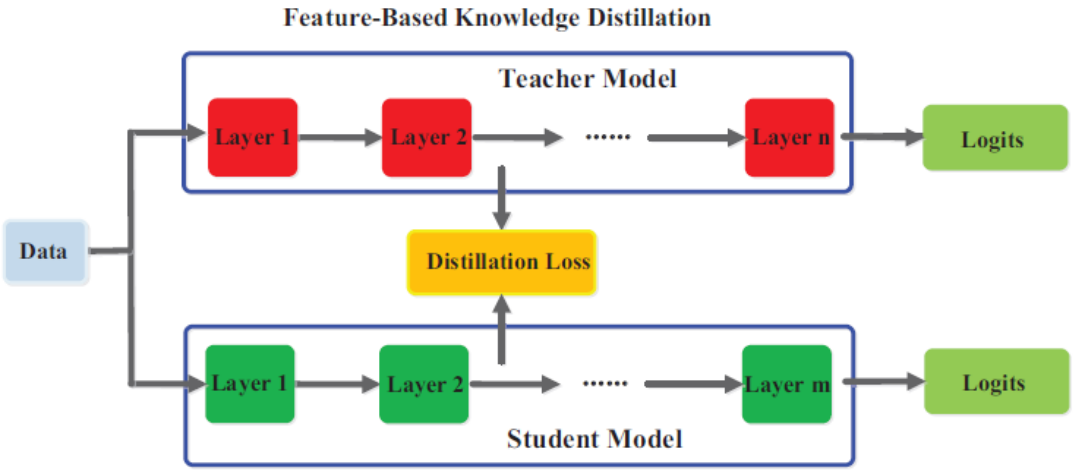


Figure 6. The generic feature-based knowledge distillation *(Gou, et al., 2021)*.

Therefore, both the output of the last layer and the output of intermediate layers, i.e., feature maps, can be used as the knowledge to supervise the training of the student model and therefore reduce the performance gap. Generally, the distillation loss for feature-based knowledge transfer can be formulated as:

$$\mathcal{L}_{FeaD}\left(f_t(x), f_s(x)\right) = \mathcal{L}_F\left(\Phi_t\left(f_t(x)\right), \Phi_s\left(f_s(x)\right)\right)$$

Formula 4. Feature-Based Knowledge Distillation General Loss

$f_t(x)$ and $f_s(x)$ are the feature maps of the intermediate layers of teacher and student models, respectively. The transformation functions, Φ_t and Φ_s are usually applied when the feature maps of teacher and student models are not in the same shape. $\mathcal{L}_F(.)$ indicates the similarity function used to match the feature maps of teacher and student models.

Specifically, feature-based knowledge from the intermediate layers is a good extension of response-based knowledge. This approach proves beneficial in scenarios where direct prediction labels are insufficient to guide the student's learning effectively.

To illustrate the relevance of this approach, imagine a scenario where a teacher model is trained to classify images of animals. As it learns, the teacher model develops intermediate features that

recognize intricate visual details, such as textures, edges, and shapes specific to different animal species. Feature-based knowledge distillation involves transferring these learned features to a smaller student model. The student learns to identify animals not just by mimicking the teacher's predictions but by capturing the underlying visual cues encoded in the intermediate features.

Feature-Based Distillation Learning		
Work	Type of knowledge	Description
FitNets: Hints for Thin Deep Nets (Adriana Romero, et al., 2015)	Feature Representation, Hint Layers	Features representation in intermediate layers capture informative patterns and other characteristics present in the data.

Table 1. Feature representations are mimicked in the intermediate layers by the student as source of feature-based knowledge.

In summary, feature-based knowledge distillation enables a student model to gain insights from the rich intermediate features learned by a teacher model. This approach empowers the student to navigate intricate data patterns and make informed predictions. How to effectively choose the hint layers from the teacher model and the guided layers from the student model remains to be further investigated. Due to the significant differences between sizes of hint and guided layers, how to properly match feature representations of teacher and student also needs to be explored.

Relation-based knowledge

Both response-based and feature-based knowledge use the outputs of specific layers in the teacher model. Relation-based knowledge further explores the relationships between different layers or data samples.

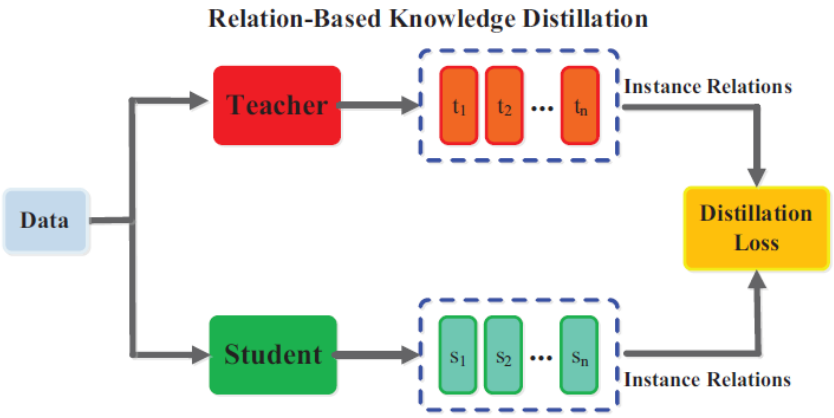


Figure 7. The generic relation-based knowledge distillation. The student tries to replicate instance relations of the teacher *(Gou, et al., 2021)*.

Relation-based knowledge distillation transfers relational information between a teacher and a student model. Response-based and feature-based distillation often focuses on transferring soft probabilities or feature representations. However, in tasks where pairwise relationships or interdependencies between data points are important. Relation-based knowledge distillation aims to transfer knowledge about these relationships.

The teacher model learns complex relationships between data points and feature maps. The goal is to transfer the knowledge of these relationships to the student model. Therefore, loss functions are designed to quantify and enforce the similarity of relational knowledge between the teacher and student. Generally, the distillation loss for relation-based knowledge transfer can be formulated as:

$$L_{RelD}(f_t, f_s) = L_{R^1}(\psi_t(f_{t,1}, f_{t,2}), \psi_s(f_{s,1}, f_{s,2})))$$

Equation 5. Relation-Based Knowledge Distillation General Loss

Where $f_{t,}$ and $f_{s,}$ are pairs of feature maps of teacher and student models. $\psi_t(.)$ and $\psi_s()$ are the similarity functions for pairs of feature maps from the teacher and student models. $L_{R^1}(.)$ indicates the correlation function between the teacher and student feature maps.

Relation-Based Distillation Learning

Work	Type of knowledge	Description
A gift from knowledge distillation: Fast optimization, network minimization and transfer learning (Yim, et al., 2017).	FSP Matrix	The FSP (flow of solution process) matrix is generated by crossing feature representation of two selected layers across data using inner product, gram representation, or other capturing-information product depending on the problem.
Knowledge representing efficient, sparse representation of prior knowledge for knowledge distillation (Liu, et al., 2019).	Parameters distribution, Hint Layers	Parameters' distribution refers to statistical properties of feature representations. Various metrics are used to quantify the similarity between the feature distributions of the teacher and student models such as the mean, variance, covariance, and higher-order statistics. The goal of addressing parameter distribution in feature-based distillation is to ensure that the student model not only mimics the teacher's predictions but also captures the underlying patterns and characteristics of the data as represented by the teacher's features.

Table 2. Different aspects of relation-based knowledge distillation

The FSP matrix is a novel example of relation-based distillation. It is generated by crossing feature representation of two selected layers across data using inner

product, gram representation, or other capturing-information product, depending on the problem and relational information we want to transfer.

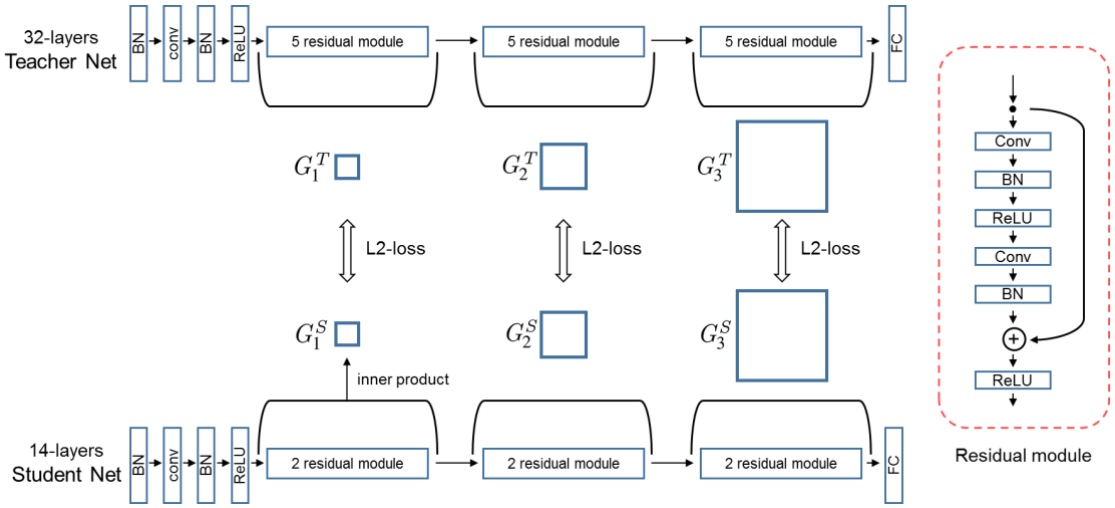


Figure 8. Complete architecture of FSP knowledge distillation (Yim, et al., 2017).

In the framework illustrated in *Figure 8*, there are two main training stages. In first stage, the student network is trained to minimize the distance between the FSP matrices of the student and teacher models. Then, the pretrained weights of the student are used to initialize weights in the second stage. Stage two represents the normal training procedure.

Distillation Training schemes

There are three main techniques employed to train both the student and teacher models: offline, online, and self-distillation. The difference between these distillation training techniques hinges on whether the teacher model is trained simultaneously with the student model. This differentiation is depicted in *Figure 9*.

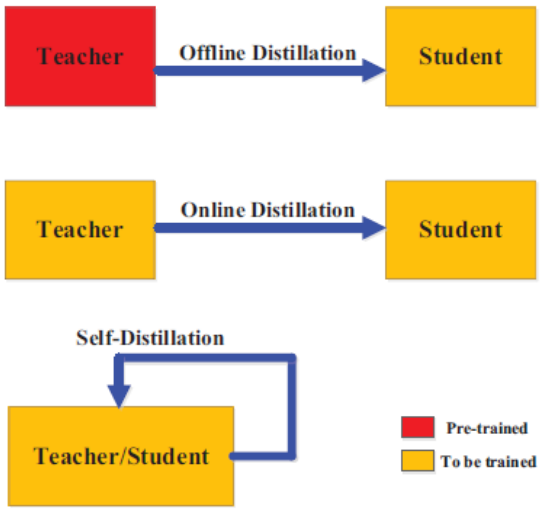


Figure 9. Different Distillation Training Modes *(Gou, et al., 2021)*

Offline Distillation – Teach me what you know!

Offline distillation is the most common method where a teacher model is pre-trained on a large dataset before the student model's training begins. For instance, in response-based distillation learning, the teacher model's predictions (soft targets) are generated for the entire training dataset. The student model is then trained using both the raw training data and the teacher's soft targets. The training process is carried out once, and the student model's parameters are updated to mimic the teacher's predictions. Once trained, the student model remains static unless a new training process is initiated. With the advancements in deep learning, there are many pre-trained models available to be used as teachers for different tasks such *BERT, VGG, ResNet, etc.* This method is well-known and developed in distillation learning field and easier to implement.

To be more explicit, here are some steps to implement offline response-based distillation learning:

1. Teacher and Student Models Selection: we select a larger model as the teacher and a smaller model as the student. The teacher should have a higher capacity and knowledge that you want to transfer to the student.

2. Data Preparation: we prepare the training dataset with labelled examples. The same dataset will be used to train both the teacher and student models.

3. Teacher Model Training: if a pre-trained teacher is not available, we train the teacher model separately from the student on the training dataset. The teacher model should be used to generate soft targets for the training data.

4. Soft Targets Generation: we use the trained teacher model to generate soft targets for the entire training dataset. These soft targets will be used to guide the training of the student model.

5. Loss Function Definition: the loss function for training the student model should include two components: a standard loss term (like cross-entropy) that compares the student's predictions with the ground truth labels, and a distillation loss term that compares the student's predictions with the teacher's soft targets (*Formula 3*).

6. Student Training: finally, we train the student model using both the raw training data and the soft targets generated by the teacher model. The objective is to minimize the difference between the student's predictions and the teacher's soft targets.

Offline Distillation has several limits that can be summarized in:

- Dependency on Teacher Model: the accuracy of the student model is limited by the quality of the teacher model's predictions. Often, there is a capacity gap[7] making the student model unable to outperform its teacher. This is because the teacher model has a bigger number of parameters making him able to generalize on data better than the student.

- Offline/Static Nature: Offline distillation might not be suitable for scenarios where data distribution changes frequently or when real-time adaptation is required. Once the distillation is done, the student model is saved. Updates are possible unless a new training process is initiated.

Despite these limitations, most of knowledge distillation work is offline. Thus, offline distillation remains a valuable technique, straightforward and easy to implement.

Online Distillation – Learning Together

To overcome the limitation of offline distillation, online distillation is proposed to further improve the performance of the student model, especially when a large-capacity high performance teacher is not available (Gou, et al., 2021) . In online distillation, both teacher and student models are updated simultaneously, and the whole knowledge distillation framework is end-to-end trainable.

In online distillation, both the teacher and student models are initialized, and the student model starts learning at the same time of teacher training. In other words, the student model is updated iteratively while teacher training. In contrast with offline distillation, the teacher is not pre-trained or trained separately.

Usually, online distillation is used when a pre-trained teacher model is not available. Thereby, student and teacher are updated simultaneously in a single end-to-end training process. Online distillation can be implemented using parallel computing thus making it a highly efficient method.

Online distillation training has several applications such as online adversarial knowledge distillation (Zhang, et al., 2021a). This method proposed to simultaneously train the student using a teacher and discriminator feedback within GANs framework to generate divergent examples.

Online Distillation has two major limitations. Since teacher and student training is done simultaneously, there is often a high training complexity that might require important computational resources. The other problem remains the student-teacher capacity gap as in offline distillation (Gou, et al., 2021).

[7] *Capacity gap refers to the difference in complexity and the number of parameters between the teacher and student models.*

Self-Distillation – Learning autonomously.

In self-distillation, the same networks are used for the teacher and the student models. In other words, the student model teaches itself by using its own predictions as additional training targets. Self-distillation is particularly useful when you have a strong, pre-trained model and want to make it even better.

To illustrate self-distillation training, suppose we want to perform knowledge-based distillation learning and we want to improve the performance of a particular model. Self-distillation is simply choosing a student with the same architecture as its teacher and perform distillation to incorporate useful features captured by the teacher and information coming from ground truth labels. Self-distillation is usually performed using an online training. Over iterations, the student model becomes more confident and accurate in its predictions, hopefully capturing subtle patterns and relationships in the data that weren't initially obvious. Self-distillation can also be regarded as a regularization technique as the student is inherently encouraged to learn more robust features from the teacher's predictions.
As a framework's example, Adversarial co-distillation Networks (ACNs) is a novel technique to enhance the performance of a CNN[8] in the image recognition task by generating divergent examples where models do not totally agree (Zhang, et al., 2021a). The goal is to have them make the same predictions accurately based on a majority vote.

Model	Standard Training[9] Performance (%)	ACNs Performance (%)
Resnet-20	68.22	70.67
VGG11	67.38	70.11
AlexNet	39.45	46.27

Table 3. Distillation learning can be used to enhance complex models' performance without compression.

[8] *Convolutional Neural Networks*
[9] *Training without distillation using original dataset with ground truth labels.*

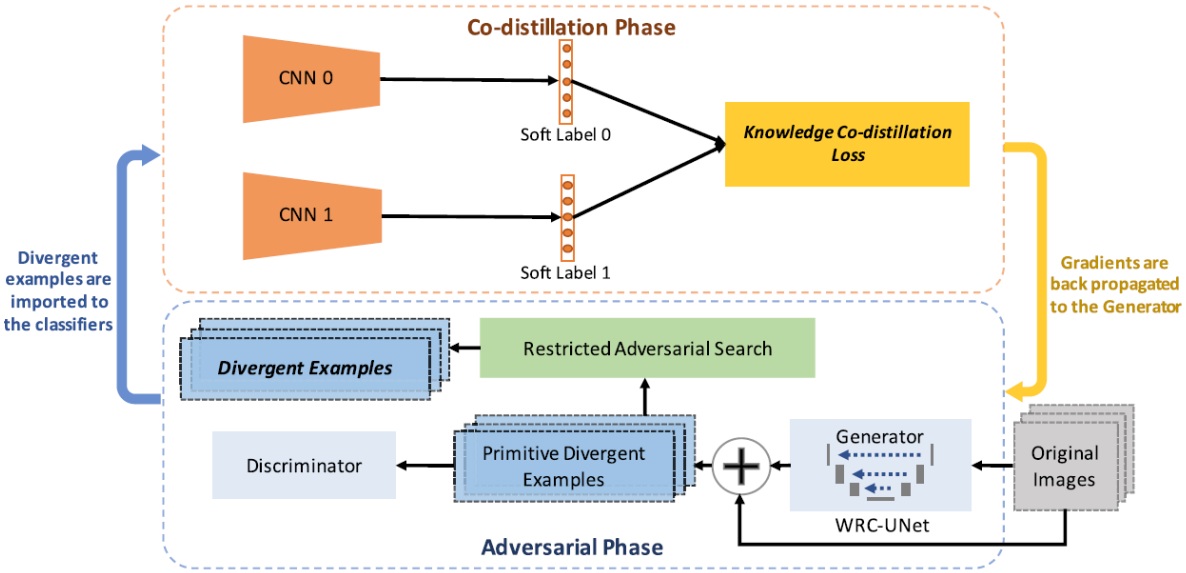


Figure 10. The framework illustration of ACNs. ACNs consist of an adversarial phase and a co-distillation phase. The adversarial phase generates the divergent examples, and the co-distillation phase learn the divergent examples. The adversarial phase is designed according to the GANs framework *(Zhang, et al., 2021a)*

As shown in *Table 3*, Adversarial co-distillation networks which is a framework of self-distillation, notably improves the three models' performance. For instance, *AlexNet* has a baseline performance of 39.45%. ACNs framework improves its performance to 46.27% which is very remarkable.

Distillation Learning Relevant Frameworks

Multi-Teacher Distillation

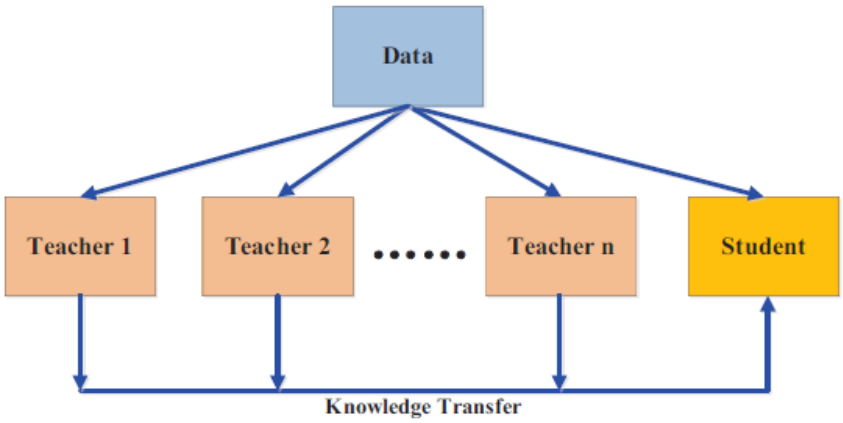


Figure 11. Multi-Teacher Distillation Framework *(Gou, et al., 2021)*.

Multi-teacher distillation is a technique in which a student model learns from predictions made by multiple teacher models, rather than just one. This approach aims to capture a broader range of knowledge and insights from different sources, enhancing the student's understanding and

generalization capabilities. Each teacher might specialize in a different source of knowledge, and the student learns to combine their collective knowledge. For instance, one teacher can transfer response-based knowledge and the other transfers feature-based knowledge in intermediate layers (Chen, et al., 2019). Multiple teachers have turned out to be effective for training student model usually using logits and feature representation as the knowledge. To transfer knowledge from multiple teachers, the simplest way is to use the averaged response from all teachers as the supervision signal (Hinton, et al., 2015).

Benefits of multi-teacher distillation include enhanced robustness, better generalization, and improved performance compared to single-teacher distillation. It can be especially useful when different teacher models have diverse strengths and areas of expertise, which can collectively contribute to the student's learning. It also reduces bias coming from one teacher due to its lack of knowledge.

Data-Free Distillation

Data-free distillation is a technique that allows a smaller student model to learn from a pre-trained teacher model without using distillation training data to overcome problems with unavailable data arising from privacy, legality, security, and confidentiality concerns. Instead, data is synthetically generated. For instance, in (Chen, et al., 2019), synthetic data is generated by a GAN. Synthesis data in data-free distillation is usually generated from the feature representations from the pre-trained teacher model.

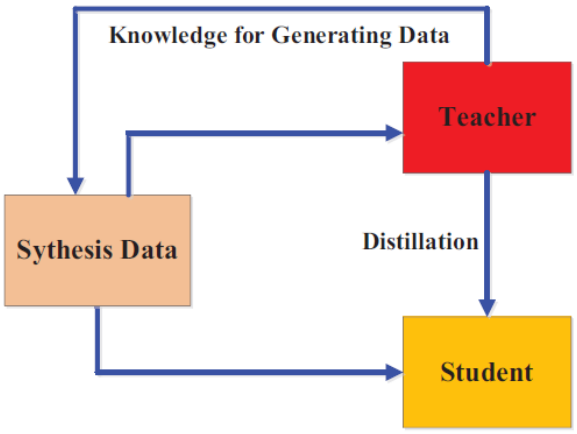


Figure 12. Data-Free distillation Framework

Adversarial Knowledge Distillation

Adversarial Knowledge Distillation is an effective framework to enhance the power of student learning via the teacher knowledge distillation using GANs. This framework tackles two main problems:

1. Difficulty for the teacher learn the true data distribution (lack of data, unrepresentative data, small model, etc.)

2. Small capacity of the student and difficulties to mimic accurately the teacher (capacity gap, unreliable teachers)

The first problem is straightforward. It consists of using GANs to generate more training data to improve teacher's generalization ability. This framework is efficient if we want to enhance the performance of simple models such as logistic regression since a powerful teacher provides more accurate students.

In the remaining of this section, we focus on GAN-based frameworks intended to reduce the performance gap between the student and the teacher. Generally, the distillation loss used in the GAN-based distillation learning can be formulated as:

$$\mathcal{L}_{KD} = \mathcal{L}_G(F_t(G(z)), F_s(G(z)))$$

Formula 6. Distillation loss of GAN-based framework

where $F_t(.)$ and $F_s(.)$ are the outputs of the teacher and student models, respectively. $G(z)$ indicates the training samples generated by the generator G given the random input vector z, and $\mathcal{L}_G$ is a distillation loss to force the match between student's and teacher's data distribution.

According to the literature different work, this framework can be divided into three main categories as shown in ***Erreur ! Source du renvoi introuvable.***.

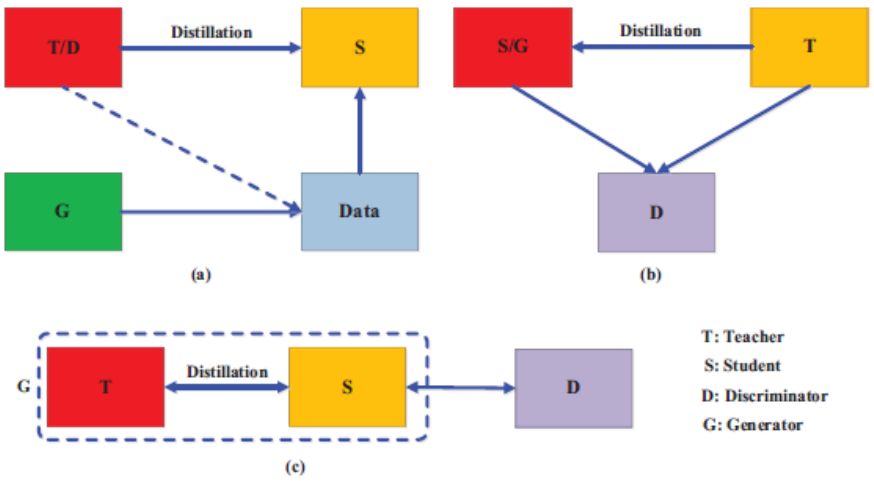


Figure 13. The Main Schemes of Adversarial Distillation Framework

In the first scheme (a), a generator is trained on true data distribution using GANs. The teacher can be used as a discriminator. In this case, generated data go through teacher discrimination based on its proper data distribution. Student learns then teacher's knowledge through generated data embedding teacher's internal feature representation (Chen, et al., 2019). Otherwise, generated data is used as additional data to improve the distillation process.

In the second scheme (b), a discriminator is trained to distinguish the samples from the student and the teacher models by using either the logits (Xu, et al., 2018) or the features. In addition to traditional distillation process, the student will generate new data based on its internal feature distribution corrected each time by the discriminator. The training stops until the discriminator is sufficiently fooled by the student. The general loss of this scheme is expressed as the following:

$$\mathcal{L}_{GANKD} = \mathcal{L}_{CE}(G(F_s(x)), y) + \alpha\mathcal{L}_{kL}(G(F_s(x)), F_t(x)) + \beta\mathcal{L}_{GAN}(F_s(x), F_t(x))$$

Formula 7. Scheme (b) distillation loss

where G is a student network and $\mathcal{L}_{GAN}$ indicates a typical loss function used in generative adversarial network to make the outputs between student and teacher as similar as possible.

In the third scheme (c), a discriminator is trained on true data distribution and corrects feature distribution of generators which are both the student and teacher. The teacher and the student are jointly optimized in each iteration in an online setting (Chung, et al., 2020).

In conclusion, adversarial knowledge distillation is a novel framework to reduce the performance gap between the student and the teacher.

Explicability Distillation

Teacher's explanations are important features and patterns driving a specific prediction. However, standard distillation doesn't transfer teacher's explanations exclusively thus, student predictions are not driven by the same features as the teacher. *Figure 14* show explanation's inconsistency between the teacher and the student.

Figure 14. Inconsistency between teacher's and student's explanation using SHAP.

(Alharbi, et al., 2021) have proposed a novel framework to distill explanation in addition to standard knowledge[10] for computer vision tasks called X-Distillation (XD). It can be regarded as feature-based knowledge distillation. The framework has outperformed response-based distillation method. However, it has a bigger number of trainable parameters.

Model	**Accuracy %**	**#Parameters**
Teacher	93.78	14,728,266
Baseline Student	89.2	2,781,386
Response-Based Distillation	90.2	2,781,386
X-Distillation	**90.9**	**3,521,276**

Table 4. X-Distillation's Framework Performance

The framework above is proposed to perform X-Distillation in the context of computer vision tasks. The challenge is to effectively feed teacher's explanation features to the student whilst controlling the number of parameters, to achieve the compression goal. The idea behind the whole framework is to provide the student, additional features of teacher's explanation in each iteration to reduce the performance gap between the teacher and the student.

[10] *Logits, feature maps, relationships, etc.*

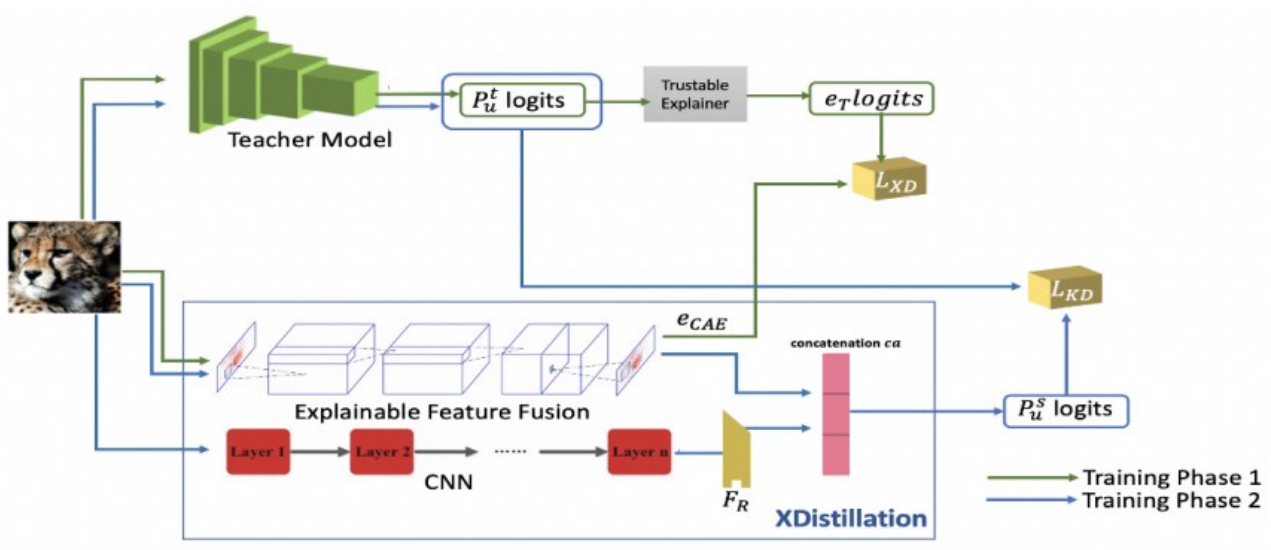


Figure 15. The overall architecture of Xdistillation as described in *(Alharbi, et al., 2021)*

This also leads to consistent explanation of both models. (Alharbi, et al., 2021) describe XD framework as the following:

1. Generate Teacher Explanation: Given teacher logits P_u^t, a trustable post-hoc explainer as SHAP or GradCAM is used to generate teacher explanation logits e_T.

2. Autoencoder Training: We train then a convolutional autoencoder[11] (CAE) to approximate teacher's explanations e_T using the mean absolute error (MAE) loss:

$$L_{XD} = \frac{1}{2n} \sum_{i=1}^{n} \left| e_{CAE_i} - e_{T_i} \right|$$

   with e_{CAE_i} being CAE's logits and e_{T_i} is the ground truth explanations logits of the teacher. The CAE's parameters are frozen after completing the training.

3. Student Training: We train the CNN student using response-based distillation loss. Teacher's explanation features are approximated by the CAE. The resulting representations are flattened and concatenated with the CNN flattened vector to form one high dimensional array before they are

[11] *Usually, an autoencoder is a semi-supervised model. Given an input X, it tries to map it to a latent space without explicitly being supervised. In our case, we force the autoencoder to supervision.*

used as an input of the classifier part (fully-connected layers) of the model. We use a penalization method to reduce the dimensionality of the concatenated array. Then, we feed the result to the fully connected layers for classification. Finally, we use a response-based distillation loss to perform distillation learning.

Finally, one might ask about the novelty of using an autoencoder instead of using teacher's explanation feature directly. As we have seen in the framework in *Figure 15*, training the CNN student requires teacher explanation at each iteration. Directly embedding teacher's explanation features will make the task of student training computationally cumbersome. This is the reason behind an autoencoder approximation for teacher's explanation features.

Distillation Learning Relevant Applications

In this section, we review some relevant distillation learning applications in the literature. Certainly, this is not an exhaustive list as we oriented our readings to potential model risk management applications. (Liu, et al., 2018) have performed distillation on the image recognition task using MNIST[12] dataset where the teacher model is a CNN, and the student is a decision-tree. (Che, et al., 2015) have distilled a Long Short-Term Memory (LSTM) into *Gradient Boosting Trees* for the *VENT* mortality task[14]. On the other side, (Tang, et al., 2019) have applied distillation learning for NLP[16] tasks by distilling a *BERT* (Devlin, et al., 2018) teacher into a smallest *BiLSTM* using a transfer set constructed by GPT-2 (Radford, et al., 2019) and TXL (Dai, et al., 2019). There are multiple other applications in speech recognition, NLP, and visual recognition (Dosilovi, et al., 2018). However, they have one thing in common which is model compression. As we have mentioned before, the core distillation learning application is model compression (Caruana, et al., 2006) to optimize model complexity and reduce computational needs. However, many applications have arisen from this definition. Reducing complexity infers using a simpler model which can be also an interpretable model such a logistic regression or a decision tree. Also, through the process of transfer learning between the teacher and the student, distillation learning also improves model's baseline performance in specific frameworks. In Table 5, we provide a review of some relevant work in the applications listed above.

Existing Python Distillation Packages

[12] *Handwritten digits classification (LeCun, et al., 1998).*
[14] *Binary classification of whether a patient dies within 60 days after admission or not on VENT dataset (Khemani, et al., 2009)*
[16] *Natural Language Processing*

To perform knowledge distillation, one might ask if there are any existing packages to avoid coding from scratch. However, a generalized knowledge distillation python library does not exist. This can be explained by the diversity of possible applications and purposes behind distillation. Also, distillation frameworks are numerous and depends mainly on the situation.

Meanwhile, *Keras* library provides some code snippets and use-cases to illustrate distillation learning training. However, implementation is performed only to distill a neural network in a less complex neural network. If we want for example to distill a neural network in a simple model like logistic regression or if we want to use another teacher like *XGBoost*. This might not be possible.

Article	Description	Performances	Commentary
Improving the Interpretability of Deep Neural Networks with Knowledge Distillation **(Liu, et al., 2018).**	Distill *deep neural networks* (CNN) into *decision trees* for the MNIST task. In this work, we seek to improve model's interpretability on an image recognition task.	Baseline student's accuracy: **84%** Teacher's accuracy: **99.25%** Distilled student's accuracy: **86.6 %**	The choice of an interpretable student always creates a performance gap between the student and the teacher. However, outperforming the baseline is always possible in interpretability applications.
Distilling Knowledge from Deep Networks with Applications to Healthcare Domain **(Che, et al., 2015).**	Distill a *Long Short-Term Memory (LSTM)* into *Gradient Boosting Trees* to learn interpretable features on *VENT* mortality task.	Baseline student's ROC AUC: 72% Teacher's ROC AUC: **76.55 %** Distilled student's ROC AUC: **75.5 %**	Gradient boosting trees are not interpretable. In the article, they decompose the ensemble and then choose the tree with the greatest weight for a given prediction to seek interpretability.
Faithful and Plausible Explanations of Medical Code Predictions **(Wood-Doughty, et al., 2021).**	For MIMIC-III task[17], authors distill a specific neural network *DR-CAML* (Mullenbach, et al., 2018) into a proxy linear regression for each class. To achieve this, they transform a multi-class classification problem to a proxy regression problem to predict each class probability provided by the DR-CAML teacher.	Baseline student, multi-class logistic regression **macro-ROC AUC: 59.6%**; **micro-ROC AUC: 88.9%** Teacher DR-CAML **macro-ROC AUC: 90.6%**; **micro-ROC AUC: 97.2%** Distilled student, proxy linear regression **macro-ROC AUC: 90.1%**; **micro-ROC AUC: 96.7%**.	Because the linear regression is not a multi-output model. Distillation must be performed for each class using teacher's weights.
Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation **(Tan, et al., 2018).**	Predicting recidivism risk using COMPAS (Angwin, et al., 2016) software as a teacher and iGAM (Caruana, et al., 2015) model as a student.	Baseline iGAM's accuracy: **74%** COMPAS Teacher's accuracy: Unknown Distilled student iGAM accuracy: **75 %**	In this case, we don't know neither the teacher model nor its performance which is a good illustration of black-box models. Distillation learning is useful for training interpretable models.
Natural Language Generation for Effective Knowledge Distillation **(Tang, et al., 2019)**	Distill a *BERT* (Devlin, et al., 2018) teacher into a smallest *BiLSTM* using a transfer set constructed by GPT-2 (Radford, et al., 2019) and TXL (Dai, et al., 2019).	*BiLSTM* baseline's accuracy: **87.6%** BERT Teacher 's accuracy: **94.9%** Distilled *BiLSTM* student on GPT-2 transfer dataset accuracy: **92.7 %**	BERT models are very heavy and requires important computational resources for training. Distillation learning reduces complexity while staying faithful to the original performance.

Table 5. Relevant Distillation Learning Applications in Literature.

In addition, there is some mistakes related to the *Keras* implementation (Borup, 2020) of response-based distillation. For instance, the use of probabilities instead of logits when softening teacher's prediction with the temperature term T is not

[17] *Multi-class classification consisting of assigning clinical notes to ICD codes (Johnson, et al., 2016).*

matching with Hinton method (Hinton, et al., 2015). **In other words, instead of using logits z_i in Formula 2, the author uses teacher's *SoftMax* predictions pi which is incorrect**.

In conclusion, we recommend performing distillation learning from scratch especially if the student or the teacher models aren't neural networks. If it is not the case, the use of *Keras* snippet codes isn't a bad idea, however, one must pay attention to the remark discussed above.

So far, we have seen relevant work, frameworks, and applications about distillation learning. It is worth nothing that most of the existing work focuses on classification tasks, but adjustments to regression problems are possible and sometimes straightforward (Takamoto, et al., 2020).

In the next chapter, we will discuss some of the useful applications of distillation learning especially in the context of model risk management (MRM).

CHAPTER 2: MRM USEFUL APPLICATIONS

In this chapter, we identify some useful applications in the context of model risk management (MRM). This can be regarded as a conclusion of what we have considered as useful for MRM through the literature review. Most of knowledge distillation advanced work is developed for image or text data. However, adaptation is straight-forward for tabular data. Tree useful applications have been identified: explainable machine learning, models' performance improving and reducing models' complexity.

Explainable Machine Learning

Distillation learning has been explored in the context of explainable artificial intelligence also known as XAI (Dosilovi, et al., 2018). XAI focuses on making complex machine learning models more interpretable. Applying distillation techniques to XAI aims to combine the predictive power of complex models with the transparency and interpretability of simpler models (Liu, et al., 2018). In the context of risk management, interpretability becomes crucial for understanding how models arrive at decisions or predictions. Banks and financial institutions need to be able to explain their models' decisions to regulatory authorities, stakeholders, and customers. Indeed, the *European Banking Authority* set an ensemble of regulatory obligations on models' interpretability and urges banking institutions to understand deployed models, their assumptions, limitations, and outputs. Thus, models' interpretability can facilitate better risk assessment and management.

Often, there is a trade-off between model interpretability and performance since transparent models are less accurate and vice versa. The application of distillation learning in XAI seeks to strike a balance between model accuracy and transparency. It enables the creation of models that are both accurate and understandable, addressing the challenges of deploying complex machine learning models in domains where transparency and interpretability are critical such health care.

There are two types of model's interpretability. Intrinsic interpretability consists of choosing and training easy-to-understand models and post hoc interpretability which is explaining models' outputs (Molnar, 2023). Distillation learning interpretability is both intrinsic since we choose an explainable student and post-hoc since student explanation are provided for teacher's output. However, distillation interpretation is not exhaustive since the distillation process can be itself not interpretable (Rudin, 2019).

To illustrate application in MRM, suppose we have an inexplicable teacher such as a deep neural network, a gradient boosting or a random forest and we want either to explain global predictions or to replace it by a good performing intrinsically explainable model. In this case, we can use distillation of the teacher to train an interpretable student model such as a decision tree, logistic or linear regression. The

result-in student is interpretable along with being close to teacher's performance. Practically, we can use the teacher for inference for faithfulness[19] alongside with student's interpretability insights.

As an example, we perform response-based distillation for the *MNIST* task. The teacher is the Convnet model already trained and fine-tuned in the *Keras Python* library. In the other hand, the student model is a *Soft Binary Decision Tree*[20] (SBDT). As explained in Figure 16, in each node, we have a filter matrix and bias vector (parameters) used to capture patterns from the original image. The training process consists of finding the optimal parameters driving student's predictions.

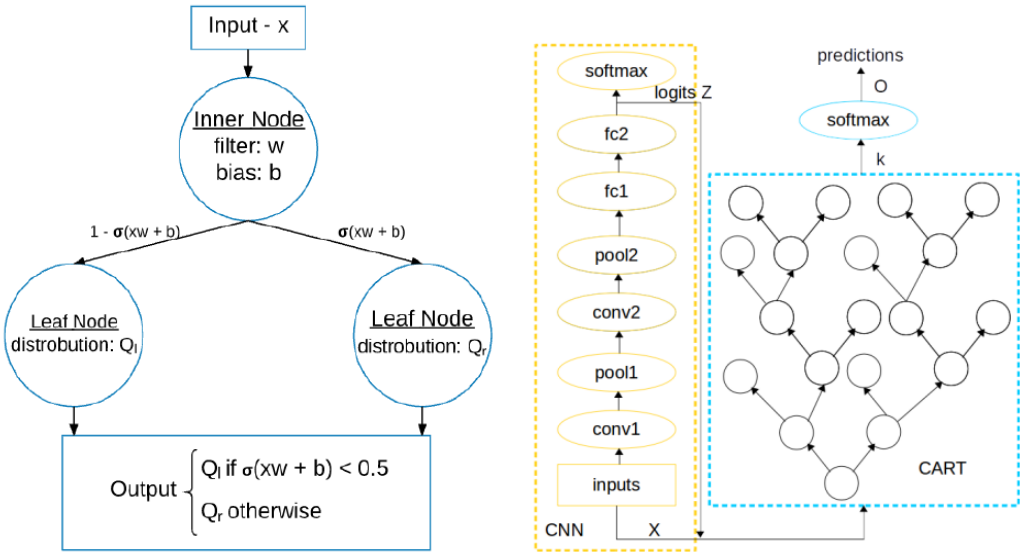


Figure 16. Framework of training *a Soft Binary Decision Tree (SBDT)* using response-based distillation of Convolutional *Neural Network* (ConvNet).

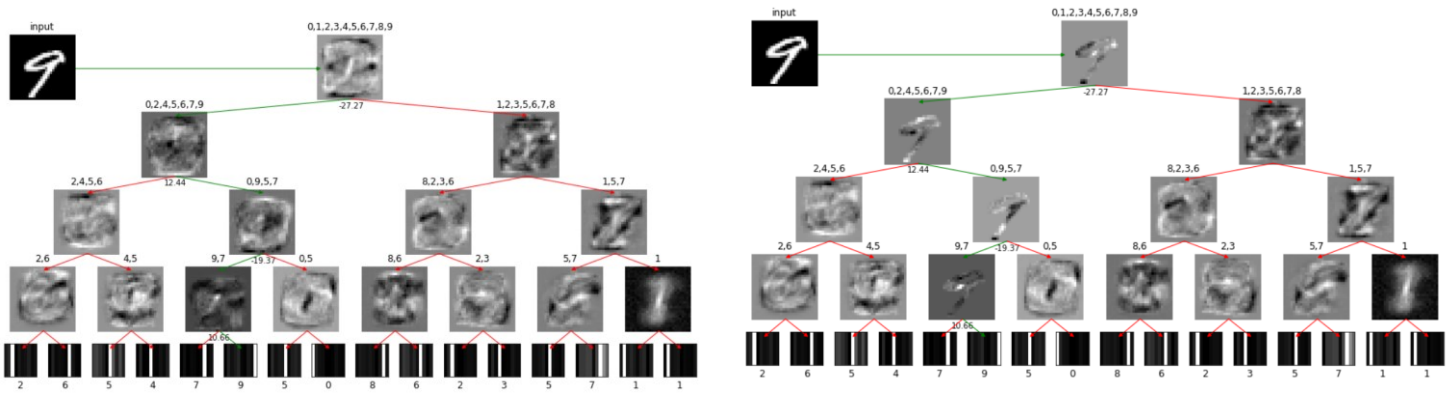
As depicted in Table 6, teacher's accuracy is outstanding (*99.29%*). However, the latter lacks interpretability. The goal is to train an interpretable model with similar or closer accuracy to the teacher model for global explicability.

[19] *The ability to render accurate predictions.*
[20] *Decision Tree model adapted to the context of image classification.*

Model	Role	Batch size	Epochs	Accuracy
ConvNet (CNN)	Teacher	16	12	99.29%
SBDT	Baseline	4	40	80.88%
SBDT Trained with Response-Based Distillation	Student	4	40	90.71%

Table 6. Distillation Performance. Distillation training outperforms standard training[21] but performs worse than the teacher. However, we gain in explicability.

SBDT is a tree-based model which means that it is interpretable. The baseline model trained without distillation has an accuracy of *80.88%* which is lower than the teacher model. We train then SBDT with response-based distillation. We obtained an accuracy of *90.71%* which is lower than teacher's performance due to the capacity gap. It is worth nothing that the student model outperforms the baseline model trained without distillation.



(A) **(B)**

Figure 17. Tree's maximum probability path for classification explicability; A) Explanation's filters provided by SBDT trained traditionally without distillation; B) Explanation's filters provided by SBDT with ConvNet distillation.

Despite the loss of performance in the student model caused by the capacity gap with respect to the teacher, we gained in interpretability whilst staying almost closely faithful to teacher's performance. There is always a trade-off between performance and interpretability that must be balanced depending on the objective of the distillation.

[21] *Training without distillation*

Figure 17 illustrates the quality of explanations made the baseline and the student models. The student model captures patterns with less noise compared to the baseline model and use reliable filters from teacher's dark knowledge to drive the predictions.

As demonstrated earlier, the distillation of a powerful teacher to train a simple student led to better performance and quality interpretation compared with training the same simple model directly on data as we will detail in the next paragraph.

Models' Performance Improving

Improving the performance of simple models

Knowledge distillation can be an effective way to improve simple models' performance. Often, simple models refer to intrinsically interpretable models such as a linear regression, decision trees or logistic regression. In bank context, models are often limited to be simple and are usually centred around shallow decision trees or logistic regression due to regulatory, cost, and time constraints. For instance, in credit's risk management, PD[22] estimation models[23] are a compilation of logistic regressions and decision trees. Despite their interpretability advantage, PD estimation models have disappointing performance on test set.

Most of the work on knowledge distillation have always managed to improve simple models' performance. Dark knowledge of distillation training helps the student generalize better than traditional training because it contains patterns that a simple student usually fails to capture. **Instead of cocking its own meal, student can only eat what the teacher has already prepared**. (Liu, et al., 2018) distil deep neural networks (CNN) into decision trees for the MNIST task for the sake of interpretability. The baseline decision tree model trained without distillation has an accuracy of *84%* during test. On the other hand, the same decision tree trained using the distillation framework in *Figure 4* has an accuracy of 86.6%. More examples are provided in *Table 5*.

In the context of using simple models' constraint, distillation training performs better than traditional training. To do so, we train a complex teacher model capable to achieve good performance and generalize better. Usually, we choose between deep neural networks or boosting trees. In some tasks like *MNIST*, a

[22] *Probability of default*
[23] *Scoring models to predict the default probability of a loan.*

trained teacher is already available to be used. We then select a simple model, or a set of simple models and we perform distillation training as described in *"Distillation Learning Fundamentals"* section.

Improving the performance of sophisticated models

Sophisticated models refer to complex and accurate models that have reached some performance on a specific task. Usually, their performance on the test set is quite good and acceptable. As an example, in NLP tasks like sentiment analysis, BERT (Devlin, et al., 2018) achieves more than *90%* accuracy on multiple benchmarks. There are a lot of other baseline sophisticated models in literature such as *VGG, ResNet, etc.* Often, sophisticated models are used as teachers in distillation learning.

To improve sophisticated models, distillation learning which is defined as a model compression technique is not an effective method due to the capacity gap between the teacher and student. However, self-distillation frameworks can secure sophisticated model improvement. Adversarial co-distillation (Zhang, et al., 2021a) as explained in *"Self-Distillation* – **Learning** section has improved *VGG* and *ResNet-20* performance, generating divergent examples where models do not totally agree. In this framework, teacher and student are the same and we generate divergent examples where models do not totally agree in purpose to make them generalize better in the weakness zone.

Reducing Models Complexity

In the context of compression, distillation can be used to reduce models' complexity. However, student cannot outperform the teacher in general due to capacity gap. In result, several frameworks were developed by scholars to reduce the performance gap between the teacher and student. Some of the most relevant frameworks are adversarial knowledge distillation, explicability distillation and transfer learning. (SANH, et al., 2020) have reduced *BERT* transformers complexity by reducing the number of the parameters by 40% whilst maintaining about 97% of the performance, which has helped to have a faster model, lighter and deployable on small capacity edge devices.

CHAPTER 4: DISTILLATION OF PD ESTIMATION MODELS

In this chapter, we perform distillation learning of **PD[24] estimation models** *used* to estimate the PD on **France's SME[26] portfolio**. The goal is to enhance PD estimation models' performance as detailed in **"*Improving the performance of simple models*"** section. An Introduction to PD estimation models is provided in appendix.

Distillation Framework of *PD estimation models*

To initiate the distillation process for PD estimation models, it's imperative to commence by training a sophisticated teacher model. This teacher model should exhibit superior performance compared to baseline[30] PD estimation models. Otherwise, the process of distillation would lack significance.

Furthermore, to isolate and accurately assess the impact of distillation, it is essential to maintain consistency with the credit risk team's training framework. This means keeping data preparation, regularization techniques, data encoding methods, and other relevant aspects identical. This approach ensures that any observed changes in model performance can be attributed solely to the distillation process, without any confounding effects related to differences in data preparation or other training procedures.

The framework depicted in the *figure 17* has consistently demonstrated superior performance. This framework involves the amalgamation of features from all modules, resulting in a total of 61 features. It's worth noting that all these features are categorical, except for one, which is continuous. To ensure comparability with the credit risk team's approach, we apply one-hot encoding to the categorical features and standardize the continuous feature, mirroring their preprocessing steps.

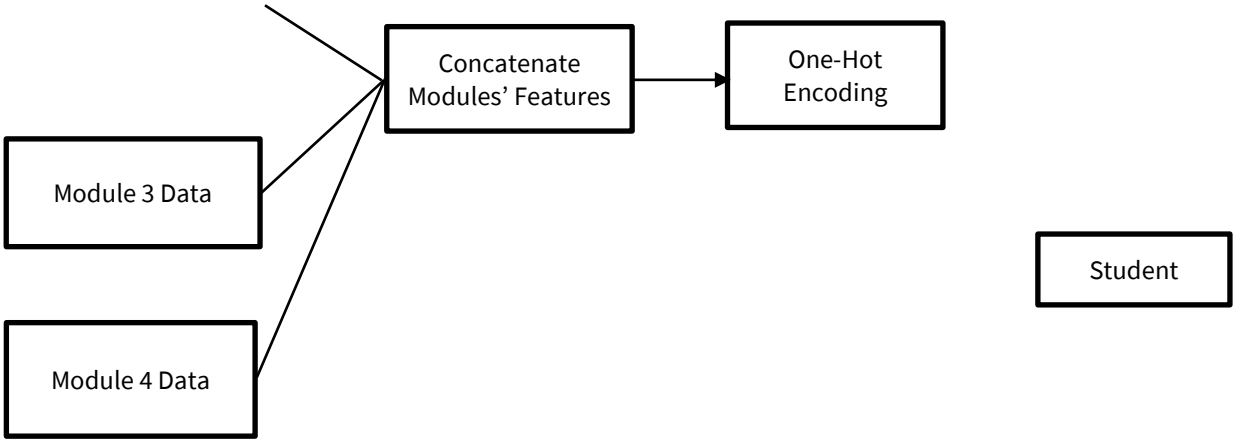


Figure 19. Distillation framework for *PD estimation models*

With this meticulously prepared dataset, we proceed to train various complex teacher models. Subsequently, we employ several distillation techniques, including Hinton-based distillation *(Figure 5),* adversarial knowledge distillation *(Figure 13),* and X-distillation *(Figure 15. The overall architecture of Xdistillation as* described in . At each stage, we evaluate the performance of the student model within each distillation framework and carefully analyze its contributions, making comparisons across different frameworks to gain insights into their relative effectiveness.

Teacher Training

In preparation for offline distillation learning, the initial step involves the training of a robust teacher model, which will later serve as the source of knowledge for distilling into a student model. Following an extensive benchmarking process using various complex models within the Data Robot *AutoML* tool of Société Générale, we have opted to retain the *LightGBM* model. This model is trained on features derived from the concatenation of all modules, as previously detailed in *Figure 17*. The primary objective is to accurately estimate the probability of default (PD) for each obligor in the dataset.

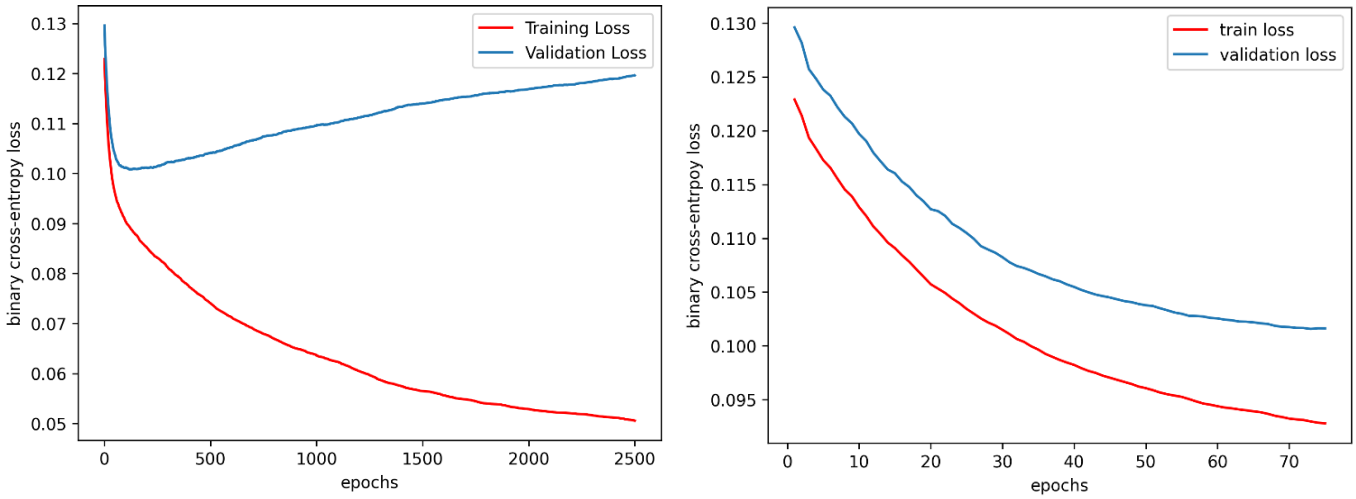


Figure 20. *LightGBM* teacher's learning curves without regularization (Left) and with regularization L1 = 0.01 and early stopping = 7 (Right)

LightGBM, being a tree-based model, does have a propensity to overfit when exposed to a large number of training epochs. As illustrated in *Figure 20*, when applying *LightGBM* to the SMEs dataset with an extended number of epochs, overfitting becomes evident. This outcome is quite expected, considering the dataset's characteristics as it comprises a substantial number of features (61 in total) and trees propensity to overfitting. To mitigate this overfitting tendency, we implement *L1* regularization and incorporate an *early stopping* mechanism during the training process. As indicated in *Figure 20*, these measures prove effective in achieving a more desirable learning curve for the teacher model. Training is halted at approximately 100 epochs, a point at which the model has generally converged and exhibited improved generalization, thereby avoiding the pitfalls of underfitting.

Models	**Role**	**Training AR (%)**	**WT Test AR (%)**	**OOT Test AR (%)**
PD estimation models	Baseline	65,40	66,20	66,4
Feed-Forward Neural Networks (FFNN)	Teacher	72.28	63.25	71.28
LightGBM with regularization	**Teacher**	**70.87**	**67.41**	**71.44**
LightGBM without regularization	Teacher	89.16	58.55	58.84

Table 7. Performance of offline trained teachers using all modules' features

To compare, we calculate accuracy ratio (AR) on each dataset. *Table 7* provides an overview of these results, revealing that the regularized *LightGBM* model outperforms feed-forward neural networks.

However, it is noteworthy that the performance on the WT test falls short of expectations. The difference in performance compared to the PD estimation models is not as significant as desired, despite the teacher's slightly superior performance. Additionally, it is worth mentioning that there is a subtle presence of overfitting in the teacher model, a deliberate choice made to explore whether any anomalies within the teacher's behavior are prone to be transferred to the student model during the distillation process.

In summary, we have employed multiple teacher models trained on the comprehensive set of features encompassing all modules, following the framework described earlier. Among these teacher models, *LightGBM* emerged as the top performer. To address overfitting concerns, we implemented regularization techniques and employed early stopping during its training. As a result, we have chosen the regularized *LightGBM* model as the teacher for our upcoming distillation experiments.

Response-Based Distillation

Our initial step involves performing *Response-based knowledge* distillation. In this process, we utilize the teacher's soft targets as outputs for student training, in conjunction with the ground-truth hard labels, as depicted in *Figure 5*. It is important to note that we cannot directly train a logistic regression model on the teacher's soft targets since the problem transitions from binary classification to a regression task. This is due to the teacher's soft targets representing continuous weights ranging from 0 to 1.

To address this, we opt to train a linear regression model to predict continuous ratios within the 0 to 1 range. However, to ensure that the predicted values are constrained within this specific range, we apply the transformation below to the teacher's soft predictions before initiating the training process.

$$y_{transformed,\,i} = log\left(\frac{pi}{1-pi}\right)$$

Formula 8. Target transformation on the negative real line to constraint the linear regression between 0 and 1.

Where pi is teacher's soft prediction for the class i, z_i are teacher's logits and $y_{transformed,\,i}$ is the new target ranging on the negative real numbers line.

The transformation of pi into the $y_{transformed,\,i}$ values effectively map them onto the negative real number line. The transformed values serve as the target for our linear regression model. Subsequently, during the test phase, we apply the inverse function to these transformed values to obtain probabilities, effectively

reversing the transformation. In essence, this problem can be likened to the training of a logistic regression model, where the goal is to estimate probabilities within the 0 to 1 range.

We implement then *Response-based knowledge* distillation as illustrated in *Figure 6* and employ the loss function defined in *Formula 3.* In the selection of the parameter α , representing the weighting factor of the cross-entropy loss with respect to ground truth labels, a cross-validation process is conducted. The objective is to determine the optimal alpha value that maximizes the accuracy ratio for both the WT test set and the OOT test set. The optimal alpha value is found to be 0.1, signifying that the influence of ground truth labels is negligible when contrasted with the teacher's soft labels.

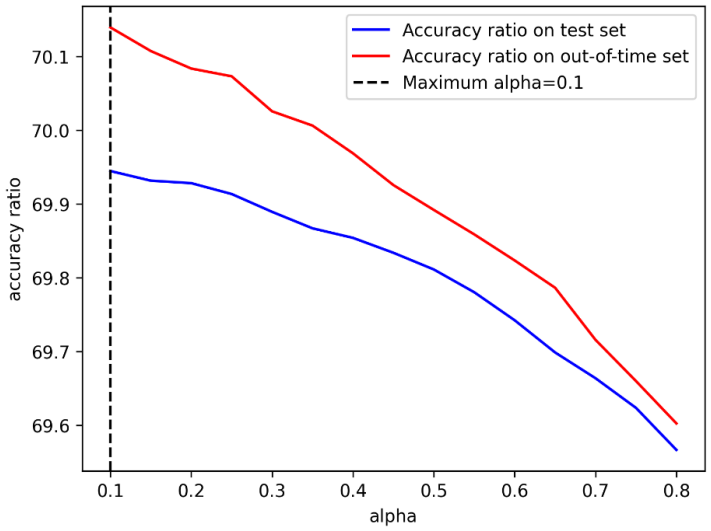


Figure 22. Cross-validation for best α selection, here α = 0.1 is the best selection

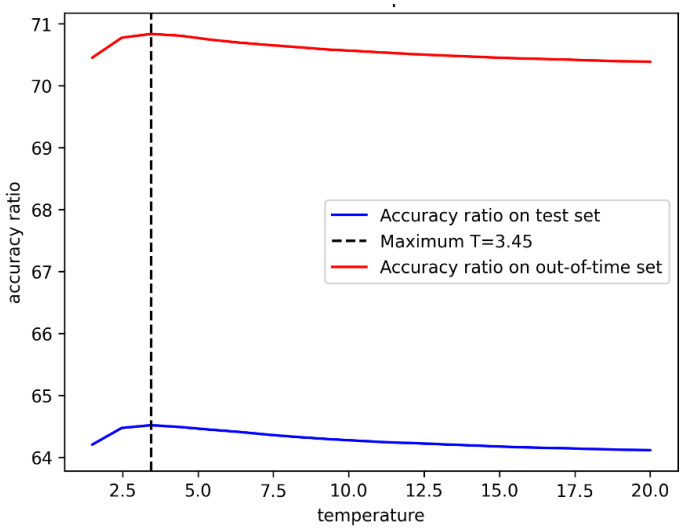


Figure 21. Cross-validation for best T selection, T = 3.45 corresponds to the best accuracy ratio on the test set.

To optimize the performance of the student model, we implement response-based distillation incorporating a temperature term within the SoftMax function, as delineated in *Formula 2*. The selection of the temperature term, denoted as T, is conducted through empirical means, utilizing a cross-validation approach to maximize the accuracy ratio on test sets. The empirically determined optimal value for T is found to be 3.44, surpassing 1. As elucidated previously, this choice of T results in teacher predictions exhibiting greater softness and a more uniform distribution across classes, thereby reducing confidence levels, and increasing entropy. Consequently, the student model benefits from a richer source of information during the learning process.

presents the outcomes of students obtained through the distillation process of the regularized LightGBM teacher, employing the **response-based distillation framework** and the **Offline Distillation – Teach me what you know!training mode**. The first student undergoes training without the use of the temperature term, while the second student is trained with the inclusion of the

temperature term in the SoftMax function. When evaluating their performance on the OOT test set, both students outperform the *PD estimation models,* achieving **70.13%** and **70.83%** accuracy ratio, respectively, as opposed to the *PD estimation models* ' **66.4%.**

Table 8. Students' performance using response-based distillation of *LightGBM teacher.*

However, the performance gap between the students and the teacher persists. On the WT test set, the student trained with the temperature term outperforms the student without temperature (**64.51% vs. 63.94%**). Nonetheless, the latter falls short of the *PD estimation models* ' performance (**64.51% vs. 66.2%),** and the performance gap relative to the teacher remains (**64.51% vs. 67.41%**). It's worth noting that a slight overfitting tendency observed in the teacher's performance is also carried over to its students, as evidenced by the analysis of the accuracy ratio on the training set (Training AR).

Models	**Role**	**Training AR (%)**	**WT Test AR (%)**	**OOT Test AR (%)**
PD estimation models	Baseline	65,40	66,20	**66,40**
LightGBM with regularization	Teacher	70.87	67.41	71.44
PD estimation models Distilled[32] without Temperature	Student	68.03	**63.94**	**70.13**
PD estimation models Distilled with Temperature	**Student**	**68.63**	**64.51**	**70.83**

In summary, we achieved good results by outperforming the baseline PD estimation models on the OOT test set. However, when it comes to the WT test set, our attempt to use response-based distillation learning didn't work as well. This happened because the teacher model wasn't very flexible for this specific test, and the difference in performance compared to the PD estimation models wasn't significant at the beginning. As a result, our student models couldn't match the performance of the PD estimation models due to this performance gap with respect to the teacher model. In the next sections, we explore other distillation frameworks such as adversarial knowledge distillation and X-Distillation to improve student performance on the WT test set.

[32] *Logistic regression student using features concatenated from all modules.*

Adversarial Knowledge Distillation Framework

So far, we have achieved great results on the OOT test set. However, student's accuracy ratio on the WT test set (WT test AR) still needs to be improved. To achieve that, we will be performing an advanced distillation framework which is *Adversarial Knowledge Distillation*.

It is an effective framework to enhance the power of student learning via the teacher knowledge distillation using GANs. This framework tackles the problem arising from the small capacity of the student and difficulties to mimic accurately the teacher leading to the performance gap.

In this specific case of PD estimation models, the most effective framework for student training is the scheme b depicted in *Figure 13*. One way to guide the student model is by using a discriminator. The goal here is to make the data generated by the student as similar as possible to both the teacher's data intrinsic distribution[33] and the real data distribution[34]. So, we train a discriminator to distinguish between the fake[35] data and the actual data distribution or the teacher's data intrinsic distribution.

As the distillation process progresses, the student keeps generating data in each iteration. The discriminator steps in to assess whether this generated data is getting closer to resembling the real data distribution and the teacher's data representation. Using response-based distillation, which we talked about earlier, the student generates data in line with the knowledge it receives from response-based distillation. In each iteration, the student creates data guided by this knowledge. The discriminator steps in to determine if this generated data aligns with both the actual data distribution and the teacher's intrinsic data representation. The distillation stops until the discriminator is more likely to be fooled by the student (generator). This method is very effective as the student is corrected and guided during the distillation process to match real data distribution and teacher's intrinsic data distribution which help for better generalization.

Noise Instance of Independent Features

[33] *Distribution of data generated by the teacher model.*
[34] *Distribution of data in training dataset*
[35] *Data that doesn't follow real data distribution in the training dataset.*

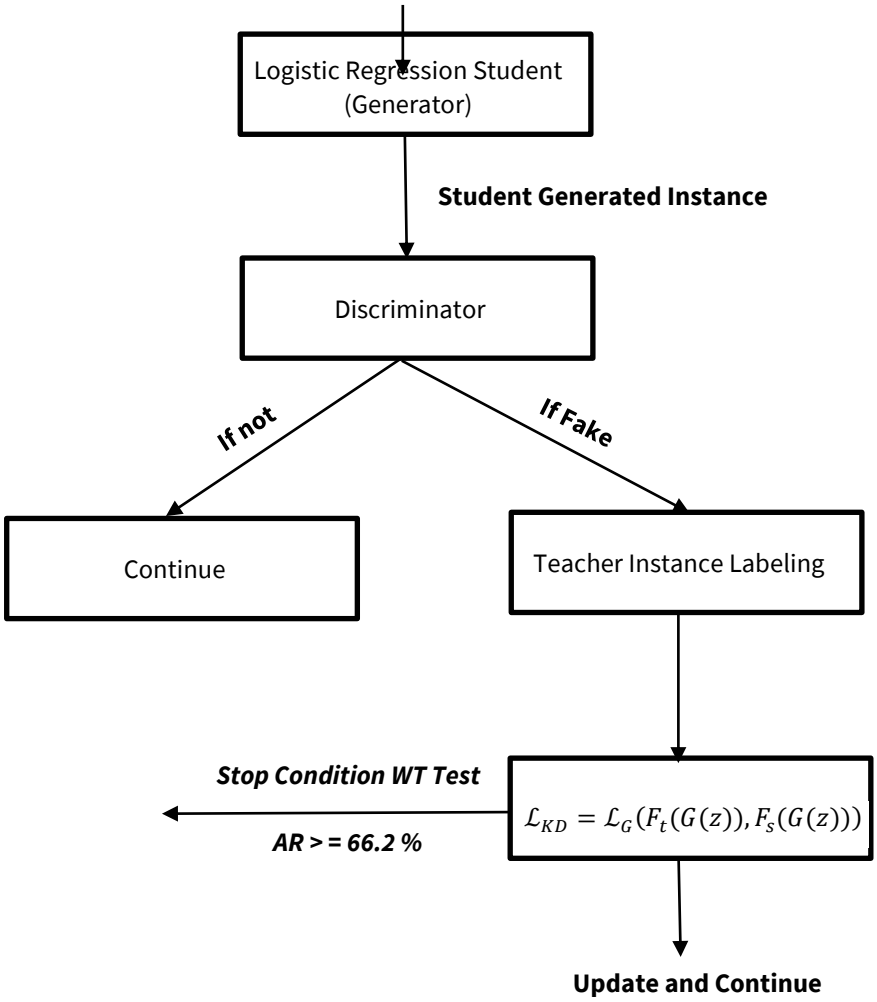


Figure 23. *Adversarial knowledge distillation* training framework performed on PD estimation models. In this case, the generator is also the student which is a logistic regression. The teacher is the regularized *LightGBM* used in the response-based framework. The stop condition corresponds to PD estimation models' performance on WT test set.

In *Figure 23,* we present the adversarial knowledge distillation framework that we used to improve the performance of the PD estimation models on the WT test set, as discussed earlier. The process begins with the creation of synthetic noise, which is then input into the student (also functioning as the generator) to produce data. In our scenario, this noise consists of independent features without labels.

The generated data is then subjected to evaluation by a discriminator, which determines whether it is real or fake. If the generated data is deemed real, the process continues by introducing another instance of noise. However, if the generated data is considered fake, we turn to the teacher for labelling, using the teacher's guidance to label the noise instance.

We then calculate a loss based on the labels assigned by both the student and the teacher, which is used to update the student's parameters. This iterative process helps enhance the student's ability to generate data that closely matches the real data distribution.

This method serves to rectify the student's errors, bringing its decision-making in closer alignment with that of the teacher and thereby reducing the performance gap between them. It's important to note that teacher labelling may not always be entirely accurate. Nevertheless, our goal is to have our student model mimic the teacher's performance precisely, so we treat the teacher as the ultimate "oracle."

The iterative process continues until the student's performance matches or surpasses that of the PD estimation models on the WT test set. This ensures that the student reaches a level of performance equivalent to the baseline models before concluding the training loop. It is noteworthy that This method will help the student to generalize well. In our case, student's parameters are updated only when the generated instance is labeled as fake which help update parameters only when the student is mistaken and adjust its data distribution. It is noteworthy that discriminator has a crucial role in this framework. If the latter is good, the outcomes will be promising. If not, outcomes will be frustrating. How to train then a good discriminator?

The discriminator is trained to predict either a generated instance from the generator (the student in our case) is real or fake. In other words, it's used to predict if a generated instance follows the real data distribution or not. To train it, we generate fake instances including the dependent variable (PD) then, we label it as fake (1). In the other hand, we label real training instances from the training dataset as real (0). Then we concatenate and shuffle fake and real instances and finally train the discriminator model.

Fake data should be a random noise, containing examples that are both far away from real data distribution and near to real data distribution to challenge the discriminator model training. The remaining question is how to construct fake data. The generation of fake data must be approached with care since the effectiveness of the discriminator hinges on the quality of this synthetic data. To accomplish this, we have considered conducting an initial training phase for the discriminator. This training phase allows us to understand the contribution of each feature, shedding light on how to construct fake data in a manner that effectively challenges the discriminator as shown in *Figure 25.* In cases where a feature carries significant

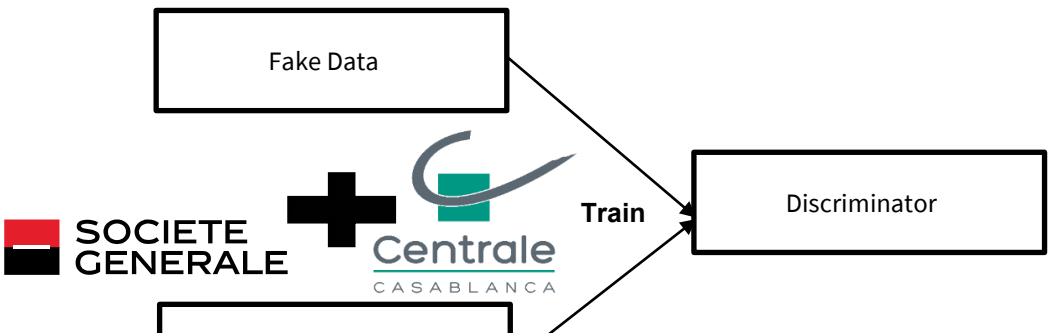


Figure 24. Discriminator Training Flow. Fake data is labeled as 1, and real data is labeled as 0. we train then the discriminator to distinguish real data from fake data.

weight in the model, it's essential to ensure that the corresponding fake data is meticulously crafted to pose a genuine challenge to the discriminator's discernment.

In our training dataset, we primarily have categorical features that are one-hot encoded, meaning they are binary indicators for various categories. However, there is one feature that stands out—it's continuous in nature. This continuous feature carries significantly more weight in our model, approximately six times greater than the contribution of the other categorical features. This means it plays a much more substantial role in the model's decision-making process.

Generating random noise for categorical features is relatively straightforward; we use a Bernoulli distribution with a probability of success set to 0.5 (p = 0.5) to create binary values. However, generating random noise for the continuous feature is a bit more complex and challenging.

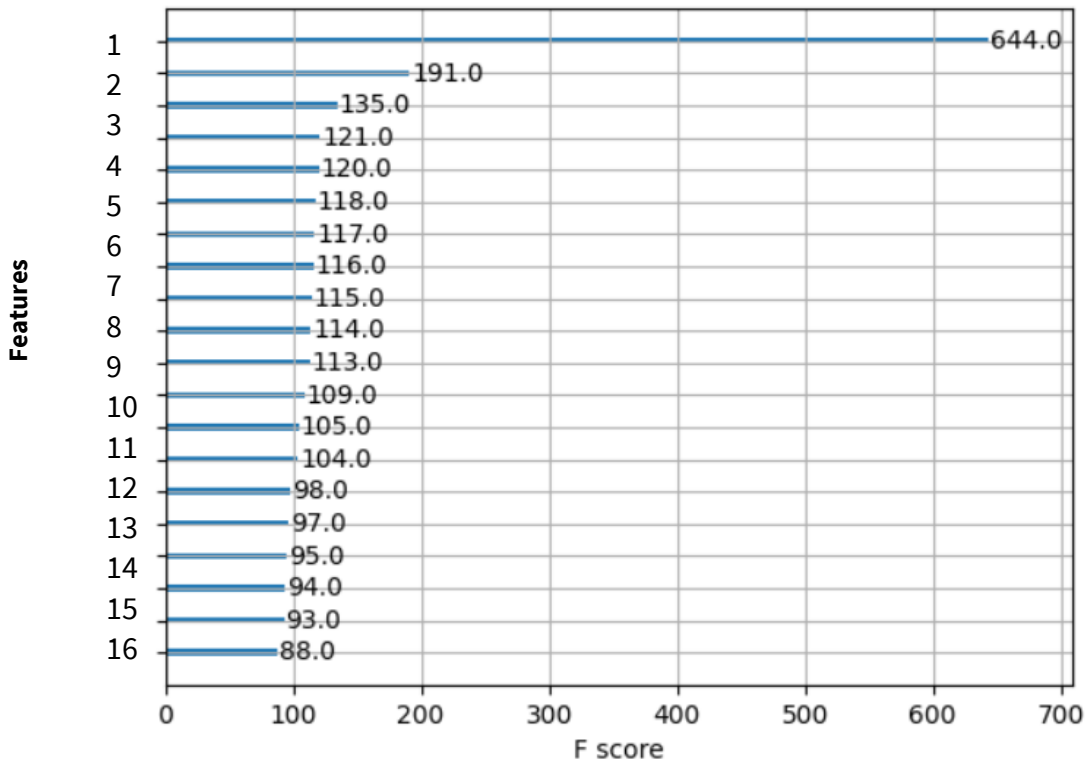


Figure 25. Preliminary discriminator (*LightGBM*) training. The continuous variable (1) has almost 6 times more importance comparing to other features.

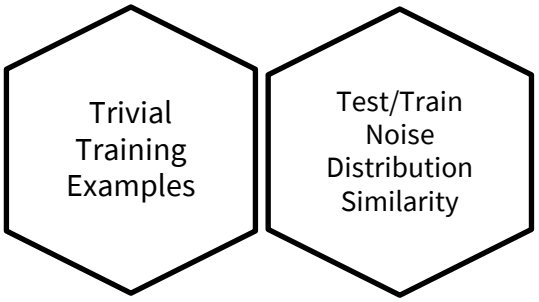


Figure 26. Constructing fake examples main challenges. This holds mainly for the continuous feature as it has the greatest contribution in the discriminator decision-making process.

- Trivial training examples: Since the continuous feature has a very consequent weight on discriminator's decision, constructing fake example of it following regular distributions can be a trivial task for the discriminator during test. It is easy for the model to discriminate fake and real data during test since he generalizes easily on the dataset. This led to biased performance metrics in test set. To tackle this issue, we construct the fake continuous feature from multiple distributions (either estimate the real distribution of data via kernel density method to make it a challenging task for the discriminator or use a mix of multiple random distribution)

- Test/Train Noise Distribution Similarity: Noise distribution used during training must be different (not strictly) from noise distribution during test especially for the continuous feature to avoid biased metrics. We want that the discriminator be able to discriminate fake data whatever is its distribution. To handle this issue, we will not use the same fake continuous feature's distribution during test phase.

The next step then is to construct an estimator of the probability distribution for the continuous feature in the training dataset with non-parametric estimation using kernel density estimation. The goal is the sampling of the noise from a close estimation of the probability distribution of the continuous variable in training dataset (real data) to avoid triviality and challenge the discriminator since the continuous variable is the most important feature of the model.

We use then the kernel density estimation of Parzen-Rosenblatt (1962):

$$\widehat{p_n^h}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where:

$\widehat{p_n^h}(x)$: is the estimated probability density at point x with bandwidth h.

K : is the kernel function, which depends on the choice of kernel (e.g., Gaussian, Epanechnikov).

X_i is the random variable which we want to estimate its distribution, in this case, the continuous variable and $x_1, x_2, \ldots\ldots, x_i, \ldots\ldots x_n$ the realization of the random variable X_i.

In our case, we will choose the normal gaussian kernel defined as:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$$

We will perform cross validation to determine the optimal bandwidth h. To do so, we compute the mean integrated squared error (MISE)[36]:

$$\text{MISE(h)} = \text{E}\left[\int \left(\text{p(x)} - \widehat{p_n^h}(x)\right)^2 \text{dx}\right]$$

$$= \text{E}\left[\int \left(\widehat{p_n^h}(x)\right)^2 - 2\,\widehat{p_n^h}(x)\text{p(x)} + \left(\text{p(x)}\right)^2 \text{dx}\right]$$

$$= \text{E}\left[\int \left(\widehat{p_n^h}(x)\right)^2 \text{dx}\right] - 2 \times \text{E}\left[\int \widehat{p_n^h}(x)\text{p(x)}\,\text{dx}\right] + \text{E}\left[\int \left(\text{p(x)}\right)^2 \text{dx}\right]$$

The goal is constructing an unbiased estimator of MISE(h) and then we minimize with respect to h. Since the last term does not depend on h, we will construct the unbiased estimator only for the first two terms. Let's denote:

$$\tau(h) = \text{E}\left[\int \left(\widehat{p_n^h}(x)\right)^2 \text{dx}\right] - 2 \times \text{E}\left[\int \widehat{p_n^h}(x)\text{p(x)}\,\text{dx}\right]$$

[36] *It can be seen as the total sum of MSE for all data point.*

- we have $\int \left(\widehat{p_n^h}(x)\right)^2 \mathrm{dx}$ is an unbiased estimator of $\mathrm{E}\left[\int \left(\widehat{p_n^h}(x)\right)^2 \mathrm{dx}\right]$

let's construct an unbiased estimator for the second term:

$$E_p\left[\int \widehat{p_n^h}(x)p(x)\,dx\right] = E_p\left[\int\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right)p(x)\;dx\right]$$

$$= \int E_p\left[\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right]p(x)\;dx$$

$$= \int\int \frac{1}{h}K\left(\frac{y - x}{h}\right)p(y)p(x)\;dy\,dx.$$

Using the leave one-out estimator leave-one out of p, let's denote:

$$\hat{G} = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{k\neq i}^{n}\frac{1}{h}K\left(\frac{X_k - X_i}{h}\right)$$

Which give this passing the expectancy

$$E_p(\hat{G}) = \frac{1}{n(n-1)}\sum_{k\neq i}\frac{1}{h}E_p\left[K\left(\frac{X_k - X_i}{h}\right)\right]$$

La loi jointe de (X_k, X_i) est $p(y)p(x)$ (Independent variables). So :

$$E_p(\hat{G}) = \frac{1}{h}\int\int K\left(\frac{y - x}{h}\right)p(y)p(x)\;dy\,dx$$

So $\hat{G}$ is an unbiased estimator of $\mathrm{E}\left[\int \widehat{p_n^h}(x)\mathrm{p(x)}\,\mathrm{dx}\right]$. Thus, the cross-validation objective is defined as:

$$CV(h) = \int \left(\widehat{p_n^h}(x)\right)^2 dx - 2\hat{G}$$

Thus,

$$h_{CV} \in arg\min_{h>0} C\,V(h)$$

In practice, it is straightforward to compute $\hat{G}$ by replacing K with normal gaussian kernel.

To compute $\int \left(\widehat{p_n^h}(x)\right)^2 dx$, we use the trapezoidal rule[37] to approximate the integral. The integral is finite and computed between the maximum and the minimum values of the continuous feature in the training dataset.

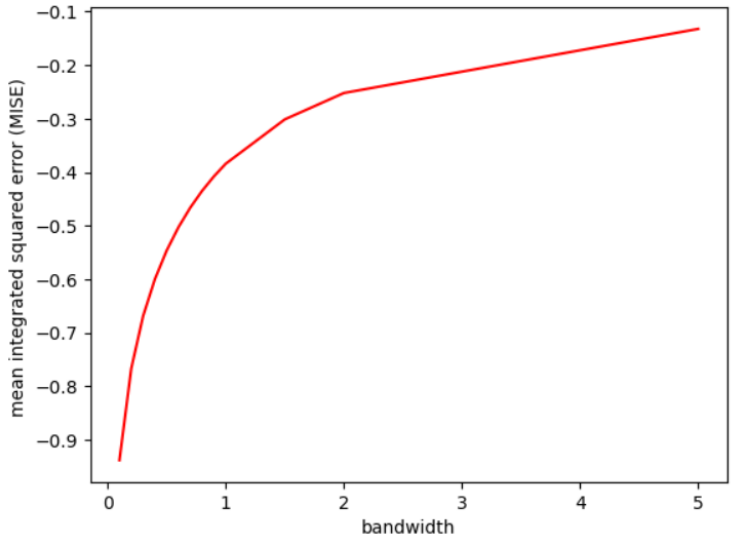


Figure 27. Cross validation to select the best bandwidth The values of h minimizing the MISE is 0.1.

We performed the cross-validation using the training data of the continuous feature. The goal is to select.

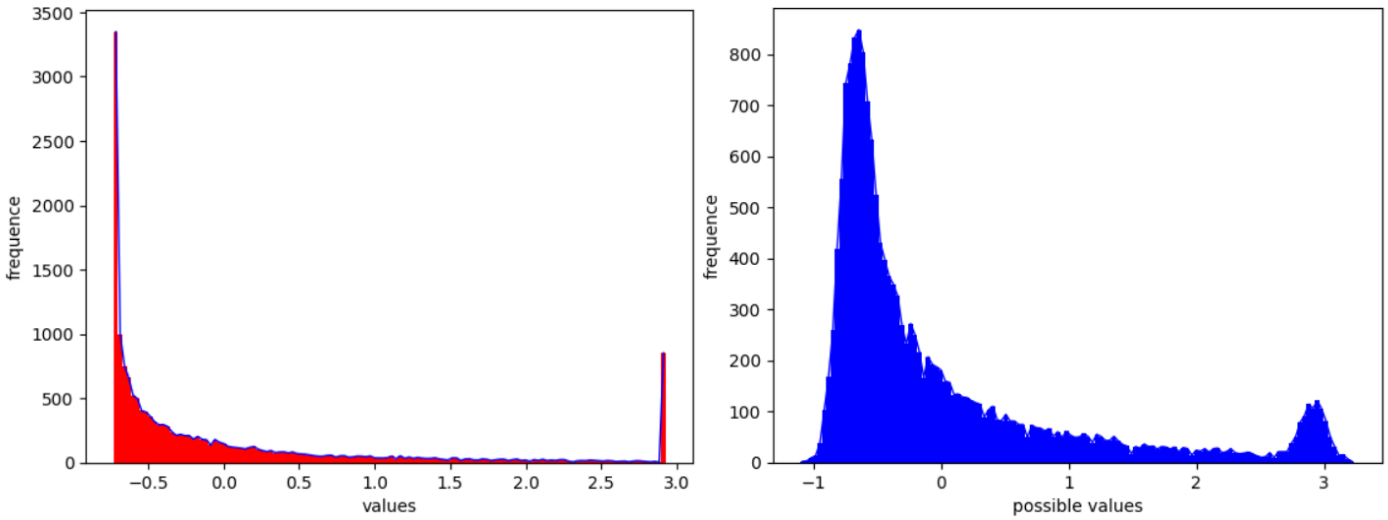


Figure 28. Continuous feature distribution in the training data Vs. the estimated distribution of the same feature using kernel density approximation method.

the best bandwidth between 0.1 and 5 minimizing the MISE. *Figure 28* showcases the successful achievement of our objective. Our crafted noise approximates the distribution of continuous features, but

[37] *numerical integration*

with different frequencies. Additionally, it extends to values beyond the training dataset, creating a mixture of possibly real and entirely fake instances. This presents a training challenge for our discriminator. While we've primarily focused on the continuous feature due to its utmost importance in the model, applying a similar approach to other features is optional in this scenario.

We proceed to train a *LightGBM* model, which will function as the discriminator. *Figure 29* illustrates the model's discrimination capability. It's worth mentioning that the discriminator begins to exhibit mild overfitting after 200 epochs, which is expected given the tree-based model training. Nonetheless, overall, the discriminator model remains satisfactory.

As previously mentioned, our goal is to ensure that the discriminator can effectively distinguish fake data from real data, regardless of the fake data's distribution. Given that the discriminator is trained on the kernel density approximation of the continuous feature and performs well on it, it may easily identify fake data during testing. This could result in biased metrics that do not accurately reflect the model's true discrimination capabilities.

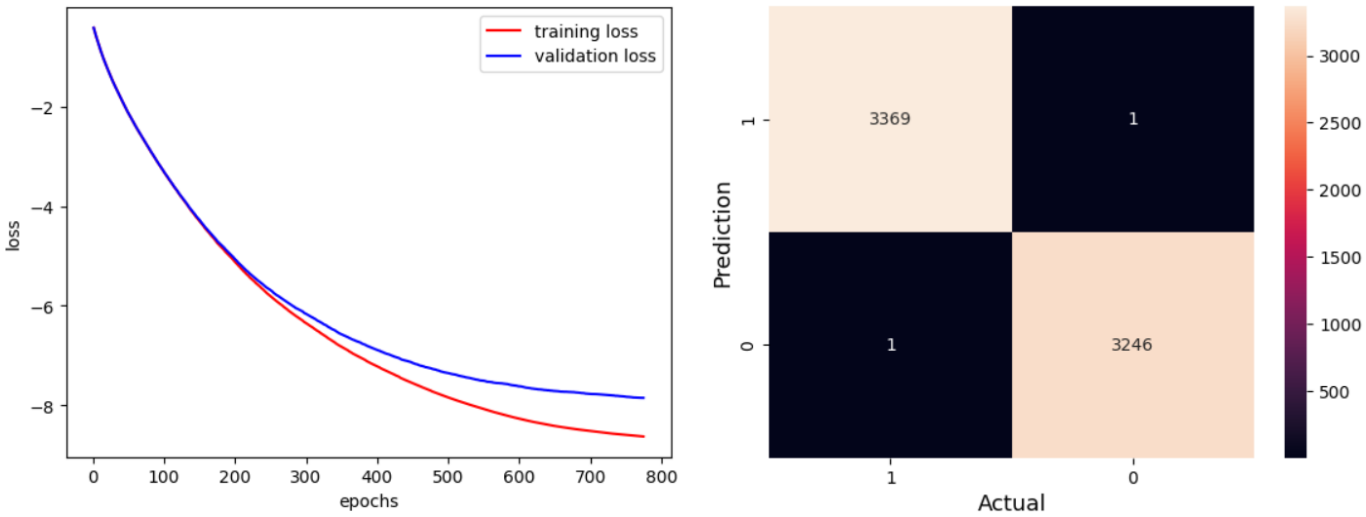


Figure 29. Discriminator *(LightGBM)* assessment. The learning curve (left figure) shows the **stability of convergence** of the discriminator and the **absence of any significant overfitting or underfitting** on almost 800 epochs. On the other hand, the confusion matrix demonstrates the discrimination power of the model since we have very **negligeable values of false positives and negatives**.

To address this concern, we will conduct tests using random noise (fake data) generated from a mixture of distributions, including the normal distribution and the previously used kernel density approximation. *Figure 30* displays the distribution of the continuous feature used during test phase. It's important to note that we should exclude instances from the kernel density approximation used during the training phase when conducting these tests.

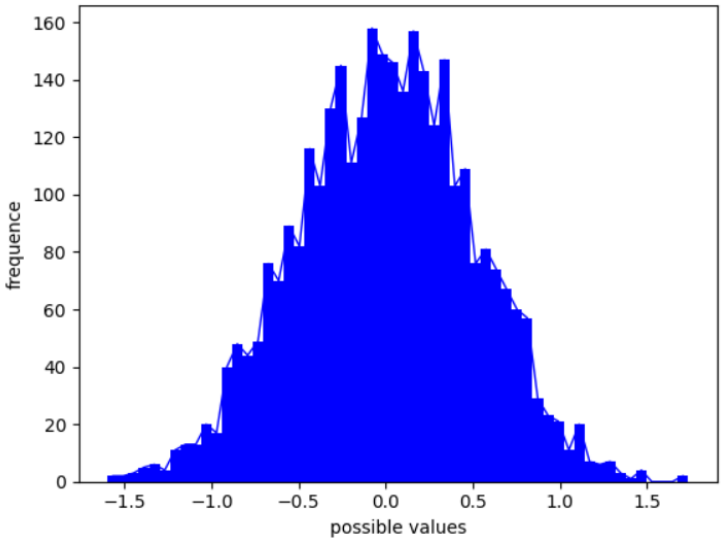


Figure 30. Noise distribution of continuous feature used during test.

Table 9 displays the outcomes achieved through the adversarial knowledge distillation framework, as depicted in Figure *23*, in contrast to the response-based distillation with temperature conducted in the preceding section. As anticipated, there is a noteworthy enhancement in performance on the WT test set. Specifically, we observe an accuracy ratio increase from *64.51%* in the response-based framework to *66.2%* with the inclusion of adversarial knowledge distillation, aligning with the performance of the baseline PD estimation models. It is s worth emphasizing that this represents the peak performance achieved on the WT test set, and further improvements appear unattainable.

Models	**Role**	**Training AR (%)**	**WT Test AR (%)**	**OOT Test AR (%)**
PD estimation models	Baseline	65,40	66,20	66,40
LightGBM with regularization	Teacher	70.87	67.41	71.44
PD estimation models distilled with ***response-based distillation*** **with temperature**	Student	68.63	64.51	70.83
PD estimation models distilled with ***response-based*** **with temperature +** *Adversarial* ***Knowledge Distillation***	Student	**68.99**	**66.20**	**72.14**

Table 9. PD estimation models students' performance using *Adversarial Distillation Framework* in addition to *Response-Based distillation.*

Interestingly, the student has demonstrated superior performance over the teacher on the OOT test set, with scores of *71.44%* versus *72.14%.* This can be attributed to the student being trained on a larger dataset. Upon closer examination of the framework depicted in Figure *23*, it's evident that the student is continually

updated with synthetic data labeled by the teacher in each iteration, while the teacher's parameters remain fixed throughout the adversarial knowledge distillation training process.

In summary, adversarial knowledge distillation has proven to be an effective approach for minimizing the performance gap between the student and teacher models. This improvement has resulted in enhanced performance on the OOT test set and parity with baseline PD estimation models on the WT test set. In the upcoming section, we will delve into the X-Distillation framework and assess its impact on performance in comparison to the previous frameworks.

X-Distillation Framework

In continuation of the previous efforts to bridge the performance gap between the student's and teacher's models, this section will delve into the implementation of *Explicability Distillation* also known as *X-Distillation*. There are several methods to carry out *X-Distillation*. One approach involves incorporating the teacher's internal layer feature maps as new features within the student model. Alternatively, we can employ a post-hoc explanation metric to extract explanations from the teacher and instruct the student to replicate them.

In the context of enhancing PD estimation models, we are bound by the necessity to retain the same features employed by the credit’s risk team. Moreover, if the teacher intervenes during a new inference, the distilled PD estimation models may lose interpretability, given the introduction of new features from a black-box teacher.

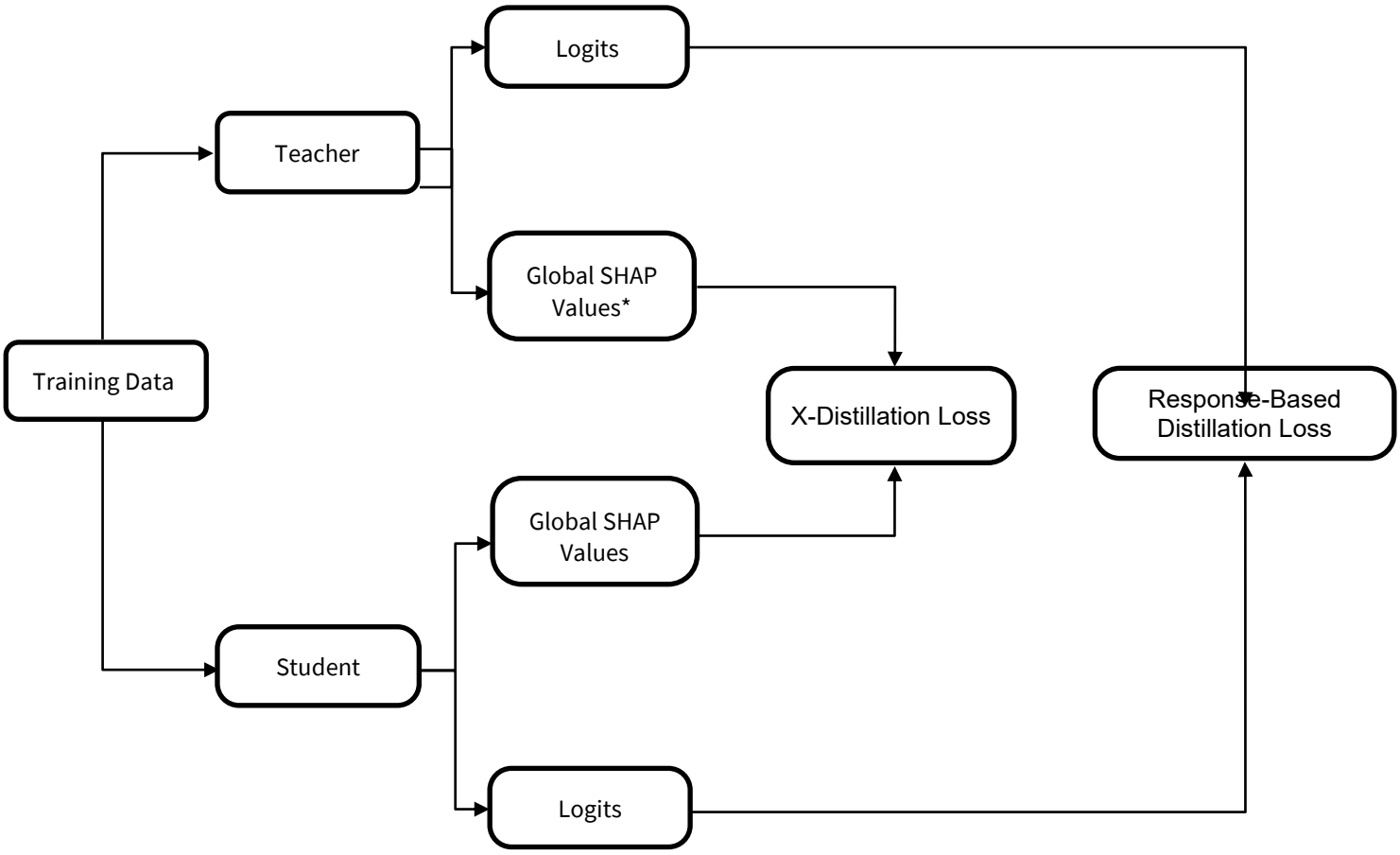


Figure 31. X-Distillation framework used to enhance PD estimation models.

Figure 31 illustrates the *X-Distillation* framework utilized to enhance PD estimation models. It's important to note that, akin to the *Adversarial Knowledge Distillation* framework, the *X-Distillation* framework runs concurrently with *Response-Based Distillation*.

In the process of *X-Distillation*, we initiate by extracting the global *Shapley* values for each feature from the teacher model. It's worth mentioning that the *Shapley* values approach is a local explanation method, meaning that for each instance, we calculate a contribution score for each feature. The global *Shapley* values for a specific feature 'i' represent the absolute mean value across all instances involving feature 'i' in the training dataset. At the same time, we compute the student's global Shapley values for each feature. To update the student's parameters, we calculate the following loss function during each iteration:

$$L_{XD} = \frac{1}{|N|}\sum_{i=1}^{|N|}(\psi_i^S - \psi_i^T)^2$$

Formula 9. X-Distillation Loss

where ψ_i^S : Global Shapley value of feature 'i' using the student model; ψ_i^T: Global Shapley value of feature 'i' using the teacher model; N : The set of features of independent variables.

To update our student's (Logistic Regression) parameters β_i. $\boldsymbol{\psi_i^S}$ must be expressed as a function of $\boldsymbol{\beta_i}$ in order to perform gradient descent. However, calculating the theorical expression of ψ_i^S as a function of logistic regression parameters is not straightforward.

To do so, we start from the expression of the *Shapley* values kernel explainer which is expressed as:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

Formula 10. *Shapley* values kernel explainer

Where f : the model we want to explain; $\phi_i(f)$: Local *Shapley* value for the instance 'i'; S : the set of a possible coalition within all features except 'i'; N : The set of features of independent variables; $f(S \cup \{i\})$: instance 'i' prediction using the model trained only on the set $S \cup \{i\}$; $f(S)$: Instance 'i' prediction using the model trained only on the set S.

In our case, f is a linear regression model[38]. Let x_j be an instance vector, and v a specific combination within $|S|$ elements defined as $v(X,S) = \{ X \,/\, X \subseteq S \ and\ |X| = |S| \}$. $\forall\, S \subseteq N \setminus \{i\}$, we have:

$$f(S \cup \{i\})(x_j) = \beta_0^i + \beta_{v(1)}^i x_{v(1),j} + \beta_{v(2)}^i x_{v(2),j} + \ldots\ldots\ldots\ldots\ldots + \beta_i^i x_{i,j} + \ldots\ldots\ldots + \beta_{v(|S|)}^i x_{v(|S|),j}$$

and,

$$f(S)(x_j) = \beta_0^{-i} + \beta_{v(1)}^{-i} x_{v(1),j} + \beta_{v(2)}^{-i} x_{v(2),j} + \ldots\ldots\ldots\ldots\ldots + \beta_i^{-i} x_{i,j} + \ldots\ldots\ldots + \beta_{v(|S|)}^{-i} x_{v(|S|),j}$$

Where $\beta_{v(k)}^i$ are the coefficient obtained by training a linear regression model on $S \cup \{i\}$ and $\beta_{(k)}^{-i}$ are the coefficient obtained by training a linear regression model on S. Given:

$$f(S \cup \{i\})(x_j) = \sum_{k \neq i}^{|S|} \beta_{v(k)}^i x_{v(k),j} + \beta_i^i x_{i,j} \quad \text{and} \quad f(S)(x_j) = \sum_{k \neq i}^{|S|} \beta_{v(k)}^{-i} x_{v(k),j}$$

Local *Shapley* value of a feature 'i' is simplified as:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[\sum_{k \neq i}^{|S|} (\beta_{v(k)}^i - \beta_{v(k)}^{-i})\, x_{v(k),j} + \beta_i^i x_{i,j} \right]$$

[38] *Using a linear regression model on transformed labels as explained in Formula 8 is equivalent to perform a logistic regression.*

Denoting $C_1^S = \frac{|S|!(|N|-|S|-1)!}{|N|!}$ and $C_{2,i}^S = \sum_{k\neq i}^{|S|}(\beta_{v(k)}^i - \beta_{v(k)}^{-i})\, x_{v(k),j}$. In this case, we have:

$$\phi_i(f) = \sum_{S\subseteq N\setminus\{i\}} C_1^S \left[C_{2,i}^S + \beta_i^i\, x_{i,j}\right]$$

As a first approximation (1), let's assume that $C_{2,i}^S = 0$, which means that $\beta_{v(k)}^i = \beta_{v(k)}^{-i}$. In other words, this avoid us to Train f for each coalition S which is a cumbersome task. Instead, we attribute the value zero to features excluded from S.

To compute global *Shapley* values for each feature 'i' in the dataset, we take the mean absolute value of $\phi_i(f)\,\forall\, i \in N$. The motivation behind taking the absolute mean is that we are interested in *Shapley* values magnitude, and we do not want any compensation effect between positive and negative values due to the mean summation. Let's denote ψ_i^S the global *Shapley* value of feature 'i' using the student model. We have:

$$\psi_i^S = E\,|\phi_i(f)| = \sum_{S\subseteq N\setminus\{i\}} C_1^S \times E_j[\,|\,\beta_i^i\, x_{i,j}\,|\,]$$

$$\psi_i^S = |\beta_i| \times \sum_{S\subseteq N\setminus\{i\}} C_1^S \times E_j|\,x_{i,j}\,|$$

Formula 11. Global Shapley values using approximation (1)

because $\sum_{S\subseteq N\setminus\{i\}} C_1^S > 0$ and β_i is not a random variable depending on instances.

The second approximation (2) arises when computing the sum in *Formula 11.* Since we have 61 features, summing over all coalition is very cumbersome in term computing complexity as the total number of coalitions is $\sum_{k\geq 1}\binom{61}{k} = 2^{61} - 1$ which is a very high number of iterations. To avoid this issue, we sum up only over 10,000 coalitions picked randomly.

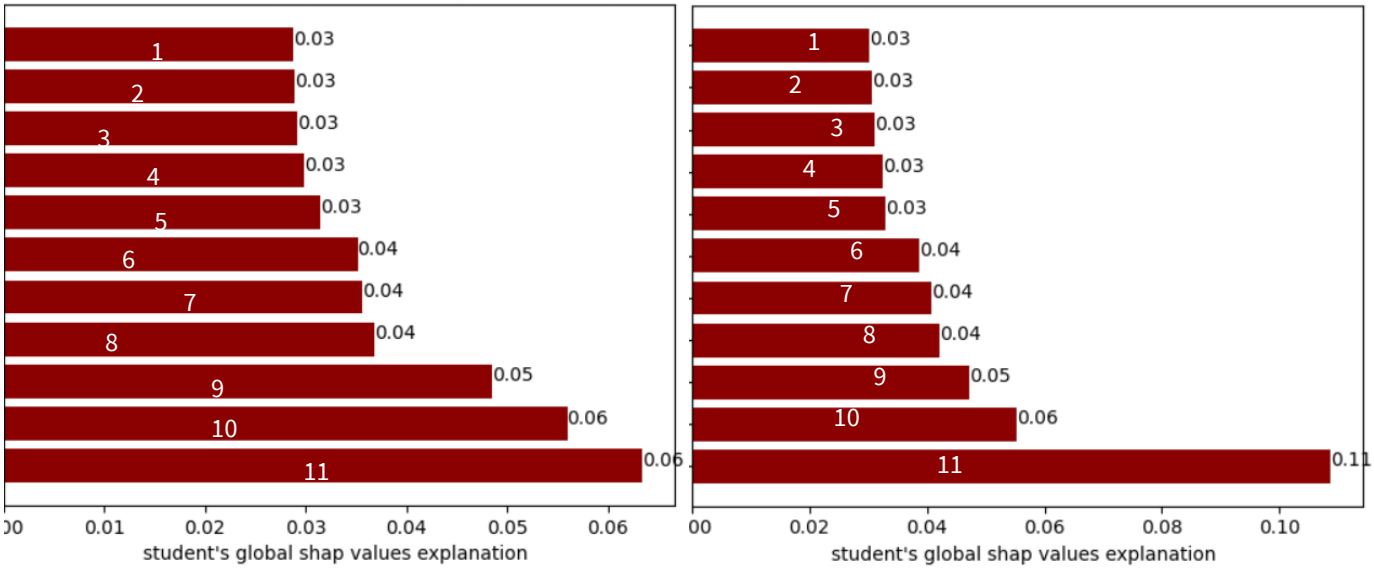


Figure 32. On the right, we have student's global *Shapley* values using SHAP library in Python Vs. Student's global Shapley values using kernel explainer with approximations on the left.

To validate our approximation, we conducted a comparison between the Shapley values computed using the *Formula 11* with approximation (2) and those computed using the *SHAP* library in Python. As depicted in *Figure 32*, our approximated *Shapley* values align closely with those obtained from the *SHAP* library for the 10 most significant variables. This observation leads us to conclude that *Formula 11* provides an accurate method for computing *Shapley* values, yielding results similar to those obtained using the *SHAP* library.

One might question the necessity of computing an approximation of *Shapley* values when there is already a well-established implementation available in the *SHAP* library. The reason behind this lies in our specific objective. We not only want to calculate *Shapley* values for the student model's features but also update the student model's parameters based on the loss defined in *Formula 9* . To achieve this, we need to establish a relationship between the student's Shapley values and its parameters. However, finding a straightforward formula for this relationship can be challenging, leading us to rely on approximations.

Figure 32 provides evidence that, despite these approximations, we are able to accurately compute the student's *Shapley* values, demonstrating the effectiveness of our approach.

Now that we have the expression of the *Shapley* values of our linear model f expressed using the coefficients β_i, we can calculate the gradient of *X-Distillation* loss between student's Shapley values and

teacher Shapley values in order to update student parameters. Given the X-Distillation loss in *Formula 9* and denoting $Cte_s^i = \sum_{S \subseteq N \setminus \{i\}} C_1^S \times E_j \left| x_{i,j} \right|$, The X-Distillation can be expressed as:

$$L_{XD} = \frac{1}{|N|} \sum_{i=1}^{|N|} (Cte_s^i \times |\beta_i| - \psi_i^T)^2$$

The gradient of the absolute value function is not well-defined at zero because it's a non-smooth function at that point. However, we can compute sub-gradients of the absolute value when $\beta_i > 0 \; and \; \beta_i < 0$. thus, we have:

$$\frac{\partial L_{XD}}{\partial \beta_i} = \frac{2}{|N|} \times (-1)^{\mathbb{1}(\beta_i < 0)} \times Cte_s^i \times (|\beta_i| \times Cte_s^i - \psi_i^T)$$

Equation 12. X-Distillation Loss's Gradient

The gradient descent optimization will be then as the following:

$$\beta_{i,m+1} = \beta_{i,m} - \alpha \times \frac{\partial L_{XD}}{\partial \beta_i}$$

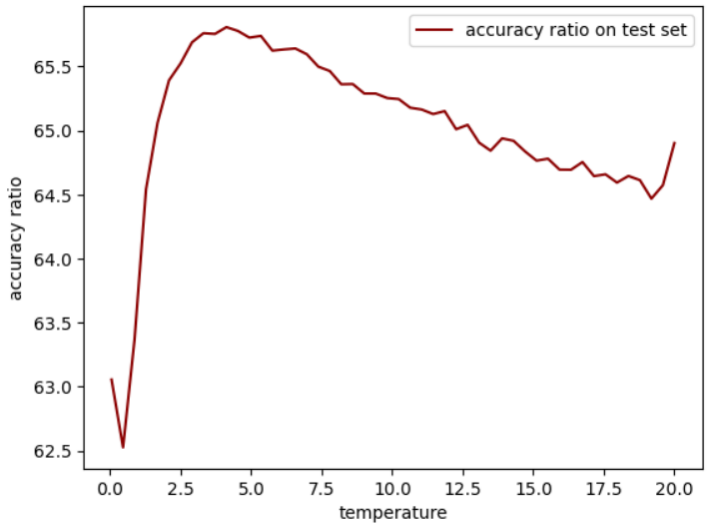


Figure 33. Cross validation for best temperature T selection for response-based distillation in the X-Distillation framework. Here T = 4.12

In the results *(*

Table 10), the *X-Distillation* framework has demonstrated superior performance compared to other frameworks on the OOT test set, achieving a score of ***72.12%*** versus the ***72.14%*** achieved by the best performer. Additionally, similar to *adversarial knowledge distillation*, the PD estimation models student distilled with *X-Distillation* outperforms the teacher on the OOT test set. This is attributed to the fact that the student model is trained using a richer source of knowledge, including teacher's soft labels, teacher's explanations, and ground truth labels, which helps it align with the true data distribution.

However, it's worth noting that the *X-Distillation* student performs slightly worse than the *Adversarial Knowledge Distillation* student (***66.02% versus 66.2%***). This discrepancy can be attributed to the role of the discriminator in the latter framework, which guides the student during the distillation process, in contrast to the *X-Distillation* framework where the student learns autonomously from the teacher's explanations without external guidance.

As a potential avenue for future work, we can explore the possibility of performing *X-Distillation* on the student obtained from *Adversarial Knowledge Distillation*. This approach could provide the student with information from different sources of knowledge, potentially yielding further improvements in performance.

Models	**Role**	**Training AR (%)**	**WT Test AR (%)**	**OOT Test AR (%)**
PD estimation models	Baseline	65,40	66,20	66,40
LightGBM with regularization	Teacher	70.87	67.41	71.44
PD estimation models distilled[39] with ***response-based distillation*** only without temperature	Student	68.03	63.94	70.13
PD estimation models trained with ***response-based distillation*** only with temperature	Student	68.63	64.51	70.83
PD estimation models distilled *with* ***response-based distillation*** *with temperature* + ***adversarial knowledge distillation***	Student	68.99	66.20	72.14
PD estimation models trained with ***Response-Based Distillation*** with temperature + ***X-Distillation***	Student	**70.09**	**66.01**	**72.78**

Table 10. Summary comparison of X-Distillation framework compared to *Response-Based* and *Adversarial Knowledge Distillation* frameworks.

To summarize our efforts in PD estimation models distillation, we employed three distinct frameworks: *response-based distillation, adversarial knowledge distillation,* and *X-Distillation,* all with the aim of improving the performance of PD estimation models, as detailed above. We systematically assessed each framework's impact on enhancing the performance of our baseline PD estimation models.

[39] *Logistic regression student using features concatenated from all modules.*

The results have been encouraging. We achieved significant performance improvements on the out-of-time (OOT) test set, with an impressive ***6.38%*** increase in accuracy ratio. However, it's noteworthy that our results on the within-time (WT) test set remained consistent with the baseline, at ***66.2%.***

These findings underscore the effectiveness of the distillation frameworks in enhancing the model's ability to handle out-of-time data. While the performance on the within-time test set remains unchanged, there may be opportunities for further refinement to improve results in this aspect.

It's important to highlight that the dataset of France's SMEs used for distillation presents several challenges. The use of one-hot encoding introduces sparsity into the training data, which can pose difficulties for a logistic regression student model in contrast to tree-based models like the teacher model. Logistic regression relies on linear combinations of features, and when dealing with high-dimensional, sparse data, it may struggle to identify meaningful patterns effectively.

Moreover, the assumption of independence between instances in logistic regression training doesn't hold in this context. The dataset comprises panel data with repeated observations of the same obligor over time. This temporal dependence violates the independence condition, potentially leading to undesirable and biased model performance.

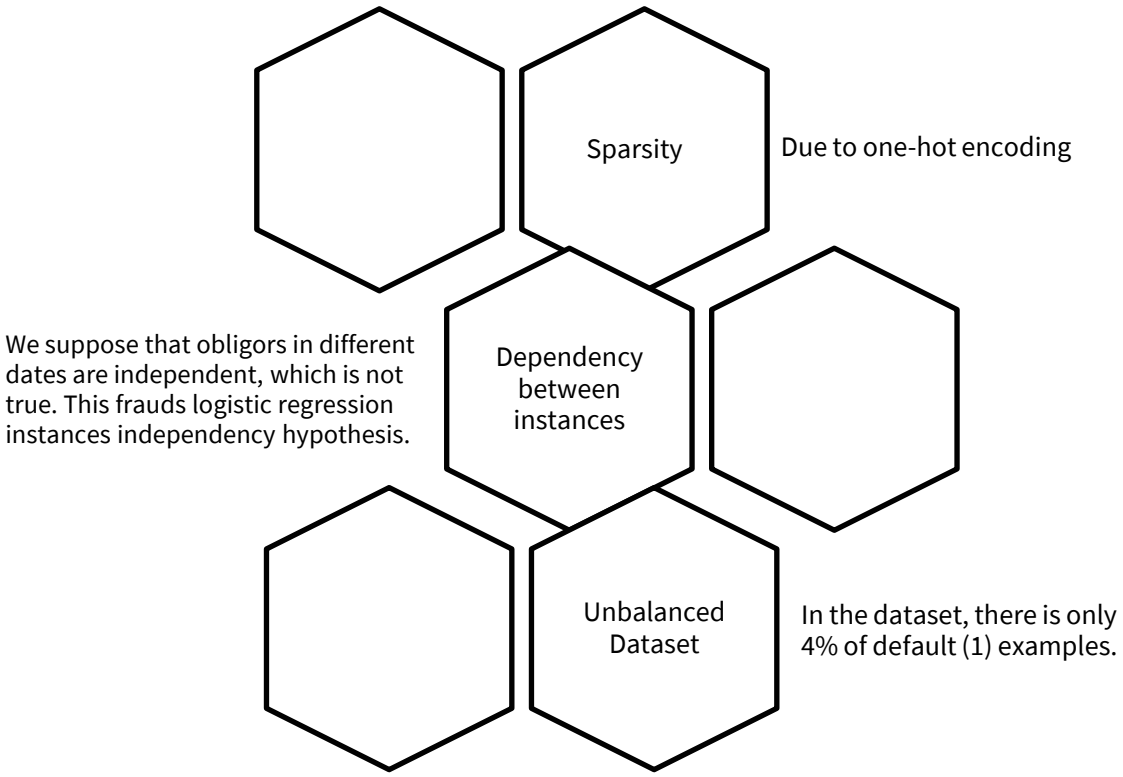


Figure 34. Main problems in *PD estimation* models training data.

Additionally, the dataset's imbalance, where one class is significantly underrepresented, poses a challenge for both the student and the teacher model. Imbalanced datasets can lead to difficulties in generalizing well on the class labeled as 1, potentially resulting in models that are biased toward the latter class.

During the distillation process, we made efforts to address certain challenges, such as the dataset's imbalance, by assigning more weight to class 1 instances during the training of both the teacher and the student models. However, we intentionally left the other issues, like the sparsity introduced by one-hot encoding and the temporal dependencies in the data, untouched. This decision was made to ensure that the assessment of the distillation's impact on performance remained isolated and uncontaminated by other factors.

In essence, the goal was to evaluate the specific effect of distillation on the performance of PD estimation models independently of any other potential factors that could enhance model performance. This approach allows for a more accurate assessment of the distillation's contribution to model improvement.

For future work, we propose replicating the same distillation approaches but using a decision-tree student model. This approach can help mitigate some of the challenges mentioned earlier, particularly those related to the logistic regression assumptions and sparsity in the data due to one-hot encoding. Using a decision-tree-based student model can be advantageous in several ways:

- Independency Assumption: Decision trees do not rely on the assumption of independence between instances, making them better suited for handling panel data.

- Handling Sparsity: Decision trees can naturally handle sparse data and do not suffer from the same challenges as logistic regression when dealing with one-hot encoding.

- Non-linearity: Decision trees can capture non-linear relationships between features and the target variable, which can be valuable when dealing with complex data patterns.

By employing a decision-tree-based student model, future work can assess the effectiveness of distillation while circumventing some of the limitations associated with logistic regression, providing a more comprehensive evaluation of the distillation approaches.

Chapter 5: DISTILLATION TESTS ON LENDING CLUB DATASET

In this chapter, we will explore additional distillation frameworks that were not previously examined with the France's SMEs dataset in the context of PD estimation models. The primary aim is to evaluate the efficacy of the *Multi-Teacher Distillation* and *Feature-based* knowledge distillation techniques. Additionally, we endeavor to distil a neural network teacher model into a neural network student model with reduced parameters, utilizing tabular data typically employed within the banking domain.

To accomplish this, we leverage the *Lending Club*[40] dataset, which provides detailed information about previous loan applicants and their **loan** repayment behavior, distinguishing between those who **defaulted** and those who did not.

The task at hand is **a binary classification problem** where the model's objective is to predict whether a new loan applicant is more likely to default (1) or not (0). To mitigate the risks mentioned above, we choose **F1-Score** and **ROC AUC** as the evaluation metrics.

Teacher Offline Training

[40] *Open-source dataset, more information is provided in appendix*

As initial step, we commence by training a teacher model using the *Lending Club* training dataset. For the teacher models, we have opted to employ a *feed-forward neural network* and *XGBoost*. As previously mentioned, the ultimate objective is to perform model distillation in a neural network with fewer parameters. Furthermore, we train the *XGBoost* teacher model to participate in the multi-teacher distillation framework.

Figure 35 illustrates the training behavior of the teacher models. It is evident that the *XGBoost* model demonstrates a more stable performance when compared to the neural network model. The neural network model exhibits minimal learning progress over epochs, as indicated by the consistently flat training loss, whereas the *XGBoost* model shows improvement by reducing its training loss from a higher initial value.

Despite the anomalies observed in the feed-forward neural network, we have chosen it as the teacher model. This decision is driven by the desire to conduct *feature-based* distillation technique, in contrast to the previous chapter where the primary focus was on enhancing PD estimation models.

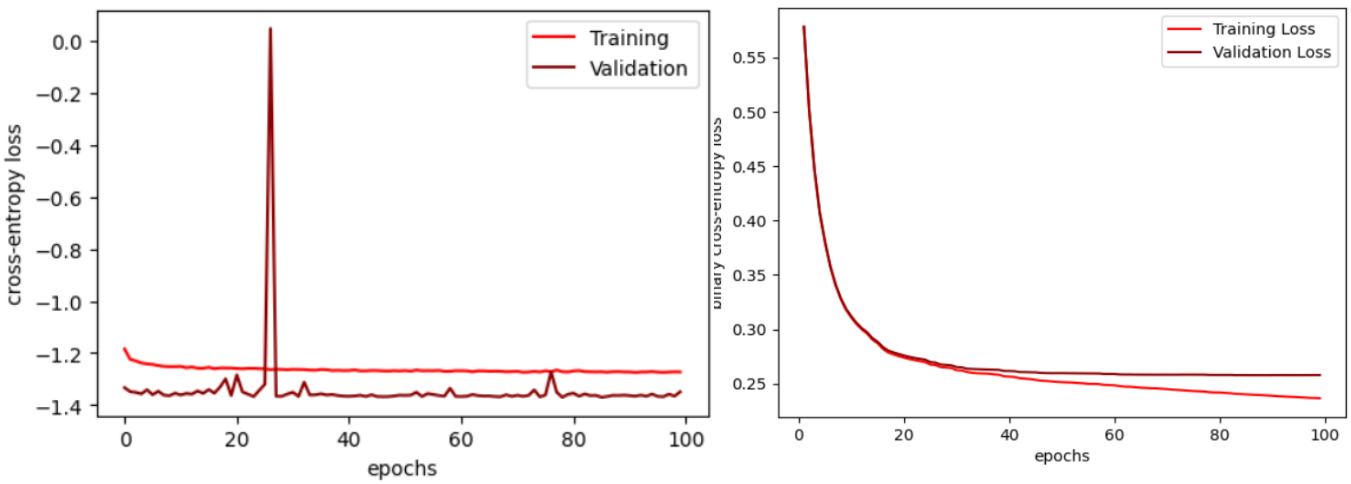


Table 11. Teachers' performance using *ROC AUC* and *F1-Score* metrics. Despite similarity in performance, *XGBoost* model remains more reliable due to its training stability behavior.

Figure 35. Teachers offline training on Lending Club data. The *XGBoost* model exhibits more stable training (right figure) over 100 epochs.

Table 11 provides a summary of the performance of the teacher models, revealing that both models exhibit comparable performance on both the training and test sets. Notably, the ROC AUC metric serves as a robust indicator of model performance as it remains independent of a specific threshold.

However, it's important to emphasize that the *F1-Score* remains the primary objective metric within the context of risk management, as it considers the balance between precision and recall, which is critical for assessing the model's effectiveness in mitigating risks associated with loan default predictions.

In the upcoming sections, we implement *Feature-based* and *multi-Teacher* distillation frameworks employing a student network with significantly reduced parameters. As shown in *Table 12*, the student model's parameters have been reduced by a factor of 51.41 compared to the teacher neural network model.

Student Distillation Training

We conduct three types of distillation: *response-based*, *feature-based*, and *multi-teacher* distillation. As previously mentioned, the student model is a reduced *feed-forward neural network* with its number of parameters reduced by a factor of **51.41**. In the response-based and feature-based distillation approaches, we employ feed-forward neural networks as teachers. In the multi-teacher distillation framework, we utilize both the feed-forward neural network and the XGBoost teacher model.

Model	Role	Training ROC AUC (%)	Training F1-Score (%)	Test ROC AUC (%)	Test F1-Score (%)
Feed-Forward Neural Networks	Teacher	90.96	62.81	90.65	62.49
XGBOOST	Teacher	92,86	65.10	90.73	62.70

Model	#Parameters
Teacher	58,357
Student	1,135

Table 12. Number of parameters in the student and the teacher models. Student's number of parameters is reduced by a factor 51.41.

Feature-based distillation involves the process of distilling the intermediate layers of the teacher model using intermediate losses, as elaborated in *Figure 6*. In practical terms, this means incorporating the feature representations from each layer of the teacher model into the feature training space of the student model. In the case of multi-distillation, it entails response-based distillation by averaging the predictions of both the *XGBoost* and *FFNN* teacher models in the student's response layer.

Table 13 presents the outcomes of the distillation frameworks, revealing a noticeable performance gap between the student and teachers, particularly in the response-based and feature-based frameworks. It's worth noting that feature-based distillation performs slightly less effectively than response-based distillation. This difference can be attributed to the introduction of aggregated noise through the intermediate layers in feature-based distillation. In contrast, multi-teacher distillation achieves better results, primarily owing to the reliability and stability offered by *XGBoost*.

Model	Role	Training ROC AUC (%)	Training F1-Score (%)	Test ROC AUC (%)	Test F1-Score (%)
FFNN[41]	Teacher	90.96	62.81	90.65	62.49
XGBOOST	Teacher	92,86	65.10	90.73	62.70
FFNN trained ***Response-Based Distillation***	Student	71.45	59.60	71.48	58.01
FFNN trained with ***Feature-Based Distillation***	Student	71.44	59.36	71.47	57.59
FFNN trained with ***Multi-Teacher Distillation***	Student	88.95	63.64	88.97	61.14

Table 13. Student's performance using response-based, feature-based, and multi-teacher distillation on *Lending Club* dataset.

[41] *Feed-Forward Neural Network*

Conclusion

During this internship's research work, we delved into the valuable concept of machine learning known as distillation learning. This model compression technique proves to be potent and finds application in several domains. Firstly, it is employed to reduce model complexity, thereby decreasing inference time, accommodating computational resource limitations, and facilitating deployment on edge devices. Secondly, distillation learning is instrumental in *Explainable Artificial Intelligence* (XAI). It helps enhance global interpretability by distilling insights from black-box models into interpretable student models. Lastly, distillation techniques serve to enhance overall model performance. Training through distillation proves to be more effective than traditional training methods, particularly when a well-performing teacher model is available.

We conducted a comprehensive literature review of knowledge distillation techniques including distillation concepts, useful frameworks, existing packages, and most relevant applications in machine learning. Subsequently, we applied distillation techniques to evaluate their suitability for the banking context. Specifically, we utilized distillation frameworks to improve models for estimating *Probability of Default (PD)*, focusing on credit risk.

We observed significant performance enhancements through various distillation methods, including *Response-Based* distillation, *Adversarial Knowledge Distillation*, and *X-Distillation*. We assessed the individual contributions of each technique in terms of performance and conducted a comparative analysis of its strengths and drawbacks compared to other frameworks.

Finally, we applied additional distillation techniques using the Lending Club dataset. Specifically, we explored *Feature-Based* distillation and *Multi-Teacher Distillation*, examining their contributions, practicality, and limitations to gain a comprehensive understanding of their contribution.

While distillation has numerous applications in image recognition, speech recognition, and natural language processing, the primary focus in this work is on discussing the fundamental principles, which are applicable across various domains. It's important to note that the underlying concepts remain consistent regardless of the specific application.

We believe that distillation learning still represents a promising research area with room for improvement, particularly in addressing the challenge of narrowing the performance gap between the teacher and the student models. The ultimate goal is to train a student model that surpasses or at least matches the teacher's performance.

REFERENCES

Adriana Romero, N. B. et al., 2015. FitNets: Hints for Thin Deep Nets. *ICLR,* p. 13.

Alharbi, R., Vu, M. N. & Thai, M. T., 2021. Learning Interpretation with Explainable Knowledge Distillation. *IEEE.*

Angwin, J., Larson, J., Mattu, S. & Kirchner, L., 2016. *Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks.*

Available at: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
[Accessed 17 08 2023].

Borup, K., 2020. *Knowledge Distillation.* [Online] Available at:
https://keras.io/examples/vision/knowledge_distillation/
[Accessed 18 09 2023].

Caruana, R., Bucila, C. & Niculescu-Mizil, A., 2006. Model Compression. *KDD.*

Caruana, R. et al., 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *KDD.*

Chen, X., Su, J. & Zhang, J., 2019. A Two-Teacher Framework for Knowledge Distillation. *Advances in Neural Networks – ISNN* .

Che, Z., Purushotham, S., Khemani, R. & Liu, Y., 2015. Distilling Knowledge from Deep Networks with Applications to Healthcare Domain. p. 11.

Chung, I., Park, S., Kim, J. & Kwak, N., 2020. Feature-map-level Online Adversarial Knowledge Distillation. *ICML.*
Dai, Z. et al., 2019. Transformer-XL: Attentive language models beyond a fixed-length context.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Dosilovi, Karlo, F., Brcic, M. & Hlupic, N., 2018. Explainable artificial intelligence: A survey. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics,* pp. 210-215.
Gou, J., Yu, B., Maybank, S. J. & Tao, D., 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision,* p. 36.

Hinton, G., Vinyals, O. & Dean, J., 2015. Distilling the Knowledge in a Neural Network. *NIPS 2014 Deep Learning Workshop,* p. 9.

Johnson, A. E. et al., 2016. *MIMIC-III, a freely accessible critical care database.* [Online]
Available at: https://www.nature.com/articles/sdata201635
[Accessed 17 08 2023].

Khemani, R. G. et al., 2009. Effect of tidal volume in children with acute hypoxemic respiratory failure. *Intensive care medicine,* pp. 1428-1437.

LeCun, Y., Cortes, C. & Burges, C. J., 1998. *THE MNIST DATABASE of handwritten digits.* [Online]
Available at: http://yann.lecun.com/exdb/mnist/
[Accessed 17 08 2023].

Liu, J. et al., 2019. Knowledge Representing: Efficient, Sparse Representation of Prior Knowledge for Knowledge Distillation. *Computer Vision and Pattern Recognition,* p. 9.

Liu, X., Wang, X. & Matwin, S., 2018. Improving the Interpretability of Deep Neural Networks with Knowledge Distillation. *IEEE International Conference on Data Mining (ICDM),* p. 7.

Lou, Y., Caruana, R. & Gehrke, J., 2012. Intelligible models for classification and regression. *KDD.*

Lou, Y., Caruana, R., Gehrke, J. & Hooker, G., 2012. Accurate intelligible models with pairwise interactions. *KDD.*

Molnar, C., 2023. *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable.* [Online] Available at: https://christophm.github.io/interpretable-ml-book/ [Accessed 18 08 2023].

Mullenbach, J. et al., 2018. Explainable prediction of medical codes from clinical text. *Conference of the North American Chapter of the Association for Computational Linguistics,* Volume 1, p. 1101–1111.

Radford, A. et al., 2019. Language models are unsupervised multitask learners. *OpenAI blog.*

Rudin, C., 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence,* Volume 1, p. 206–215.

SANH, V., DEBUT, L., CHAUMOND, J. & WOLF, T., 2020. DistilBERT, a distilled version of BERT: smaller,faster, cheaper and lighter. *Hugging Face.*

Takamoto, M., Morishita, Y. & Imaoka, H., 2020. An efficient method of training small models for regression problems with knowledge distillation.

Tang, R., Lu, Y. & Lin, J., 2019. Natural Language Generation for Effective Knowledge Distillation. *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP,* p. 202–208.

Tan, S., Caruana, R., Hooker, G. & Lou, Y., 2018. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation.

Wood-Doughty, Z., Cachola, I. & Dredze, M., 2021. Faithful and Plausible Explanations of Medical Code Predictions.

Xu, Z., Hsu, Y.-C. & Huang, J., 2018. Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks. *ICLR Workshop.*

Yim, J., Joo, D., Bae, J. & Kim, J., 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 13.

Zhang, H. et al., 2021a. Adversarial co-distillation learning for image recognition. *Pattern Recognition,* p. 111.

APPENDIX

RISQ/MRM Department

Model risk management (RISQ/MRM) team ensures the quality and relevance of models developed within the SG group, as well as monitoring the bank's models risks. The team is also responsible for carrying out benchmarks such as stress tests to test the limits of the audited entities' models. By

subjecting models to various scenarios and stressors, the team gains insights into the models' behavior and robustness under different conditions. This process is indispensable for ensuring that the models are capable of withstanding real-world challenges and uncertainties.

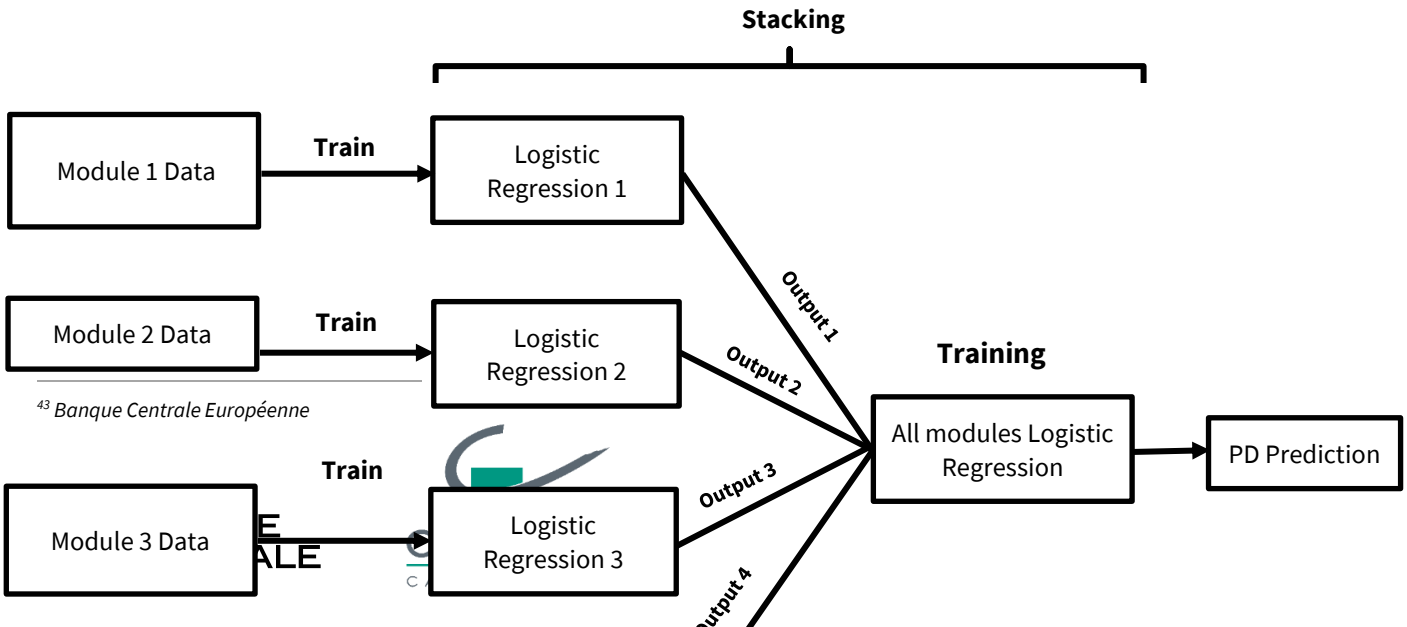
In the context of bank's risk management as defined by the European central bank, RISQ/MRM team operates in the line of defense 2 (LOD2) for risk models' risk control. the team acts as a robust barrier against potential risks that may arise due to model deficiencies.

Introduction to PD Estimation Models

PD estimation models are scoring models for credit's risk internal notation that aims to predict the default probability of a borrower (PD). They are a **stacking** of simple models, usually formed by logistic regressions or decision trees (*Figure 18*).

BCE[43] regulation imposes that models of this kind must be interpretable. This justifies the use of simple models such as logistic regression. However, simple models have usually lower performance in comparison to complex models. The idea we want to demonstrate is by training complex models and distilling them in the same PD estimation models will enhance the latter performance whilst staying interpretable.

We use the dataset of France's SMEs portfolio containing the PD estimation for various obligors across different time points. The PD estimates are derived by analyzing a collection of features that describe each borrower. These features are organized into four distinct modules. For instance, within the first module, we capture essential attributes pertaining to each SME. These attributes encompass metrics like net cash, equity, social and fiscal debt, turnover, and various other key financial indicators.
Modules' features are considered confidential information. Familiarity with these features is not essential for comprehending the upcoming tasks and does not constitute a part of the internship personal work.



[43] *Banque Centrale Européenne*

Model	Training AR44(%)	WT Test AR (%)	OOT Test AR (%)
Logistic regression 1	55,3	53,2	60,2
Logistic regression 2	53,91	52,98	57,43
Logistic regression 3	64,7	64,4	66,76
Logistic regression 4	28,86	28,88	32,78
Stacked LR	65,4	66,2	66,4

Table 14. Performance of PD ESTIMATION MODELS

Figure 36. *PD estimation* models training workflow.

The PD estimation models employ a structured approach, which involves conducting logistic regressions for the estimation of probability of default (PD) within each of the four modules. To elaborate, we utilize the first module's features to train a logistic regression model exclusively dedicated to estimating PD based on module's characteristics. This procedure is replicated for the remaining three modules.

Subsequently, the outputs generated by each of the four logistic regression models are employed as inputs for a final logistic regression model. This final model is designed to estimate PD through a stacking

[44] *Accuracy Ratio* = $2 \times ROC\ AUC - 1$

mechanism, effectively integrating the insights derived from all four modules to enhance the overall PD estimation process. Table 6 illustrate the performance of each logistic regression on each module and also the stacked model integrating insights from each module for final PD estimation.

We employ a specific metric called accuracy ratio, which differs from the standard accuracy measures, to evaluate our models. To assess the performance of the models, we utilize two distinct sets of test data:

- Within-Time (WT) Test Dataset: In this test dataset, each obligor's data comprises records from various time points (panel data).

- Out-of-Time (OOT) Test Dataset: In this test dataset, each obligor is associated with a single data record fixed at a specific time point, which, in our case, is set at 2018.

It is important to note that in the training dataset, each obligor may have records from different time points (panel data). We assume that obligors from different time points are independent of one another to assure the logistic regression independency assumption.

To conclude, the goal is to enhance performance of the overall model (stacked model) on the within-time and the out-of-time test sets.

Lending Club Dataset

The objective of training a model on the *Lending Club* dataset is to detect discernible patterns that can serve as predictive indicators of an individual's likelihood to default on a loan. This information can then be applied to make informed decisions, such as denying a loan application, adjusting the loan amount, or offering loans to high-risk applicants at elevated interest rates, among other potential actions. When an individual applies for a loan, there are two distinct outcomes to consider:

- Fully Paid: In this scenario, the loan applicant has successfully met their obligation by repaying both the principal amount and the accrued interest rate in a timely manner.

- Charged-off: Conversely, in this situation, the loan applicant has failed to make the required installment payments within the stipulated time frame for an extended period, resulting in a default on the loan.

There are two primary types of risks associated with the bank's decision-making process, and our objective is to minimize in the context of model risk management:

- Risk of Missed Opportunities: When an applicant is likely to repay the loan, declining their application leads to a loss of potential business for the company. In this case, our aim is to **maximize precision**, ensuring that loan approvals are granted with a high degree of confidence to minimize the instances of unnecessarily denying creditworthy applicants.

- Risk of Default: Conversely, when an applicant is not likely to repay the loan and is at risk of default, approving their loan could result in a financial loss for the company. To mitigate this risk, we aim to **maximize recall**, identifying and declining applications from individuals who are likely to default, thus minimizing the chances of financial losses due to loan defaults.