LUISS

Corso di laurea in Management and Computer Science

Cattedra Artifical Intelligence and Machine Learning

Redefining Anti-Money Laundering Detection: The Role of Artificial Intelligence in Financial Security

Prof. Giuseppe Italiano
RELATORE

Matr. 267091
CANDIDATO

Anno Accademico 2023/2024

Table of contents

1. Introduction

1.1 Motivation

The ever-increasing complexity and number of financial transactions around the world provides fertile ground for the hiding of illicit funds, and therefore serves to highlight the necessity of developing innovative anti-money laundering (AML) solutions. In light of the fact that sophisticated laundering schemes are always evolving in order to bypass regulatory frameworks, traditional anti-money laundering measures, which are frequently rule-based and restrictive, are becoming increasingly unsuitable. While the digital transformation of financial services has increased accessibility and convenience, it has also opened up new channels for financial crimes that can surpass conventional detection systems.

Because of the urgent need for AML systems that are not just reactive but also proactive, utilising new technology to anticipate and prevent illegal activity, this research was motivated by the need to fulfil this demand. Machine learning (ML) is a promising frontier in this regard, giving the potential to dramatically boost the detection capabilities of anti-money laundering systems. This is because ML has the ability to learn from complicated datasets and recognise subtle patterns that are indicative of money laundering procedures.

1.2 Money Laundering

As derived from a study conducted by McKinsey & Co. in the United States in 2022, there is a global increase in the magnitude of money laundering and other financial crimes, accompanied by the continuous development of increasingly sophisticated methods to elude detection. This has generated a robust reaction from the banking industry, which annually allocates billions of dollars to enhance their safeguards against financial crime (estimated $214 billion was spent by institutions on financial-crime compliance in 2020). In addition, regulatory sanctions associated with compliance are increasing annually due to the imposition of harsher sanctions by regulators. However, traditional rule and scenario-based approaches to fight financial crimes employed by banks have consistently seemed a step behind the bad guys, presenting compliance,

monitoring, and risk organisations with an ongoing challenge in their battle against money laundering.

Now, instead, there is a favourable circumstance for banks to establish a leading position. Recent advancements in machine learning are assisting financial institutions in substantially enhancing their anti-money-laundering initiatives, particularly with regard to the transaction monitoring component. In order to incentivize banks to test and implement innovative approaches for combating financial crimes, US agencies are reducing barriers from existing regulations, guidance, and examination practices, in accordance with the Anti-Money Laundering Act of 2020 and the subsequent National Illicit Finance Strategy.

This momentum in the struggle against financial crimes is generating considerable interest among industry leaders in machine learning. Twelve chiefs of anti-money laundering and financial crime from fourteen of the largest North American institutions were invited by McKinsey to discuss the implementation of ML solutions in transaction monitoring. Over eighty percent of the respondents had initiated the adoption of ML solutions, and the majority anticipated devoting significant resources to ML solution implementation within their anti-money-laundering programmes within the next few years.

But let's start from scratch. What exactly is money laundering?

Well, money laundering is legally defined as "transferring illegally obtained money through legitimate people or accounts so that its original source cannot be traced" (Black's Law Dictionary 2009: 1097). The cumulative amount of global money laundering, as estimated by the International Monetary Fund (IMF), is around $3.2 trillion, which is equivalent to 3% of the global GDP (Jorisch 2009). Money laundering is frequently employed as a means to fund illicit activities such as terrorism, human trafficking, drug trafficking, and the sale of prohibited armaments (Jorisch 2009).

1.3 Phases of Money Laundering

As stated by the United Nations Office on Drugs and Crime (UNODC), the money-laundering cycle can be broken down into three distinct stages. The stages of money-laundering include:

1. Placement (i.e. moving the funds from direct association with the crime)
2. Layering (i.e. disguising the trail to foil pursuit)
3. Integration (i.e. making the money available to the criminal, once again, from what seem to be legitimate sources)

The placement stage is when criminal proceeds enter the financial system. This stage relieves the criminal of substantial amounts of illegally obtained currency and deposits it into the legal financial system. This is when money launderers are most likely to be detected since big deposits into the legitimate financial system may raise concerns.

Layering — also called "structuring" — follows placement. It's the most complicated stage of money laundering and often involves international transfers. The placement stage focuses on separating illicit money from its source. A complex financial transaction stacking technique hides the audit trail and breaks the link to the original illicit activity.

The final step in money laundering is "integration." The offenders receive money from a seemingly reputable source at this point. After being placed as cash and stacked through financial transactions, illegal proceeds are now fully integrated into the financial system and can be used for any legitimate purpose.

Financial institutions, such as banks, employ anti-money laundering systems to detect and prevent money laundering activities. These systems are designed to identify potential money launderers, money laundering risks, and money laundering transactions (Unger and Waarden 2009). Third-party supervision is necessary for financial institutions to justify the operations they conduct, the risks they assume, and the policies they adopt (or abstain from adopting) (Parkman 2012). And when a financial institution

doesn’t respect the required AML standards, they face reputational risk, this is defined by the US Federal Reserve Board (2017) as “the potential that negative publicity regarding an institution’s business practices, whether true or not, will cause a decline in the customer base, costly litigation, or revenue reductions” (pp. 6.5–6.6). This could lead to a significant reputational cost.

While financial institutions are required to adopt strong anti-money laundering procedures, the efficiency of these systems is currently constrained by the technology used, revealing an important area for improvement in the incorporation of more powerful AI technologies.
But currently, gaps exist between financial institutions' AML systems and cutting-edge AI solutions; their current AML systems rely on a combination of human expertise and machine automation. These methods frequently use AI or data-mining tools; However, there is still a substantial reliance on human auditors. In fact, financial institutions' AI systems are often overly simplistic and rule-based, resulting in a huge number of suspicious transactions that need significant time and resources to be assessed. The use of AI improves and simplifies the overall decision-making process while maintaining compliance with rules such as GDPR. It can reduce the amount of transactions that are incorrectly classified as suspicious and increase the productivity of financial institutions.

1.4 Objective of the research

We are going to integrate machine learning techniques into AML procedures, and their performance will be evaluated. This study aims to explore the current implementation of AML processes and to understand where AI can be used to increase their efficacy. Then we will conduct a comprehensive evaluation of various classes of algorithms in order to determine which models are the most effective when it comes to detecting and analysing activity related to money laundering within financial systems.

Through the utilisation of the SynthAML synthetic dataset, which we're going to talk about later, this thesis will investigate a variety of machine learning methodologies in order to determine which models are most effective in capturing the intricate patterns

that are indicative of illegal financial transactions. For the purpose of accomplishing this goal, the research will concentrate on the development, implementation, and improvement of machine learning models that not only properly detect possible instances of money laundering but also improve the overall effectiveness of anti-money laundering monitoring systems.

1.5 Structure of the research

This thesis is structured to methodically explore the integration of machine learning techniques in enhancing anti-money laundering systems, using the SynthAML synthetic dataset to benchmark and evaluate the effectiveness of the proposed models. The organisation of the thesis is as follows:

Chapter 1 begins by outlining the motivation of this research. It proceeds by providing an overview of what money laundering actually is. Then, it considers the challenges posed by modern money laundering techniques and the potential of machine learning to address these problems.

In-depth discussion of present AML procedures, their drawbacks, and the most effective approaches with which machine learning can enhance AML is covered in Chapter 2.

Focused on the SynthAML synthetic dataset, Chapter 3 describes the data characteristics, the process of data preprocessing, and the insights that guide the experimental design. It details the rationale for using synthetic data, emphasising its relevance and utility in circumventing the challenges of data scarcity and privacy concerns in AML research.

Chapter 4 focuses on the development and testing of the machine learning algorithms at the core of this research. Different techniques are proposed and evaluated to find the one that best suits the goal of the research, which is effectively identifying fraudulent transactions among millions.

Chapter 5 discusses the output of the models on the SynthAML dataset. Chapter 6, the last one, reviews the most significant findings from the study, considers the AML implications, and highlights the possible constraints that were faced in the implementation of the models. It comes to end with suggestions for future research and proposing directions for improving financial crime recognition with AI.

2. Theoretical Background

2.1 Anti-Money Laundering (AML)

The term "anti-money laundering" refers to a group of laws, rules, and practices designed to stop criminals from passing off money they have gotten unlawfully as legitimate earnings. The goals of AML frameworks are to deter, detect, and disrupt money laundering procedures.

2.1.1 Current AML Procedures

Effective anti-money laundering procedures are critical for ensuring the stability and integrity of both national and international financial markets. In Italy, for example, AML compliance means not only obeying national legislation established by the “Unità di Informazione Finanziaria” (UIF) of the Bank of Italy, but also complying to international standards established by the Financial Action Task Force (FATF). The FATF is an intergovernmental organisation founded in 1989 during the G7 meeting in Paris with the goal of protecting the financial system and the economy as a whole from threats such as money laundering, terrorism financing, and the proliferation of weapons of mass destruction.

The FATF requires all AML programmes to incorporate specific data analysis and reports. They also require that an institution be able to identify and authenticate its clients, known as the "know your customer" (KYC) requirement. This forbids anonymous accounts and fake account holder names, and requires institutions to take precautions when dealing with correspondent and shell banks. Another requirement is that banks maintain records of all transactions for at least 5 years. The data must contain the names of customers and/or beneficiaries, their addresses, the nature of the transactions, the dates of the transactions, the types of currencies, the quantities of currency, the types of accounts, and the identification numbers of any account utilised.

2.1.2 AML Historical Milestones

AML security measures have changed dramatically since the 1980s, when the subject of money laundering first gained international notice. Key milestones in anti-money laundering laws and procedures include:

- The *Financial Action Task Force* (FATF), established in 1989, and its "Forty Recommendations on Money Laundering" set the international tone for anti-money laundering operations.
- The *EU Money Laundering Directive* was introduced in the early 1990s and, along with successive amendments, strengthened due diligence obligations.
- Following 9/11, the *USA PATRIOT Act* of 2001 considerably increased the scope of anti-money laundering legislation to cover counterterrorism financing.

2.1.3 Future Perspectives

As digital financial services have grown, so have money laundering approaches, demanding continuous modifications to AML strategies. The growth of cryptocurrencies, for example, has posed new obstacles that required the creation of blockchain analysis tools to detect illicit digital currency transfers. New products, in fact, are emerging such as the crypto investigation platform "Coinpath Moneyflow", by the company *Bitquery,* that help crypto investigators and law enforcement agencies in tracking crypto addresses and transactions on all the major blockchains. The European Union is also taking action in this matter: "The new Regulation on the Traceability of Transfers of Funds", that will take effect in December, ensures the traceability of crypto-assets transfers and the authentication of users, aligning with FATF standards.

2.1.4 AML Policies

There are two main types of AML policies:

- *Rule-based* policies;
- *Risk-based* policies.

The 40 recommendations released by FATF, along with "The Nine Special Recommendations on Terrorist Financing", enacted after 9/11 terrorist attacks in New York, are rule-based policies.

Rule-based policies are specific guidelines or procedures set up to detect and prevent activities that may involve money laundering. They are designed to flag certain types of transactions that are typically associated with money laundering based on some predefined criteria. However, the clarity of these regulations makes it more difficult to identify fraud because transactions can be set up to avoid the rules. The guidelines also result in an overreporting of suspicious behaviour, which is costly and time-consuming.

Risk-based policies, on the other hand, overcome the limitations of rule-based solutions by estimating client and transaction risks, and by identifying outlier behaviour. However, this is a higher-risk method that can lead to a worsening in 0ver-reporting or even cause banks to lose customers or be fined for under-reporting.

2.2 General Data Protection Regulation (GDPR)

Recently, the European Union approved several crucial regulations, known as GDPR, governing the collection, storage, and use of personal information; GDPR took force as legislation across the EU in May 2018 and superseded the EU's 1995 Data Protection Directive.

According to Rhahla, Allegue, and Abdellatif (2021), the challenges of implementing GDPR in big data systems are considerable. GDPR lays out precise guidelines for data-driven systems to safeguard personal data and guarantee privacy rights. These rules are meant to promote safe and legal personal data management inside the EU. The GDPR sets out the following requirements for data-driven systems:

1. Lawfulness, Fairness, and Transparency:

- Lawfulness: Data processing must have a legal basis, such as consent, contractual necessity, compliance with legal obligations, protection of vital interests, public interest, or legitimate interests.

- Fairness: The rights of the data subjects must not be harmed by data processing.
- Transparency: Data subjects have to be told how their data is being gathered, utilised, processed, and why.

2. Purpose Limitation:

- Data collection must be done for specific, unambiguous, and legitimate goals; it cannot be processed further in a way that goes against those goals.

3. Data Minimization:

- Adequate, relevant, and limited data collection is required in respect to the processing goals.

4. Accuracy:

- Data needs to be accurate and up to date.

5. Storage Limitation:

- Personal data should be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed.

6. Integrity and Confidentiality:

- Data processing must take place in a way that guarantees adequate security, including protection against illegal or unauthorised processing as well as against unintentional loss, destruction, or damage.

7. Accountability:

- The data controller bears accountability for and needs to be able to prove conformity to the other GDPR standards. This comprises keeping accurate records of data processing operations, evaluating the effects of risky processing operations, and implementing data protection policies and measures.

2.3 AML Industry Frameworks

Today, the ordinary AML workflow in industry is a linear pipeline connecting a data source to a rule-based system. Analysts then do their own research to assess whether transactions are legitimate or fraudulent. According to Han et al. (2020), a specific multistage process is followed. First, AML frameworks gather and process data. Second, they screen and track transactions. If a transaction is discovered to be suspicious, it is flagged, and a human analyst determines if the flagged transaction is fraudulent.

AML frameworks can be divided into four layers. The first layer is the *Data Layer*, which handles the gathering, management, and storage of pertinent data. The second layer, the *Screening and Monitoring Layer*, checks transactions and clients for suspicious activity. Financial organisations have mostly automated this layer, turning it into a multistage system based on rules or risk analysis. If a suspicious behaviour is detected, it is forwarded to the *Alert and Event Layer* for further investigation. This method entails augmenting the data with past transaction information and supporting evidence in order to analyse the flagged transaction. A human analyst in the *Operational Layer* makes the choice to either block or authorise a transaction.

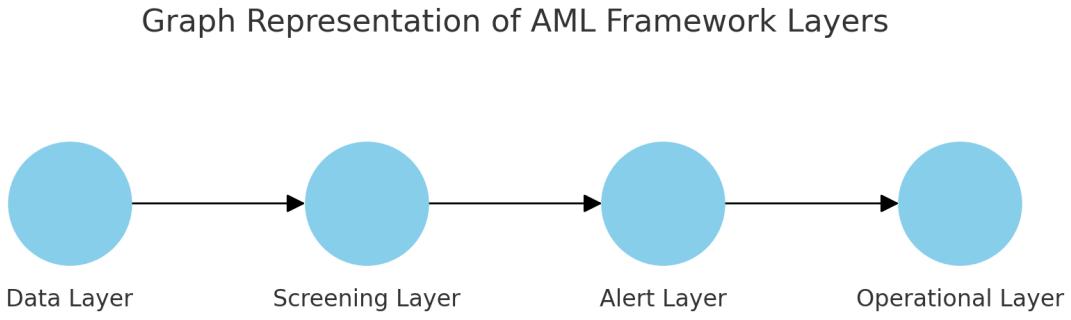

Figure 1: Graphical Representation of AML Framework Layers

2.3.1 Data Layer

The Data Layer is the initial layer in the AML system, responsible for data collection, management, and storage. This layer processes both *internal* and *external* data. Some data sources are accessed and acquired directly, such as customer accounts and real-time transactions. These are used to assess client profiles, measure transaction risk, and

behaviour diagnosis. These data are utilised to validate the final conclusions made while evaluating transactions across the system. External data are data gathered from sources other than the financial institution; these can include regulatory agencies, government authorities, sanctions, and watch lists. Social media and news websites are commonly regarded as external data sources; yet, they are currently underutilised in AML solutions.

Traditional systems have architectural limitations in terms of data quality, data management, and data governance. Big data technology and distributed data processing are not yet commonly used in the AML community. Different databases (relational and linear) are used to store raw and processed data, as well as analytical results. We categorise this layer into the following components based on their purpose:

Collection agent: It is responsible for collecting data both internally and externally. Generally, this component employs a distributed data collection and processing system.

Processing agent: Data processing is an important but time-consuming operation. Because the data comes from diverse sources, it is collected and generated using several sources. All incoming data must be standardised for later usage. The data is frequently enhanced using meta-data generation and linking techniques, and it is formatted and compressed using information retrieval tools and NLP techniques, such as tokenizers.

Insight agent: This component extracts insights from data using various data analysis approaches. For example, similar transactions and consumers are grouped together based on their profile. Anomalous transactions can also be detected using consumer or organisation profiles. This information is utilised in the succeeding layer to determine whether a transaction is suspicious.

Storage agent: Data storage is an important component of this layer. Data collected, created, and processed in this tier must be kept in relational or linear databases. This layer is frequently used to integrate many data management frameworks.

Security agent: Data security is one of the top priorities for financial institutions. Sensitive financial data, such as credit card and account information, must be protected in the bank's database and pipeline. Furthermore, organisations must meet the criteria of global data protection policies. The Security Agent meets these standards; its primary goals are to prevent data breaches, by safeguarding sensitive financial data, and to update security measures to new kinds of cyber-attack. This agent makes use of technologies like secure key management, access controls, data access monitoring, firewalls, and advanced encryption techniques.

2.3.2 Screening and Monitoring Layer

Screening and monitoring make up the second layer of the AML system. Screening occurs prior to the execution of a transaction and includes both name and transaction screening. The monitoring procedure runs on an ongoing basis, surveying transactions and client profiles using analytical models. The components in this layer work together in a collaborative framework that includes numerous tools. They maintain bidirectional communication to the Data Layer for data retrieval and post-operational storage. To better comprehend these components, we categorise them based on their respective tasks:

Transaction-screening module: This module runs before a transaction is conducted and is used to ensure compliance with sanctions. In general, the Wolfsberg Statement of AML Screening Monitoring and Searching is used worldwide for real-time transaction screening. The transaction-screening module maintains links to the Data Layer in order to receive and filter external data.

Name-screening module: This module is designed to identify payments made to individuals or organisations that regulatory authorities have recognised as potential money launderers. The tests are conducted continually and in real time, thus the module must maintain connections to the Data Layer.

Transaction monitoring module: This module detects suspicious transaction patterns and generates a suspicious activity report or Suspicious Transaction Report (STR). This module implements several data mining, AI, and visualisation techniques, such as link analysis and outlier detection. Along with its own analysis engine, this module incorporates the results of the screening modules and communicates with the Data Layer to acquire information and save reports.

Client profile-monitoring module: This module examines a client's account to provide an overview of their profile. It also maintains a bidirectional connection with the Data Layer, allowing it to accept client data and store analytical results. The component works with other modules in the Screening and Monitoring Layer, but it concentrates on specialised tasks, such as warning high-risk countries, analysing financial connections and business partnerships, and determining political affiliations. Transactions deemed suspicious are flagged for further processing by the alert and operational layers.

2.3.3 Alert and Event Layer

If a suspected transaction requires human inspection, the Alert and Event Layer sends an alert. The huge number of transactions to be analysed, along with the poor supporting data provided for the investigation, increases the time required to inspect each transaction. Financial institutions are incorporating cutting-edge statistical and data-related technologies into their anti-money laundering strategies in order to reduce the risks and costs associated with manual inspection. This layer makes decisions based on historical data from past decisions as well as comparisons to similar transactions and decisions. If a transaction is flagged, the layer supplements it with additional information for the evaluator. This comprises a history of similar transaction decisions, risk scores derived by preceding layers, and the transaction's priority for clearance. This layer generates and prioritises alerts, which are then routed to the Operational Layer for operator intervention to permit, block, or reject the transaction. This layer has a high number of false positives — legitimate transactions that are wrongly flagged as suspicious. As a result, the human evaluator is left with a big stack of tasks, which tends to be overwhelming.

2.3.4 Operational Layer

Human agents make the final choice to block, release, or queue a transaction depending on the data from preceding layers. A human agent must make the final decision on a transaction, as required by law. Human agents utilise a variety of approaches to collect and visualise additional information about the suspected transaction. Rule-based industry solutions dominate the existing methodologies used by financial organisations. In these systems, established rules (such as transaction thresholds) are applied to incoming transactions to identify whether or not they are suspicious. Despite their simplicity and ease of manipulation, rules are often used because they make it simple to demonstrate regulatory compliance.

3. Machine Learning for AML

Although traditional AML frameworks have provided a solid foundation for recognising and managing money laundering risks, they frequently struggle to keep up with the ever-changing nature of financial fraud. As financial institutions continue to face issues with the efficiency and adaptability of their frameworks, there is an increasing trend towards more advanced technology solutions. Machine learning is a critical innovation in this area and has the potential to dramatically improve the predictive capacity and efficiency of AML systems.

3.1 Fundamentals of Machine Learning in AML

The term machine learning (ML) was first coined in a 1959 paper by Arthur Samuel, who was an IBM researcher and pioneer in the field of AI. He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed".

ML is a method of approximating a function that maps input space to output space by extracting compressed non-redundant information from data samples. (Swarnendu Ghosh et al., 2022). It differs from rule-based programming in that rule-based systems are based on hand-crafted rules that are composed to carry out intelligent tasks, while machine learning techniques attempt to formulate a function that maps input space to output space.

Figure 2: An illustration of the pipeline of building a machine learning system

Figure 2 illustrates a three-step process for creating an effective machine learning system. In the first stage, we must collect an adequate amount of training data to reflect previous experience from which computers can learn. In the second stage, we typically

need to use domain-specific processes to extract the so-called features from the raw data. In the last stage, we select a learning algorithm to construct some mathematical models using the extracted feature representations of the training data.

There are several types of machine learning, each with special characteristics and applications. Some of the main types of machine learning algorithms are Supervised ML, Unsupervised ML and Reinforcement Learning.

3.1.1 Supervised Machine Learning

Supervised learning is defined as when a model gets trained on a “labelled dataset”. Labelled datasets have both input and output parameters. In Supervised Learning algorithms learn to map points between inputs and correct outputs. There are two main categories of supervised learning: *Classification* and *Regression*.

Classification deals with predicting categorical target variables, which represent discrete classes or labels. Some classification algorithms are: Logistic Regression, Support Vector Machine, Random Forest, Decision Tree, K-Nearest Neighbors (KNN).
Regression, on the other hand, deals with predicting continuous target variables, which represent numerical values. Here are some regression algorithms: Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, Decision tree, Random Forest.

Supervised ML comes with many advantages such as the interpretability of the decision-making process or the high accuracy potential with high quality data. However, it also comes with some limitations: it has problems in recognizing patterns and may struggle with unseen or unexpected patterns that are not present in the training data.

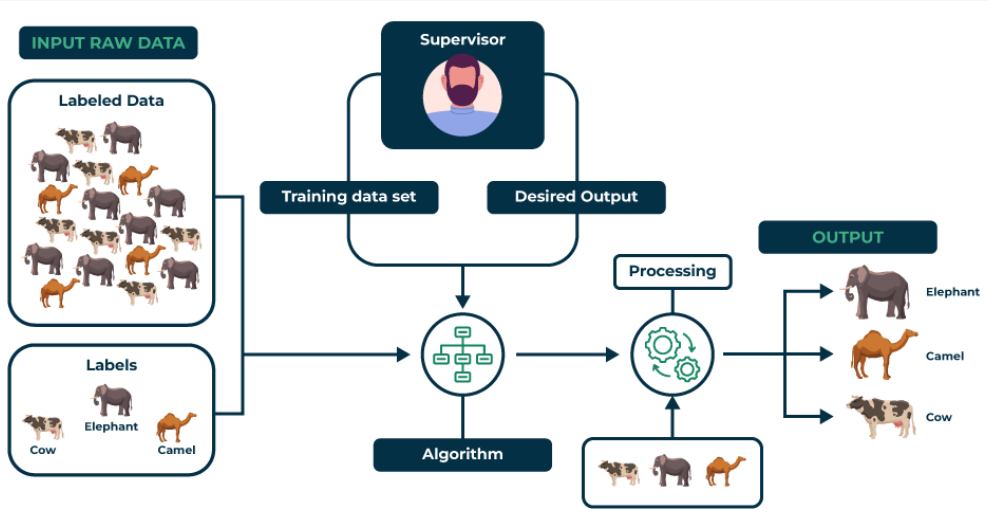

Figure 3: Visual Representation of Supervised ML Process

3.1.2 Unsupervised Machine Learning

Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data. Unlike supervised learning, unsupervised learning doesn't involve providing the algorithm with labelled target outputs. The primary goal of Unsupervised ML is often to discover hidden patterns within the data.

The main category of algorithms belonging to Unsupervised ML are Clustering algorithms. is the process of forming groups of homogeneous data points from a heterogeneous dataset. It evaluates the similarity based on a metric like Euclidean distance, Cosine similarity, Manhattan distance, etc. and then groups the points with highest similarity score together. Here are the most famous clustering algorithms: K-Means Clustering algorithm, DBSCAN Algorithm, Principal Component Analysis, Independent Component Analysis.

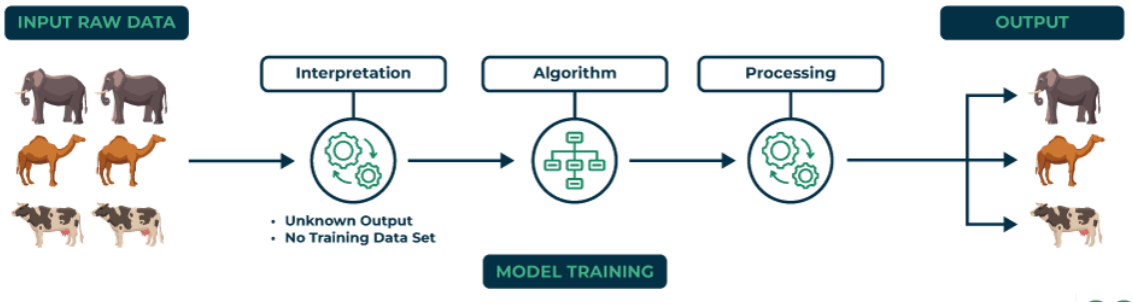

Figure 4: Visual Representation of Unsupervised ML Process

3.1.3 Reinforcement Learning

Reinforcement machine learning algorithm is a learning method that interacts with the environment by producing actions and discovering errors. Trial, error, and delay are the most relevant characteristics of reinforcement learning. In this technique, the model keeps on increasing its performance using Reward Feedback to learn the behaviour or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with humans and even itself to get better and better. Each time we feed in data, they learn and add the data to their knowledge which is training data. So, the more it learns the better it gets.

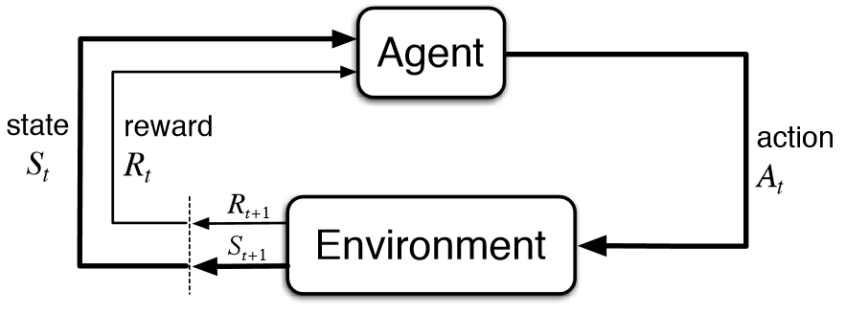

Figure 5: Simple Representation of Reinforcement Learning Process

3.2 General Principles in ML

3.2.1 Occam’s Razor

Occam's razor is a problem-solving principle rooted in philosophy and science. It is usually stated as "the simplest solution is most likely the right one." In the context of machine learning, Occam's razor refers to a preference for simplicity in model selection. If two distinct models offer similar results on training data, we should pick the simpler model over the more complex one.

Furthermore, there’s a formalisation of Occam's razor in machine learning, that is the principle of minimum description length (MDL), which states that all machine learning methods seek regularities in data, and the best model (or hypothesis) for describing those regularities is also the one that can compress the data the most (Grünwald 2007).

3.2.2 No-Free-Lunch Theorem

The no-free-lunch theorem in machine learning states that no single strategy is superior to others for all learning problems. Given any two machine learning algorithms, if we utilise them to solve all conceivable learning problems, their average performance must be equal. Even worse, their average performance is equivalent to random guessing. This is true regardless of the learning strategy we apply to estimate the model. The average prediction performance of an estimated model over all potential scenarios for the ground-truth function is similar to a random guess, as each good-prediction instance can also result in a number of bad-prediction situations.

The no-free-lunch theorem states that no machine learning system can learn anything useful only from training data. If a machine learning approach performs well for some issues, it must have explicitly or indirectly employed knowledge of the underlying problems other than the training data. That's why, since there is no "best" algorithm for every task, selecting the appropriate approach for a given problem is critical. The performance of an algorithm will be determined by how well it meets the unique qualities and requirements of the task at hand, Anti-Money Laundering Detection in our case.

3.2.3 Law of the Smooth World

Despite the aforementioned no-free-lunch theorem, one of the primary reasons why many machine learning algorithms succeed in practice is that our physical world is always “smooth” (Jiang, 2021, p. 12). Because of the harsh restrictions that exist in reality, such as energy and power, any physical process in the macro world is inherently smooth, meaning that the fluctuations are often continuous, gradual and predictable rather than abrupt or erratic. As a result, when we apply machine learning to real-world problems, the law of the smooth world is always applicable, substantially simplifying many of our learning challenges.

3.2.4 Curse of Dimensionality

The curse of dimensionality describes a number of phenomena that emerge in high-dimensional environments when data is analysed. Richard E. Bellman first used the phrase when referring to issues with dynamic programming. Generally speaking, the curse describes problems that occur when the number of datapoints is small in relation to the intrinsic dimension of the data. Usually, a large volume of training data is needed to guarantee that there are several samples with each combination of values in machine learning problems that involve learning a "state-of-nature" from a finite number of data samples in a high-dimensional feature space with each feature having a range of possible values. Abstractly, we require exponentially more data to make correct generalisations the more characteristics or dimensions there are. For every dimension in the representation, it is usual practice to have at least five training instances. We won’t incur this issue in our research, since SynthAML, the data set we are going to use, has only a few dimensions, with millions of rows.

3.3 Machine Learning Algorithms in AML

The techniques most akin to the nature of anti-money laundering are Outlier Detection and Risk Scoring. They stand out due to their direct alignment with the objectives of identifying and prioritising suspicious activities. We’re going to mention and explore the best-known algorithms for each use case. These are also the algorithms that we will try and implement later.

3.3.1 Outlier Detection

K-means clustering

K-Means Clustering is a technique for unsupervised machine learning that divides the unlabeled dataset into various clusters. By using K-Means clustering, data points are assigned to one of the *K* clusters based on how far they are from the cluster centres. First, the cluster centroid in the space is assigned at random. Next, based on how far each data point is from the cluster centroid, it is assigned to a cluster. New cluster centroids are assigned once each point has been assigned to a cluster member. Iteratively, this approach searches for a good cluster. For the purpose of detecting outliers, K-Means clustering works best when the data is well-separated. This approach is faster than other clustering techniques but diverse clusters can result by modifying the initial cluster centroid assignment.

K-Means clustering's purpose is to divide the set of data points into a number of groups so that the data points within each group are more comparable to one another while remaining distinct from the data points within the other groups. It is essentially a classification of objects based on how similar and different they are to one another.

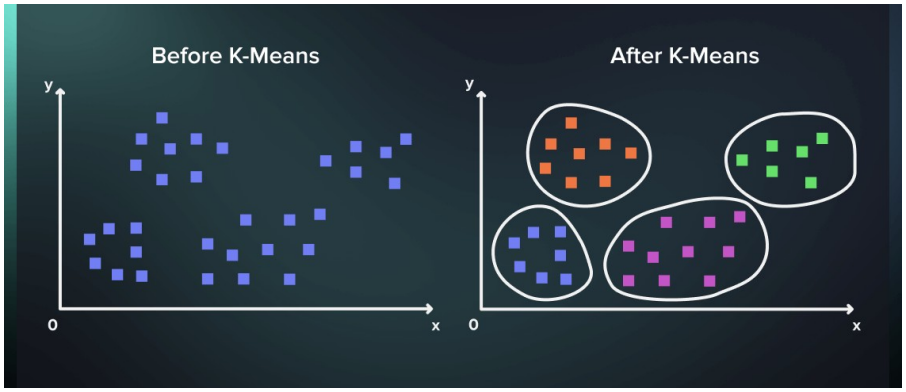

Figure 6: Visual Representation of K-Means Clustering

Isolation forest

Isolation Forest is a state-of-the-art anomaly detection technique known for its efficiency and simplicity. Using binary partitioning to remove anomalies from a dataset, it rapidly discovers outliers with low computing expense. Isolation forest isolates irregularities within a dataset using recursive partitioning.

- Isolation Forest methodology consists of randomly selecting features and separating them along random values until separate data points are identified.
- This "isolating" procedure is in charge of constructing partitions or "trees" that try to distinguish anomalies from normal observations.
- Outliers usually require fewer splits to isolate, since they are fewer in number and further from the norm, and that makes them easier to discover.

This algorithm uses the idea of isolation to effectively tell the difference between normal and abnormal behaviour. This lets you take instant action to remove potential threats or find valuable insights hidden in data anomalies.

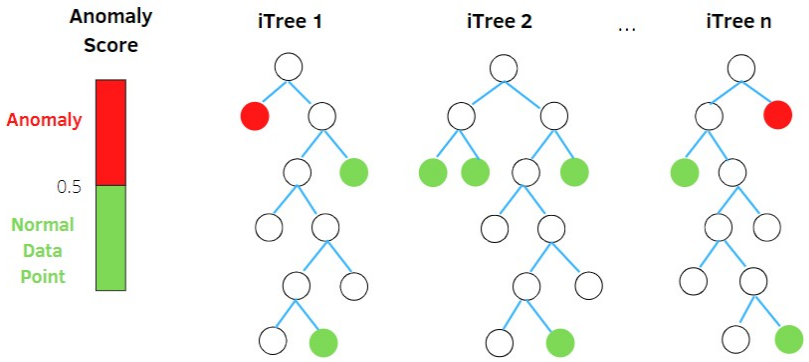

Figure 7: Visual Representation of Isolation Forest

K-Nearest neighbours

The k-nearest neighbours (KNN) algorithm is a non-parametric, supervised learning classifier that uses proximity to classify or predict the grouping of a certain data point. The KNN algorithm is versatile and may be applied to both regression and classification problems. However, it is commonly employed as a classification algorithm, based on the underlying assumption that points that are close to each other are likely to be similar. In classification problems, a class label is assigned based on the majority vote, meaning that the label that appears most frequently around a given data point is utilised. It is important to mention that the KNN method belongs to a group of "lazy learning" models, which means that it only retains a training dataset without going through a training stage. Furthermore, this implies that all the computational processes take place

during the classification or prediction phase. Due to its heavy reliance on memory for storing training data, this learning method is commonly known as instance-based or memory-based.

3.3.2 Risk Scoring

Support Vector Machine

In supervised machine learning, a support vector machine (SVM) classifies data by identifying the best line or hyperplane in an N-dimensional space that maximises the distance between each class. SVMs find extensive application in classification problems. Through the search for the best hyperplane that maximises the margin between the nearest data points of opposing classes, they differentiate between two classes. The hyperplane is a line in a 2-D space or a plane in an n-dimensional space depending on how many features are in the input data. The algorithm can determine the optimal decision boundary between classes by maximising the margin between points since several hyperplanes can be discovered to differentiate classes. Its ability to produce precise classification predictions and generalise well to fresh data follows from this. Because it can handle both linear and nonlinear classification problems, the SVM technique finds extensive application in machine learning. Kernel functions are applied to shift the data into higher-dimensional space in order to permit linear separation when the data is not linearly separable. The "kernel trick" refers to this usage of kernel functions; the particular use case and data properties determine the kernel function—linear, polynomial, radial basis function (RBF), or sigmoid—to utilise.

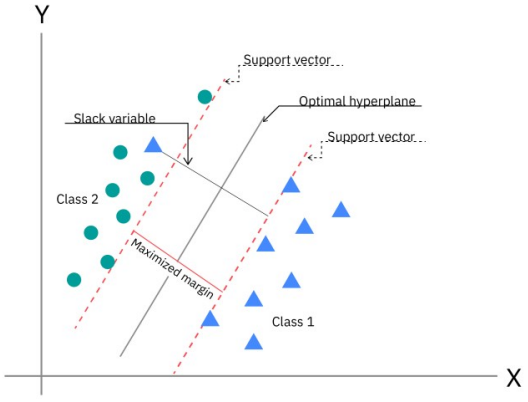

Figure 8: Visual Representation of SVM

Tree-based algorithms (CART, Random Forests, XGBoost)

Classification and Regression Tree (CART) is a popular modelling tool for making predictions about a dependent variable using multiple explanatory variables.

Whereas a regression tree is used when the response variable is continuous, a classification tree is used when the response variable is categorical.

Because CART analysis is simple to interpret and effective with a broad range of data sets, it is widely used. Furthermore, it does not need any model assumptions (that is, no distributional patterns of the variables are necessary).

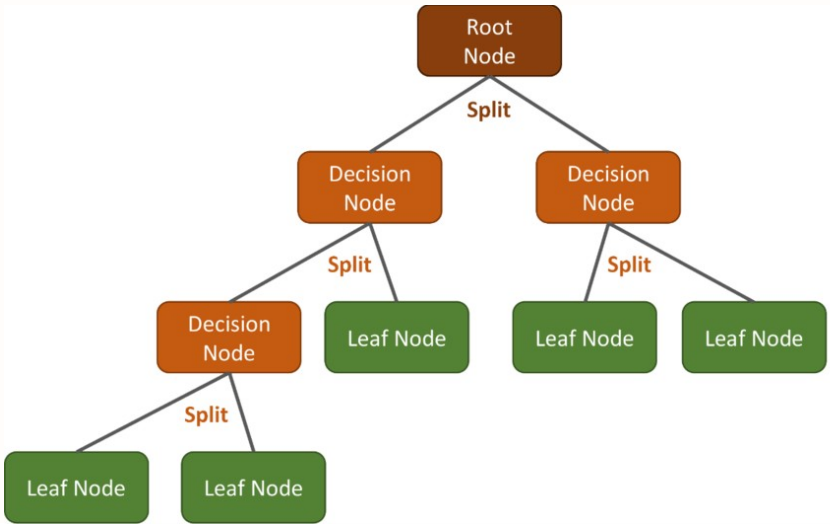

Figure 9: Decision Tree Structure

Random forest, like every tree-based algorithm, is a supervised learning algorithm. The ensemble of decision trees that it constructs is referred to as the "forest," and it is often trained using the bagging approach. The bagging approach is based on the principle that the final outcome can be improved by utilising a number of different learning models simultaneously. Through the process of bagging (bootstrap aggregating) each decision tree is trained on a random subset of instances included in the training set. In other words, every single decision tree that is generated by the random forest is trained using a distinct subset of instances.

XGBoost, or Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning technique. It supports parallel tree boosting and is among the most used machine learning algorithms for regression and classification tasks. A GBDT is a decision tree ensemble learning algorithm for classification and regression that works similarly to random forest. Ensemble learning methods use numerous machine learning algorithms to create a better model. Both random forest and GBDT create a model made up of several decision trees. The distinction is in the way the trees are constructed and combined. The term "gradient boosting" refers to the process of "boosting" or enhancing a single weak model by merging it with a number of additional weak models to get a collectively stronger model. Gradient boosting is a form of boosting in which the process of additively producing weak models is formalised as a gradient descent method over an objective function.

3.4 Shortcomings of current AI models

Policy makers and AML system designers constantly struggle with the definition of money laundering itself. Money laundering and real transactions are sometimes confused because there is no clear pattern that defines fraud. Rule-based procedures and laws find it difficult to keep up with the ever-changing fraud practices. Institutions are forced by these challenges to choose between the *effectiveness* and *efficiency* of their AML system. A system that is effective in identifying frauds fast depends less on human analysts and runs the danger of missing fraudulent transactions. Although most transactions will be thoroughly checked, a lower-risk system is more secure; but, this is expensive for the financial institution as well as for the analysts. Institutions must balance risk and expense while designing their systems. When creating their systems, individual institutions need to evaluate the *trade-off* between *risk* and *cost*.

Artificial intelligence is among the prominent approaches to screening and adaptation problems. However, technical issues and data availability have restricted research into automatic AML. System training should ideally be done with actual data rather than simulated data. The need to protect client privacy translates to the fact that there is currently no open-source data available for money laundering studies. Private institutions must thus supply the data; this is a difficult undertaking because disclosing

client data might damage an institution's reputation and violate data privacy laws. The distribution, storage, and processing of data is among the most important challenges financial institutions must face when putting AML into practice. Most transactions in most institutions are legal and ought to be accommodated. Analysts flag any transactions as fraudulent and forward them to the authorities for additional review. The ultimate decision of whether a transaction is indeed fraudulent isn't always communicated to the financial institution. Financial institutions thus get a lot of suspicious transactions, a portion of suspicious transactions that their analysts flag as fraudulent, but they never hear from the authorities about how accurate their conclusions were. It is consequently difficult to build models for predicting fraudulent transactions because the data that financial institutions can offer researchers is noisy and frequently lacks a classification of "fraudulent" or "legitimate". Lack of shared resources is another major issue; institutions worldwide do not maintain a shared data pool that can be used for the benefit of AML research, apart from watch-lists and specific regulatory guidelines. The difficulty of applying the data privacy and ownership rules of GDPR in such circumstances could impede the advancement of AML research. Moreover, if the justification of decision-making procedures is derived from a private data source, great caution should be used. Important information might always be revealed. Organisations processing personal data are further required by some GDPR regulations to perform a "Data Protection Impact Assessment" in order to identify and reduce data protection concerns. The capacity of contemporary systems to justify a certain action made or anticipated limits their effectiveness in several ways. The explainability of algorithms has received attention recently (Gunning 2017; Ehsan et al. 2018); nonetheless, no uniform or common framework of explanation has been applied, and the kind of explanation differs depending on differences in the data and algorithms. Explanatory nature and model complexity are always trade-offs. Furthermore required is a framework of explanation for the facts. Transparent and obvious must be the legal and functional data descriptions. Particularly supervised machine learning, is entirely data reliant, and a model can capture biases or contradictions found in the data. An important part should be played by a data-explanatory framework, to handle bias, compatibility, and security.

4. Data

We saw how research in AI applied to AML is restrained by limits dictated by the law in data processing. Hence, it is a major issue for financial institutions that must respect strict guidelines from GDPR and other regulations in this matter. The development of advanced and accurate models heavily depends upon transaction data; however, there is currently no publicly available dataset of this nature due to data privacy and ownership policies. SynthAML addresses this gap by providing a synthetic dataset that closely mimics real transaction data while ensuring privacy and security.

SynthAML

SynthAML was developed using advanced statistical and AI techniques to generate synthetic data that reflects the patterns found in actual AML alerts and transactions. This dataset includes 20,000 AML alerts and over 16 million transactions derived from real data provided by Spar Nord, a major Danish bank. The data generation process ensures that the synthetic data retains the statistical properties of real-world data, making it a valuable resource for benchmarking and testing AML methods. Experimental results indicate that performance on SynthAML can be applied to real-world scenarios, facilitating AML research and system evaluation.

4.1 Data description

We will now proceed with the empirical phase of the research. This involves analysing the SynthAML data set to extract the necessary details required for the modelling phase. SynthAML is composed of two tables: one that stores *transaction* details and the other one for the *alerts*.

The “Transactions” table holds transaction histories. We have a one-to-many relation where each alert is associated with a sequence of transactions (identifiable through the alert ID number). Each transaction has four features:

1. a transaction timestamp,
2. the transaction entry (credit vs. debit),

3. the transaction type (card, cash, international, or wire), and
4. the transaction size (measured in log Danske Kroner (DKK) and standardised to have zero mean and unit variance).

In the SynthAML dataset, transactions linked by an alert ID collectively signal potential fraudulent activity, not individually. Each alert ID groups transactions that, together, are considered suspicious. This structure implies that our predictive model's task is to learn from groups of transactions to recognize patterns indicative of fraud. Instead of identifying single fraudulent transactions, the model focuses on detecting suspicious transaction patterns within these grouped sequences, enabling the identification of fraudulent activities based on collective behaviours rather than isolated incidents.

The "Alerts" table holds information about individual AML alerts, including:

1. an alert ID,
2. the date the alert was raised,
3. the outcome of the alert (i.e., if the alert was reported to the authorities or dismissed).

4.2 Exploratory Data Analysis (EDA)

Now let's delve deeper into the nature of the data. We start by conducting a comprehensive analysis of the data sets to understand their characteristics and structure. Exploratory Data Analysis is crucial to understand the dataset's characteristics and structure. The EDA involves summarising the main features of the dataset, identifying patterns, and detecting anomalies. This part is fundamental for the subsequent stages of modelling and evaluation.

The complete code, including analyses, modelling and evaluations, can be found in the Google Colab notebook available at Prova Finale AML - Vitale.ipynb.

We begin by loading the necessary libraries for the EDA (i.e. pandas, matplotlib and seaborn) known to be the go-to libraries for data analysis on Python. We thus proceed by uploading the two data sets in our Integrated Development Environment (IDE).

AlertID	Timestamp	Entry	Type	Size
1	2019-01-01 00:00:41	Credit	Wire	-0.438941
1	2019-01-01 00:00:41	Debit	Wire	-1.656562
1	2019-01-01 00:00:44	Credit	Card	-0.749267
1	2019-01-01 00:00:45	Debit	Wire	0.006226

Table 1: Sample entries from the "Transactions" dataset.

In the table above, an excerpt of the initial rows from the 'Transactions' dataset is presented. In the 'Size' column we have negative values since the variable has been standardised to have zero mean and unit variance, therefore negative numbers simply represent transactions whose size was below the sample mean.

The "AlertID" is a reference to the alert raised by a given group of transactions. The "Timestamp" refers to the time when the transaction was performed and the "Entry" and "Type" columns provide specific details regarding the method and category of the transaction.

The following section presents a series of plots that provide insights into the distribution and trends of transaction data within the dataset.

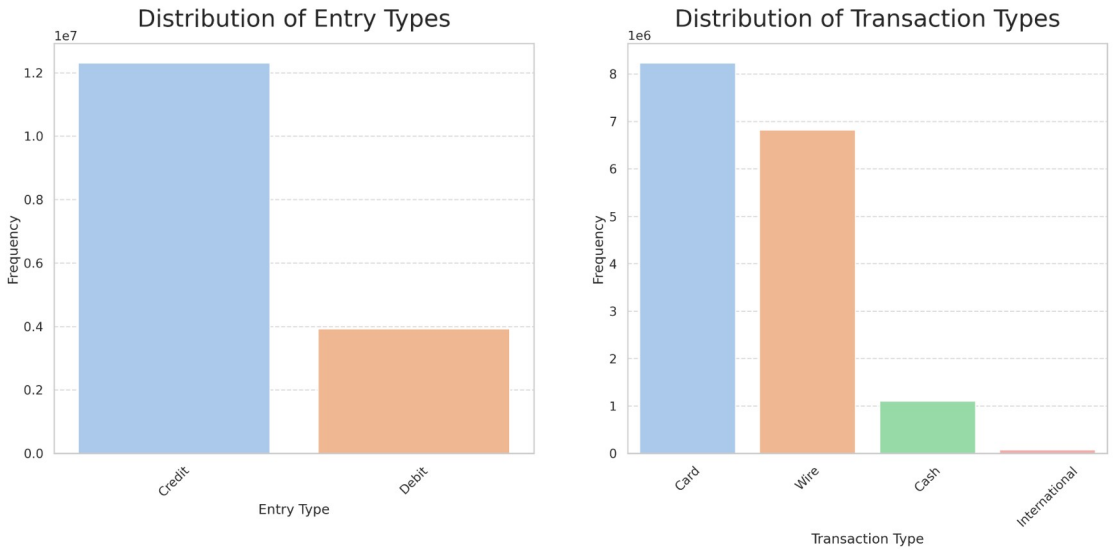

Figure 10: Distributions of transaction entries and types within "Transactions" dataset.

The bar plots illustrate the distribution of transaction entries and types within the dataset. Credit transactions are significantly more frequent than debit transactions, indicating a higher volume of incoming transactions. Among transaction types, card and wire transactions dominate, while cash and international transactions are less common. This suggests a transactional landscape where electronic payments via cards and wire transfers are prevalent, with less reliance on cash and international transactions.

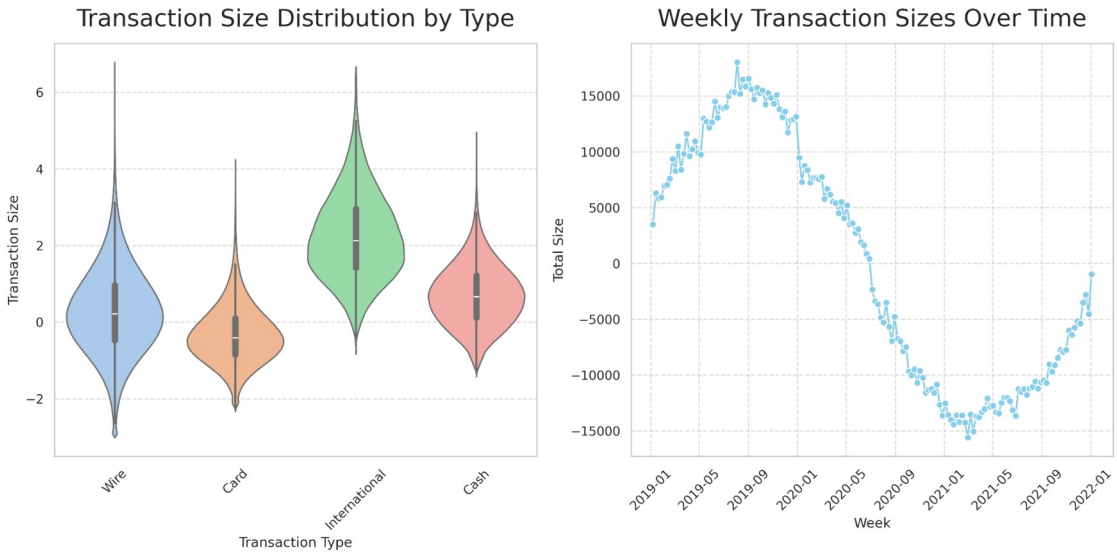

Figure 11:
Distribution of transaction sizes by type and weekly distribution in "Transactions" dataset.

The violin plot shows the distribution of transaction sizes across all the four types: Wire, Card, International, and Cash. Wire and International transactions exhibit higher variability, with a broader range of sizes, while Card and Cash transactions are more consistent, clustering around the median. The plot highlights distinct patterns, such as the (slightly) bimodal distribution in International transactions, suggesting varied transaction behaviours within each type. It appears that International transactions tend to be higher in size with respect to other types.

The line plot, instead, shows the sequence of transaction sizes, aggregated weekly, over a three-year time span. It highlights significant fluctuations, with a peak in mid-2019

followed by a decline in 2020, probably influenced by economic factors, such as the COVID-19 pandemic. From mid-2020 a slow recovery is observed starting, with an upward trend towards the beginning of 2022, indicating a recovery in transaction activity.

Next, let's analyse the "Alert" dataset to uncover patterns and insights that may be relevant to our analysis.

AlertID	**Date**	**Outcome**
1	2020-01-01	Report
2	2020-01-01	Report
3	2020-01-01	Dismiss
4	2020-01-01	Dismiss

Table 2: Sample entries from the "Alert" dataset.

Here we have only three columns, "AlertID", that can be considered the primary key to which the homonymous column in the "Transactions" table refers to. Essentially, when a set of transactions is considered to be fraudulent an alert is raised, all transactions in the same group are assigned the same ID. The "Date" column corresponds to the day when the alert was raised and "Outcome" column is a binary categorical variable and indicates whether the alert was reported to authorities or dismissed. This happens in the *Operational Layer*, discussed in Chapter 1, that is the final layer of an AML system, where a human agent decides whether to report the transactions to authorities or not.

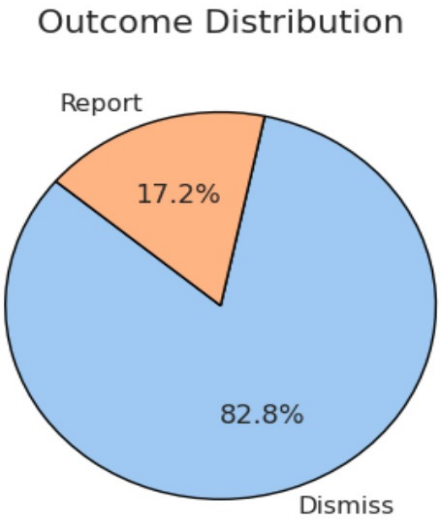

Figure 12: Outcome distribution in "Alert" data set

The pie chart shows that only 17% of the alerted transactions is actually reported. The total number of alerts is 20.000, this means that the reported transaction is about 3.400. This shows a clear class imbalance, it is a problem because a machine learning model will learn that by predicting the class "Dismiss", it will be right most of the time. Thus, it won't be of any utility in spotting fraudulent transactions. That's why class imbalance is a major issue in machine learning and will be addressed later, in the next section.

4.3 Data processing

After we have carefully explored and understood the nature of the dataset, we can now proceed to the data preparation phase, essential for the modelling process. In this phase we will go through several steps in order to make our data set effective for the training of different machine learning models. The first step we'll go through is the feature engineering.

Feature engineering

Feature engineering involves the transformation of unprocessed data into meaningful and useful information that can be utilised by machine learning algorithms. Feature engineering refers to the process of generating predictive model features. A feature, often known as a dimension, is an input variable utilised to generate predictions in a model. Model success heavily relies on the quality of data used for training. Therefore,

feature engineering is a critical preprocessing strategy that involves identifying the most relevant aspects of the raw training data for the predictive task.

With data in the current form, there's not much we can do for predictive modelling. We need a single data set, where each row has numerical features, these features will be used as independent variables. Each row must have one feature that is dependent on the others, this will be the one to predict. In our problem we will structure our data so that we have a row corresponding to each alert, with transaction data as independent variables and the "Outcome" column as the binary variable to predict. Thus, this is a binary classification problem. However, to get where we need, we must perform some intermediate steps.

To begin with, we have two separate data sets, one for Transactions, composed of millions of rows, and one for the Alerts, composed of 20.000 rows. However, each alert comprehends multiple transactions, and each transaction must be part of an alert, since it has an AlertID, referencing an alert in the other table. Hence, we have to aggregate all the transactions belonging to the same alert in a single row. How can it be done without losing too much information?

The Python library Pandas will be instrumental here, due to its great capabilities of manipulating data sets. We will combine many functions from the aforementioned library, to group transactions based on their AlertID and calculate several summary statistics, such as:

- *Total Transaction Count*: Counts how many transactions are associated with each alert.
- *Total Transaction Size*: Sums up the sizes of all transactions for each alert.
- *Average Transaction Size*: Calculates the average size of transactions for each alert.
- *Variance of Transaction Size*: Measures how much the transaction sizes vary for each alert.
- *Maximum Transaction Size*: Finds the largest transaction size for each alert.
- *Minimum Transaction Size*: Finds the smallest transaction size for each alert.

- *Count of Wire Transactions*: Counts how many wire transactions are associated with each alert.
- *Count of Card Transactions*: Counts how many card transactions are associated with each alert.
- *Count of International Transactions*: Counts how many international transactions are associated with each alert.
- *Count of Cash Transactions*: Counts how many cash transactions are associated with each alert.

After calculating these statistics, we will create a new table, where each row corresponds to an alert and contains the summarised information about the whole group of transactions that triggered the alert. All of this can be easily done with a single line of code, as shown below:

```
aggregated_features = df_transactions.groupby('AlertID').agg(
    total_transaction_count=('Size', 'count'),
    total_transaction_size=('Size', 'sum'),
    avg_transaction_size=('Size', 'mean'),
    var_transaction_size=('Size', 'var'),
    max_transaction_size=('Size', 'max'),
    min_transaction_size=('Size', 'min'),
    count_wire=('Type', lambda x: (x == 'Wire').sum()),
    count_card=('Type', lambda x: (x == 'Card').sum()),
    count_international=('Type', lambda x: (x ==
'International').sum()),
    count_cash=('Type', lambda x: (x == 'Cash').sum())
).reset_index()
```

Figure 13: Code snippet for summary statistics with Pandas

This is actually a single line of code that exploits pandas functions and python lambda functions to calculate all the statistics at once.

In our project, we inspected the data set that was just created to be sure that no irregularity was present and we noticed that about 200 null values appeared in the column corresponding to the Variance of Transaction Size, but with a deeper analysis

we discovered that it was occurring in the alerts raised by a single transaction. This is normal since the variance to be calculated needs at least two values. Hence, we substitute the null values with 0s.

Now, we have a new data set, containing in each row most of the information of all the transactions that raised the respective alert. However, it doesn't contain the dependent variable, that is the "Outcome" column, present in the "Alert" data set. We can now merge the newly obtained dataset, to the "Alert" data set, so as to include the feature we need. It can be done again with a simple Pandas function that joins two tables horizontally based on a common column, that in this case is "AlertID". The "Outcome" variable is a categorical binary variable, it means that it only assumes two values (and this is good), but these values are words (and this is not good).

One-Hot Encoding

Machine learning models (or at least most of them) only take as input numerical variables. To transform categorical variables into numerical variables, there's a quite straightforward method. Let's make a practical example with our data. The feature we have to encode is "Outcome", it takes as values "Report" or "Dismiss". To perform one-hot encoding we can simply change the value "Report" into the numerical value "1", and the other one into the value "0". In Python, this can be done in a single line of code as well, using the function "LabelEncoder()" from the sklearn library, that's one of the most used libraries in the field of machine learning.

Resampling

One could now think that we are ready to go, but we must take in consideration an important issue that was already mentioned earlier. It's the problem of class imbalance: we cannot train a machine learning model if 80% of the examples predict the same class. The model will be biased, it means that it will tend to always predict the majority class, and then will be of no utility. To overcome this issue resampling techniques can be used. Resampling is a technique used in the data science field to alter the distribution of data. Most used techniques are "oversampling" and "undersampling". Oversampling consists in increasing the number of samples in the minority class, Undersampling in

reducing the number of samples in the majority class. We will use an advanced oversampling technique called SMOTE. This technique creates new synthetic examples instead of oversampling by substitutions. SMOTE introduces synthetic examples to oversample minority class samples by generating new data points that mimic the distribution of the real ones.

Splitting & Scaling

After we applied the SMOTE technique to rebalance the classes, the data set is almost ready for modelling. We just need to perform a few more operations, to make sure the whole project goes on smoothly. The data set must be splitted into a *training set* and a *test set*. The training set, as the name suggests, will be used to train the model in predicting whether a group of transactions can be fraudulent based on its characteristics. The test set is a batch of data never seen by the model, it will be used to evaluate the model performance on unknown data.

Another important operation is *Scaling*. Scaling important because it standardises the feature values, ensuring that all features contribute equally to the model's learning process, which can improve the model's performance. Not scaling is a bad practice because it can lead to features with larger ranges dominating the model training, potentially resulting in poor model performance. Moreover, only the independent variables of the training set must be scaled. Scaling the dependent variable can distort the model's predictions and in our case it makes no sense at all since it’s a binary variable. When scaling data for machine learning, it's essential to ensure that the test set is scaled using the same parameters (mean, standard deviation) as the training set. If the test set were scaled independently, this would cause a data leakage, that can lead to overly optimistic performance estimates because the model has indirectly seen information from the test set during training.

5. Models Implementation

In this chapter, we are finally ready to dig into the practical aspects of applying machine learning techniques to AML detection. Two approaches will be tested, which have already been mentioned in Chapter 3: Outlier Detection and Risk Scoring. In the same chapter we also selected the most appropriate algorithms for each approach, based on the literature on the topic.

5.1 Outlier Detection

K-Means Clustering

We will begin by training and testing the K-Means algorithm. Being an unsupervised model, it doesn't require labelled data, after choosing the number of necessary clusters, it will autonomously identify groups of data points with similar underlying patterns. However, it will perform well only if the classes ('Reported' or 'Dismissed') are well-separated, meaning that patterns that lead to a set of transactions being reported or not are clearly distinguishable.

Before training the model, we performed a dimensionality reduction technique, known as Principal Component Analysis (PCA). PCA is commonly used to reduce the dimensionality of large data sets by converting a large set of variables into a smaller one that retains the majority of the information from the larger set. This will also come in hand to visualise the results of the clustering. It wouldn't be possible on a 10-dimensional space, while selecting the 2 principal components we have bidimensional points that can easily be plotted, while keeping most of the explained variance.

The first approach we try is to select only one cluster, and all the points with distances greater than the 95th percentile of the distances of the points from the cluster center, are considered outliers. If sets of fraudulent transactions behave differently than regular ones, they should be far from the cluster center, and thus considered outliers.

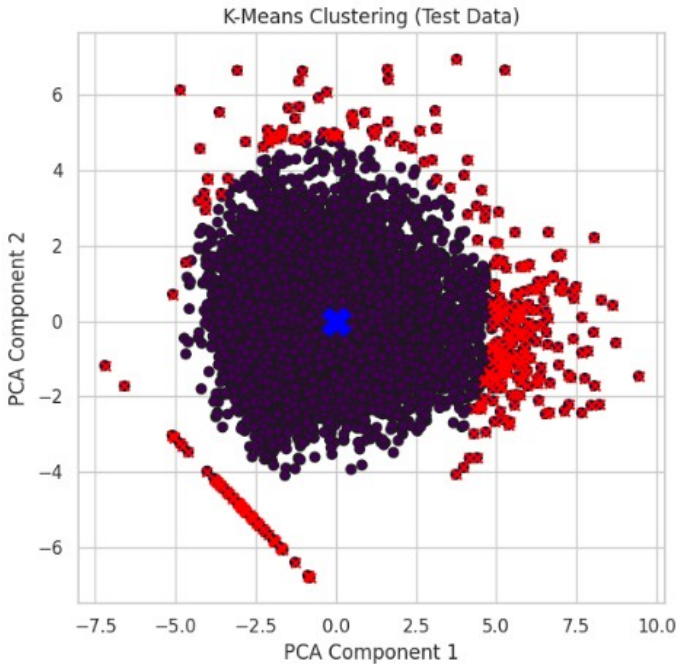

Figure 14: K-Means Single Cluster Visualization

The cluster has been created and the outliers have been identified, now we have to check whether the outliers correspond to the reported transactions or not.

```
Total Outliers in Test Data: 332
Outliers Corresponding to Reported Transactions: 162 (48.80%)
Outliers Corresponding to Non-Reported Transactions: 170 (51.20%)
```

We can see that there are actually more non-reported transactions in the outliers, this shows that this approach wasn't effective in spotting fraudulent transactions and this is likely due to the fact that classes overlap and thus this may not be the most appropriate method.

We could try other approaches on the same algorithm, but it wouldn't make much sense, since, as shown in the plot, the data points are too overlapping and there isn't a distinction among groups of points. In our code, we even tried by tuning different hyperparameters such as number of principal components and number of clusters, but the results haven't improved at all.

Isolation Forest

We proceed with the second outlier detection algorithm, Isolation Forest, again, an unsupervised learning algorithm. We start by fitting the Isolation Forest model on the training data with a contamination parameter that indicates the expected proportion of outliers. Then, predictions are made on the both training and test set and performance are evaluated and visualised. PCA is used here as well to reduce the dimensionality of data for visualisation purposes.

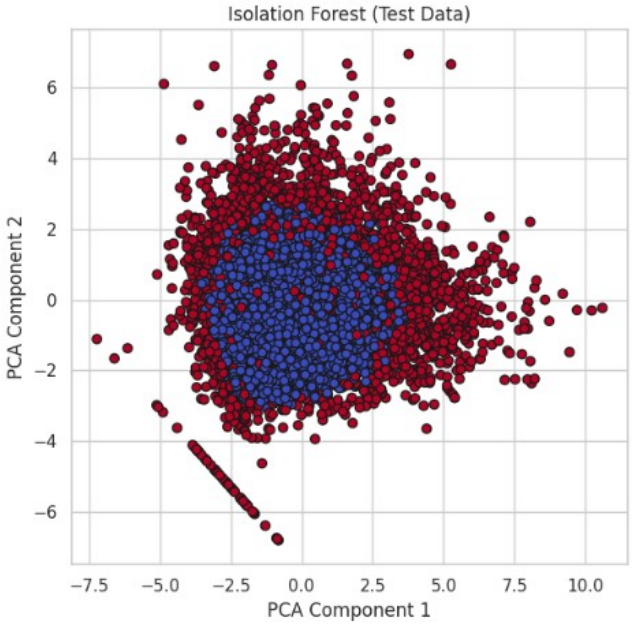

Figure 15: Isolation Forest Visualisation

Total Outliers in Test Data: 1987
Outliers Corresponding to Reported Transactions: 878 (44.19%)
Outliers Corresponding to Non-Reported Transactions: 1109 (55.81%)

The situation here is the same as before, outliers are detected, but they don't correspond to reported transactions. In fact, the number of outliers corresponding to non-reported transactions is even higher than the number of outliers corresponding to reported transactions, as in the K-Means clustering implementation. We can notice how

unsupervised algorithms for outlier detection are not effective with this data structure. Later we'll discuss the possible reasons of why this is the case.

K-Nearest Neighbors (KNN)

The last algorithm included in the "Outlier Detection" category is K-Nearest Neighbors. KNN is a supervised learning algorithm, it uses the labels of the K neighbouring data points to create a prediction. In our study we use the KNN classifier, provided by the machine learning Python library 'sklearn'. We start by initialising the classifier with a parameter corresponding to the number of neighbours to consider. We, then, fit it on the training data. Now the classifier is ready to predict the labels for the test data. The predictions are evaluated using a confusion matrix, and evaluation metrics such as precision, recall , f1-score and accuracy. To have a visual representation of how the model works, we will use PCA (again) to reduce the data dimensions to two and fit another KNN model to the reduced data points. This way we can plot the predictions and compare the plots of all the Outlier Detection algorithms we saw.

KNN achieves an accuracy of 73.1%. This indicates that the KNN model correctly classifies about 73% of the instances in the test set.

Precision and recall metrics:

Class 0 (Non-Reported Transactions):

- Precision: 0.75 - This means that 75% of the transactions predicted as non-reported are actually non-reported.
- Recall: 0.70 - This means that 70% of the actual non-reported transactions are correctly identified by the model.

Class 1 (Reported Transactions):

- Precision: 0.72 - This means that 72% of the transactions predicted as reported are actually reported.
- Recall: 0.76 - This means that 76% of the actual reported transactions are correctly identified by the model.

The KNN model, overall, shows an adequate level of performance with an accuracy over 70%. The evaluation metrics are balanced across both classes, this indicates that the model is reasonably effective at classifying both reported and non-reported transactions.

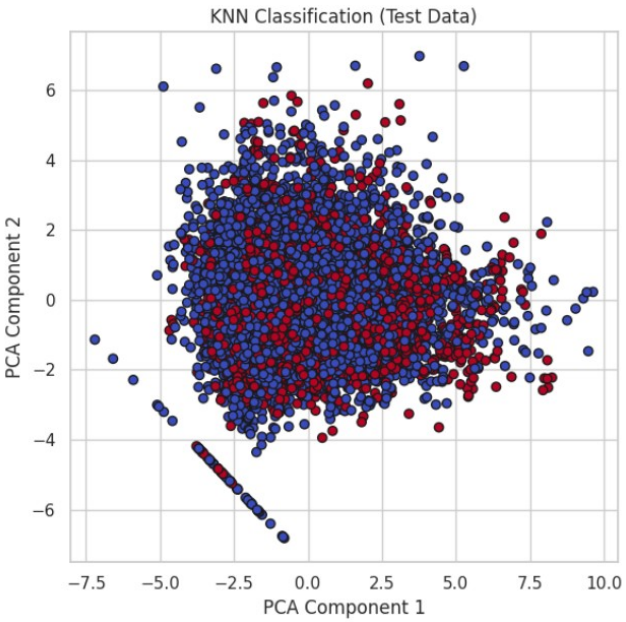

Figure 16: KNN Visualisation

The predictions made by the KNN algorithm are visibly very different from the previous two models. This shows how this algorithm is better at modelling high dimensional data sets with complex, non-linear relationships.

5.2 Risk Scoring

We proceed, now, to develop algorithms belonging to the "risk scoring" category. These are Support Vector Machine and tree-based algorithms such as CART (Classification and Regression Trees), Random Forest and XGBoost. All of these are supervised learning algorithms which means they require labelled training data to learn from.

Support Vector Machine (SVM)

SVM is a powerful classification algorithm that works by finding the hyperplane that best separates the classes in the feature space. If the data is not linearly separable, SVM

uses kernel functions to transform the data into a higher-dimensional space, where a linear separator can be found. It could be a perfect candidate for our data set because it's very effective in high-dimensional space and it can handle non-linear data with the appropriate choice of kernel.

At first, we tried to initialise the 'sklearn' SVM classifier with the RBF kernel, which is a kernel suitable for non-linear classification problems. SVM is very sensitive to hyperparameters, so we are going to use a hyperparameter tuning technique known as Grid Search. Grid search is a methodical approach to optimising hyperparameters by exploring a predetermined group of hyperparameters. It evaluates the performance of the model for each combination using cross-validation, and eventually determines the optimal set of parameters. Cross-validation is a method for evaluating a machine learning model by partitioning the dataset into several subsets, training the model on some of them and validating it on the other ones, and repeating this procedure to guarantee robust model evaluation.

Support Vector Machines demand substantial computational resources on high dimensional data, resulting in prolonged durations for parameter tuning and model training. The Sklearn library is designed to operate primarily on CPU architectures, which provide limited computational capacity and may not be suitable for computationally intensive tasks. That's why another library was used, ThunderSVM, that enables SVMs to harness GPU's computational power and makes the training process substantially faster.

After the best combination of parameters has been found, we train the so-called "best model" on the training set. Next, the model predicts the labels for the test data and finally the model is evaluated using a confusion matrix and its related evaluation metrics. We then visualise the results using the same approach we used for other models.

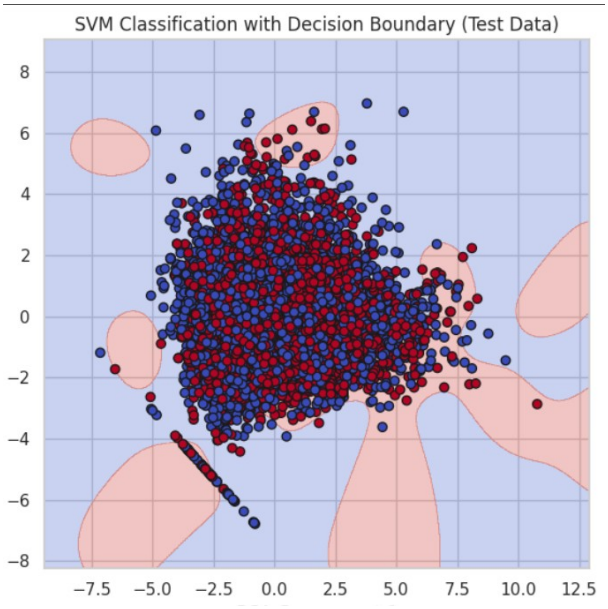

Figure 17: SVM Visualisation with Decision Boundary

The model achieves an accuracy of 81.6%.
Precision and recall metrics:

Class 0 (Non-Reported Transactions):

- Precision: 0.79 - 79% of the transactions predicted as non-reported are indeed non-reported.
- Recall: 0.87 - 87% of the actual non-reported transactions are accurately identified by the model.

Class 1 (Reported Transactions):

- Precision: 0.85 - 85% of the transactions predicted as reported are actually reported.
- Recall: 0.76 - 76% of the actual reported transactions are accurately identified by the model.

The SVM model shows a commendable performance with an accuracy of 81.6%. The precision and recall metrics for both classes highlight the model's efficacy in minimising false positives and false negatives, respectively. Class 0 exhibits higher recall, reflecting the model's strong detection capability for non-reported transactions.

Class 1 demonstrates high precision, indicating reliable identification of reported transactions.

Classification and Regression Trees (CART)

CARTs are decision trees algorithms that splits the data into subsets, based on the value of input features. Each node in the tree represents a feature and each branch represents a decision rule. Leaves represent the outcome. CART is easy to interpret and visualise, but it's susceptible to overfitting.

We initialise the Decision Tree classifier, again from the 'sklearn' library and we tune the hyperparameters with Grid Search. Here the parameters to be tuned are `'max_depth'` and `'min_samples_split'`. The former is used to control the maximum depth of trees, limiting how many levels it can have, the latter sets the minimum number of samples required to split an internal node, it's useful to prevent splits that would result in too small leaf nodes. Found the optimal parameters, the standard procedure is followed: training, prediction, evaluation and visualisation.

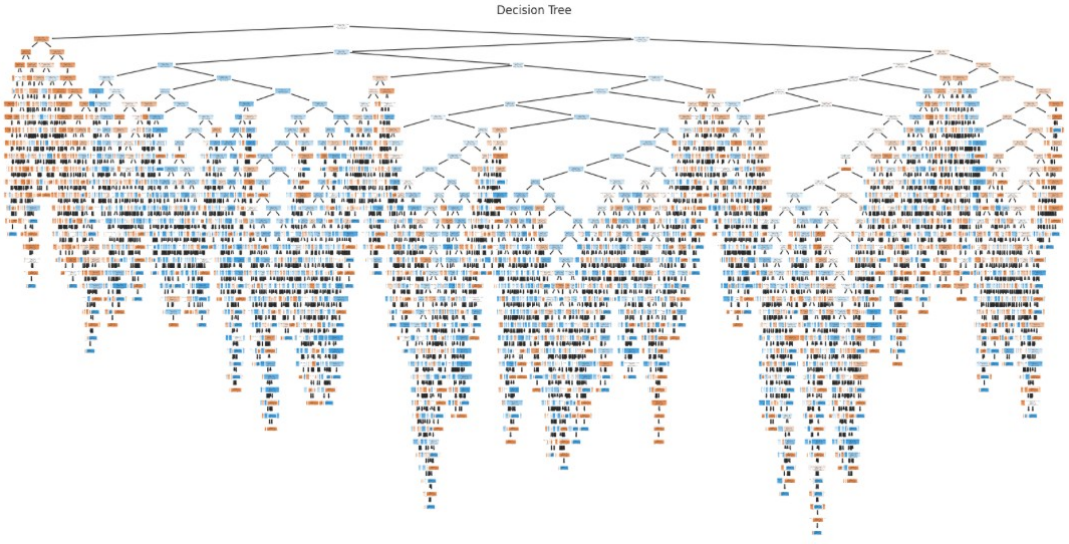

Figure 18: Decision Tree Visualisation

We can see that the decision tree that was created has a large number of branches and leaves, this reflects the complex relationship among the data set variables.

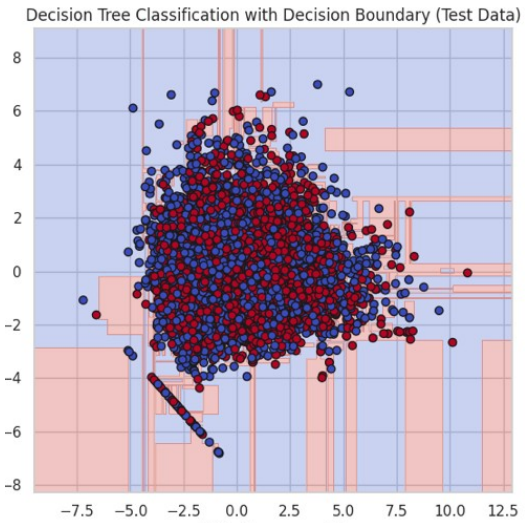

Figure 18: CART Decision Boundary

It's interesting to see how the decision boundary from SVM differs from the decision tree's one. SVM Produces smooth, rounded decision boundaries due to the non-linear transformation of the feature space and the smooth nature of the RBF kernel. Decision Trees produce squarish, piecewise constant decision boundaries due to axis-aligned splits and hierarchical partitioning of the feature space.

The best parameters found through grid search were max_depth: 'None' and min_samples_split: 2. This configuration allows the decision tree to grow without a depth limit and to split nodes with as few as two samples. This parameter setting maximises the tree's complexity and its potential to capture patterns in the data. However it might also lead to overfitting, which can explain the moderate performance observed.

The model achieves an accuracy of 70.0%.
Precision and recall metrics:

Class 0 (Non-Reported Transactions):

- Precision: 0.71 - This indicates that out of all the transactions predicted as non-reported, 71% were correctly identified as non-reported. The precision value here shows a balanced level of false positives.
- Recall: 0.67 - This reveals that 67% of the actual non-reported transactions were correctly classified. This recall suggests that there are a significant number of false negatives, where non-reported transactions are incorrectly classified as reported.

Class 1 (Reported Transactions):

- Precision: 0.69 - This means that 69% of the transactions predicted as reported were indeed reported transactions. This precision indicates that the model has a considerable rate of false positives for reported transactions.
- Recall: 0.73 - This indicates that 73% of the actual reported transactions were correctly identified. A recall of 0.73 reflects a strong ability to capture reported transactions, although there are still some false negatives.

The Decision Tree model demonstrates a not exceptional performance with an accuracy of 70.0%. The precision and recall metrics for both classes highlight the model's strengths and weaknesses. Class 0 (Non-Reported Transactions) has a higher precision than recall, indicating that the model is more conservative in classifying non-reported transactions, leading to fewer false positives but more false negatives. Conversely, Class 1 (Reported Transactions) shows a higher recall than precision, suggesting the model's better ability to identify reported transactions, albeit with a notable amount of false positives.

Random Forest (RF)

Random Forest is a technique in ensemble learning that builds several decision trees during training and produces the average prediction of these individual trees. Randomness is introduced by randomly selecting a subset of characteristics and data samples for each tree. It’s able to reduce overfitting by averaging multiple trees and it’s robust to noise and outliers. For the implementation of this model we follow the standard procedure, as with the other models. The hyperparameters of Random Forest

are the same as CART, with the addition of `'n_estimators'`, it refers to the number of individual decision trees that are built and used in the ensemble model to make predictions.

The best parameters identified through grid search were max_depth: 'None', min_samples_split: 2, and n_estimators: 300. This setup allows the Random Forest to grow without a depth limit and to utilise a large number of trees (300), enhancing its ability to capture intricate patterns in the data. The high performance observed indicates that these parameter settings effectively balance the model's complexity and generalisation capabilities.

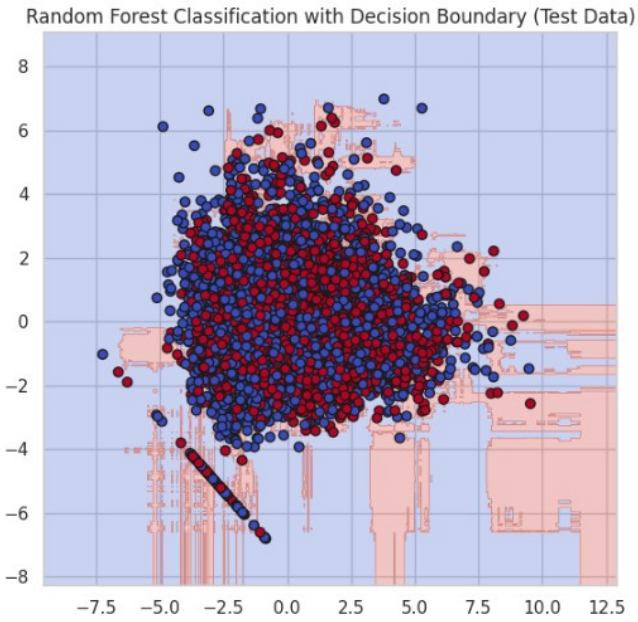

Figure 18: Random Forest Decision Boundary

The model achieves an accuracy of 83%.
Precision and recall metrics:

Class 0 (Non-Reported Transactions):

- Precision: 0.87 - This indicates that 87% of the transactions predicted as non-reported are indeed non-reported. This high precision shows a low rate of false positives for non-reported transactions.

- Recall: 0.78 - This reveals that 78% of the actual non-reported transactions were correctly classified.

Class 1 (Reported Transactions):

- Precision: 0.80 - 80% of the transactions predicted as reported were indeed reported transactions.
- Recall: 0.88 - This indicates that 88% of the actual reported transactions were correctly identified. A high recall reflects the model's strong capability to capture reported transactions, minimising false negatives.

The Random Forest model demonstrates strong performance with an accuracy of 83%. The precision and recall metrics for both classes highlight the model's effectiveness. Class 0 (Non-Reported Transactions) exhibits high precision, indicating that the model is reliable in identifying non-reported transactions with few false positives. The recall for Class 0 is slightly lower, suggesting some non-reported transactions are misclassified. Class 1 (Reported Transactions) shows a balanced performance with high recall and precision, suggesting the model's strong ability to identify reported transactions accurately, with minimal false negatives and a reasonable number of false positives.

XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced boosting algorithm that builds trees sequentially, with each tree correcting errors made by the previous ones. It uses a gradient descent algorithm to minimise the loss function, leading to highly accurate models. We proceed to train and evaluate the model. XGBoost hyperparameters are `'n_estimators'`, `'max_depth'` and `'learning_rate'`. We already have come to know the first two in the other tree-based algorithm, the last one is a hyperparameter that controls the contribution of each tree to the final model by scaling the weight updates, determining effectively how quickly the model adapts to the data.

The best parameters identified here were learning_rate: 0.2, max_depth: 15, and n_estimators: 500. The high number of estimators and the depth enable the model to capture intricate details in the data, while the learning rate ensures the model converges efficiently.

The model achieves an accuracy of 82.3%.
Precision and recall metrics:

Class 0 (Non-Reported Transactions):

- Precision: 0.86 - This indicates that 86% of the transactions predicted as non-reported are indeed non-reported. This high precision suggests a low rate of false positives for non-reported transactions.
- Recall: 0.78 - This reveals that 78% of the actual non-reported transactions were correctly classified. The recall value indicates that there are some false negatives, where non-reported transactions are missed.

Class 1 (Reported Transactions):

- Precision: 0.79 - This means that 79% of the transactions predicted as reported were indeed reported transactions. This precision indicates a moderate rate of false positives for reported transactions.
- Recall: 0.87 - This indicates that 87% of the actual reported transactions were correctly identified. A high recall reflects the model's strong capability to capture reported transactions, minimising false negatives.

The XGBoost model demonstrates strong performance with an accuracy of 82.3%. The precision and recall metrics for both classes highlight the model's effectiveness. Class 0 (Non-Reported Transactions) exhibits high precision, indicating that the model is reliable in identifying non-reported transactions with few false positives. The recall for Class 0 is slightly lower, suggesting some non-reported transactions are misclassified.

Class 1 (Reported Transactions) shows a balanced performance with high recall and precision, suggesting the model's strong ability to identify reported transactions accurately, with minimal false negatives and a reasonable number of false positives.

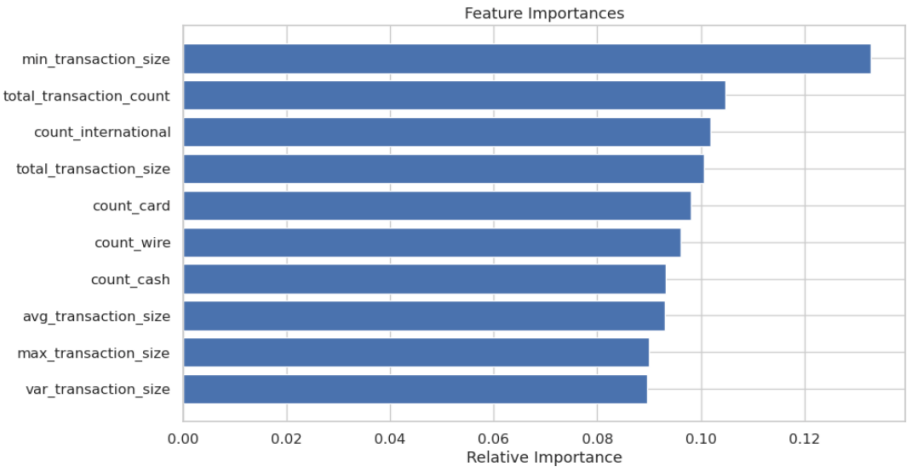

Figure 19: Feature Importance

The feature importance plot indicates that ‘min_transaction_size’ is the most critical factor in predicting fraudulent transactions, followed by ‘total_transaction_count’ and ‘count_international’. This suggests that the minimum transaction size, the total number of transactions, and the count of international transactions are key indicators of potential fraud.

5.3 Models Comparison

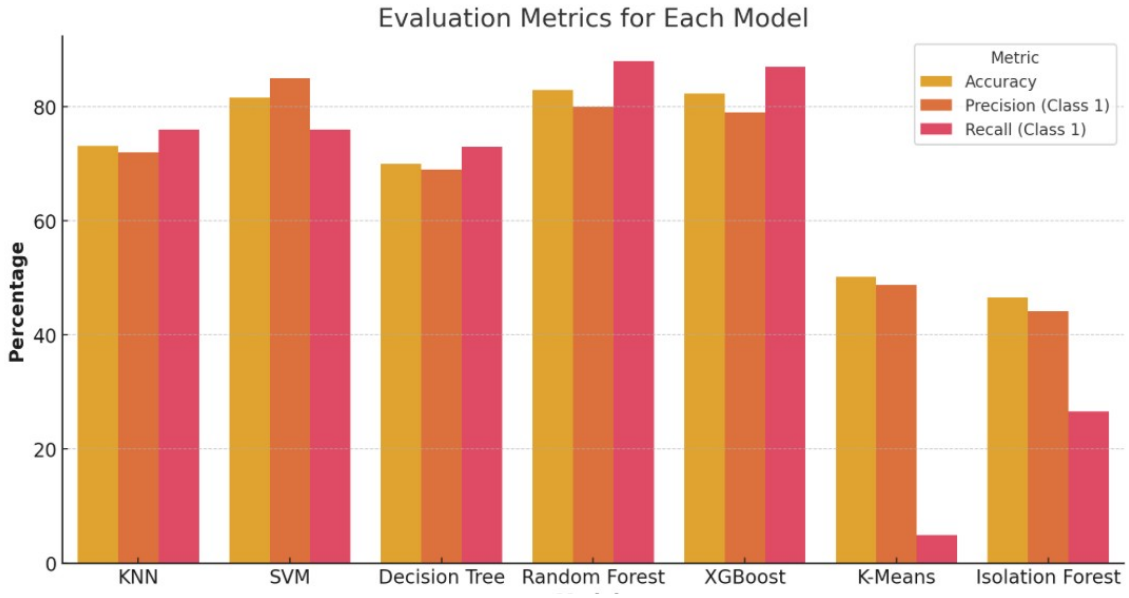

Figure 20: Evaluation Metrics Comparison

Outlier Detection

In the outlier detection category, we evaluated Isolation Forest, K-Means Clustering, and K-Nearest Neighbors (KNN) for detecting fraudulent transactions. The first two are unsupervised learning models. To ensure a consistent comparison with other models, we adapted the metrics from these unsupervised algorithms to the common evaluation metrics used for supervised models. Specifically, we treated the identified outliers as predicted positives and non-outliers as predicted negatives. By aligning the confusion matrix and calculating precision, recall, and accuracy, we could effectively compare their performance against the supervised models. Isolation Forest and K-Means Clustering both showed significant limitations. Isolation Forest, which isolates anomalies, struggled with the high dimensionality and complex, non-linear relationships in the data, leading to poor performance with an accuracy of 46.6%, precision of 44.2%, and recall of 26.6%. Overlapping distributions of inliers and outliers further hindered its effectiveness. Similarly, K-Means Clustering, assuming spherical clusters and linear boundaries, performed poorly in capturing non-linear patterns, resulting in an accuracy of 50.2%, precision of 48.8%, and recall of 4.9%.

K-Nearest Neighbors (KNN), however, outperformed the other two. Its non-parametric nature and ability to handle complex, non-linear relationships enabled it to achieve an accuracy of 73.1%, with balanced precision (72%) and recall (76%). KNN's flexibility in not assuming specific data distributions made it more effective for detecting fraudulent transactions.

Risk Scoring

SVM showed strong performance with an accuracy of 81.6%, high precision (85%), and recall (76%) for reported transactions. Its ability to capture complex patterns in the data contributed to its high accuracy. However, SVMs can be computationally intensive and require careful tuning of parameters like the kernel type and regularisation.

Decision Tree provided moderate performance with an accuracy of 70.0%, precision of 69%, and recall of 73% for reported transactions. While Decision Trees are easy to interpret and visualise, their tendency to overfit and reliance on linear splits limited their effectiveness in capturing the complexities of fraudulent transactions.

Random Forest emerged as one of the top performers, with an accuracy of 82.9%, high precision (80%), and recall (88%) for reported transactions. By aggregating multiple decision trees, Random Forest mitigated the overfitting issue and captured complex patterns more effectively. Its robustness and high performance make it a strong candidate for detecting fraudulent activities.

XGBoost also demonstrated strong performance with an accuracy of 82.3%, precision of 79%, and recall of 87% for reported transactions. XGBoost's ability to handle missing values, regularisation features, and ensemble approach allowed it to capture intricate patterns in the data efficiently. Its scalability and performance make it another top choice for this task.

While all models in the risk scoring category performed well, Random Forest stands out as the best algorithm. Its combination of high accuracy, precision, and recall, along with robustness against overfitting and ability to handle complex patterns, makes it the most

effective model for detecting fraudulent transactions. XGBoost is a close second, offering comparable performance with additional benefits in handling data complexities.

Conclusion and Discussion

6.1 Key Findings

The goal of this study was to understand how AI could be implemented in the anti-money laundering processes of financial institutions to strengthen and streamline financial security. The results offer significant perspectives on the wider difficulties and possibilities related to utilising artificial intelligence for AML purposes.

6.1.1 Data Quality

One of the fundamental aspects of implementing AI in AML is the quality of the data. High quality data is essential to develop algorithms able to spot suspicious activities effectively. There is a need for data that is accurate, thorough, and reflective of real-life scenarios. Still many obstacles have to be overcome, such as data privacy concerns and the need for large data sets to train AI models and even the most sophisticated AI models will struggle to execute without high-quality data.

6.1.2 The right approach

For AML detection to be effective, it's important to select the appropriate approach. Different models have different characteristics, and their performance might vary greatly based on the nature of the data and the specific AML tasks. As we saw, some models are better than others when data points have complex non-linear relationships. Understanding this is crucial for financial institutions to choose a model tailored to their needs. There isn't a one-size-fits.all solution, the results depend on the context and the goals of the analyses. The adaptability of models to new types of transactions is becoming even more important, with the advancement of the financial technologies (FinTech) and decentralised finance (DeFi).

6.1.3 Efficiency

It is important, during the implementation of AI models, to consider model efficiency, in terms of training time and computing power. Every day, financial institutions handle millions of transactions, thus, they need models capable of processing and analysing data fast and reliably. The processing power necessary to manage such enormous

amounts of data is significant, hence institutions must invest in strong infrastructure to support these systems. Efficient models improve detection capabilities while also ensuring that AML processes are quick and scalable. Cloud computing and technology improvements have enabled institutions to improve the efficiency and scalability of their AI-driven AML systems.

6.1.4 Adapting to emerging trends

The financial sector is constantly changing. The expansion of digital banking, cryptocurrencies, and online financial services has added new complexities to the ecosystem. Institutions' ability to react to these changes and detect new patterns of financial crime is critical. As economies become more interconnected and transactions become faster and more frequent, the demand for improved AML solutions that can keep up with these trends grows more urgently than ever.

6.2 Implications

AI can't only be used alone to spot fraud at the moment because of its limited technological capabilities. A lot of progress has been made in the field of financial security using AI to strengthen anti-money laundering protocols. Even though modern AI systems are very advanced, human control and judgement are still needed to make sure that illegal financial activity is found completely and correctly.

An important thing that artificial intelligence algorithms have shown is that they can find trends and patterns that humans might miss. These systems are very good at quickly handling large datasets and finding problems with a high level of accuracy. However, because financial deals are complicated and vary from case to case, AI systems may give both false positives and false negatives, which means that human agents are still needed to check and make decisions.

In addition, current regulatory systems and legal obligations require human agents to be a part of the AML system. It is essential to have the knowledge of human professionals in order to understand the results of AI, make smart choices about transactions that have been flagged, and make sure that legal standards are followed. AI can make human

agents much smarter and more efficient by giving them powerful analysis tools for quickly and accurately finding signs of possible fraud. However, AI can't replace (completely) yet the understanding and personal judgement that human agents bring to the table.

AI being used in AML methods will have big effects on the financial sector because it means that fraud detection systems will become smarter and better able to predict what will happen. Less money spent on operations, better detection rates, and faster response times will all be additional benefits for financial organisations.

6.3 Future Directions

The ability to mask fraudulent transactions may become increasingly challenging as smart algorithms improve their capacity to detect it. Stronger data verification, more computational power, and updated regulations will assist these systems. Here are few potential transformations that could occur in the future:

6.3.1 Collaborative AI models

A novel approach involves developing AI systems that may be collectively utilised by many financial institutions. They would utilise shared data to more effectively identify and prevent the concealment of funds. Federated learning, a technique where models are trained across multiple decentralised servers holding local data samples without exchanging them, could play an important role. By utilising large, diverse datasets, all parties are able to enhance the performance and security of intelligent algorithms.

6.3.2 Regulatory Sandboxes

The individuals responsible may establish secure locations where financial institutions can experiment with new artificial intelligence technologies without violating any regulations. These locations provide an opportunity for them to experiment with new ideas and ensure their effectiveness in real-world scenarios, while also ensuring compliance with legal requirements. According to the World Bank, these designated areas facilitate the secure and regulated growth of financial technology. The knowledge

acquired here has the potential to enhance the performance of AI systems and ensure their adherence to established guidelines.

6.3.3 Natural Language processing and Generative AI

Utilising advanced artificial intelligence techniques such as natural language processing (NLP) and generative AI has the potential to have a significant impact. NLP analyses textual data such as messages, emails, and transaction logs to identify suspicious financial transactions. These tools have the ability to thoroughly analyse complex hiding strategies and enhance the AI's predictive capabilities.

Advanced artificial intelligence will significantly influence the battle against financial fraud. Collaboration facilitated by federated learning, regulatory sandboxes, and the utilisation of advanced AI technologies such as NLP and simulated AI will contribute to higher financial security in the international landscape. Even if there are still challenges, AI continues to improve and has the potential to significantly enhance the security of financial transactions.

As we look ahead, let us remember the words of Bob Dylan: *"The times they are a-changin'"*. This change, driven by AI and human ingenuity, promise a new era in the fight against financial crime, one where vigilance and innovation go hand in hand.

References

McKinsey. "The Fight Against Money Laundering: Machine Learning Is a Game Changer." *McKinsey & Company*, https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/the-fight-against-money-laundering-machine-learning-is-a-game-changer.

Unger, Brigitte, and Frans van Waarden. "Attempts to Dodge Drowning in Data: Rule- and Risk-Based Anti Money Laundering Policies Compared." *Working Papers*, no. 09-19, 2009, Utrecht School of Economics

Parkman, Tim. *Mastering Anti-Money Laundering and Counter-Terrorist Financing*. Pearson, 2012.

Han, Jingguang, et al. "Artificial Intelligence for Anti-Money Laundering: A Review and Extension." *Digital Finance*, vol. 2, 2020, pp. 211-239, https://doi.org/10.1007/s42521-020-00023-1

Jensen, R.I.T., et al. "A Synthetic Data Set to Benchmark Anti-Money Laundering Methods." *Scientific Data*, vol. 10, 2023, article 661, https://doi.org/10.1038/s41597-023-02569-2

United Nations Office on Drugs and Crime. "Organized Crime Module 4 Key Issues: Money-Laundering." *UNODC Education for Justice*, https://www.unodc.org/e4j/en/organized-crime/module-4/key-issues/money-laundering.html

"The Financial Action Task Force (FATF)." *Embassy of Italy in Paris*, https://italiarapparigi.esteri.it/en/litalia-e-le-ooii/altre-organizzazioni/the-financial-action-task-force-fatf/

Bitquery. "Coinpath MoneyFlow - Crypto Investigation Tool." *Bitquery*, https://bitquery.io/products/moneyflow.

Rhahla, Mouna, et al. "Guidelines for GDPR Compliance in Big Data Systems." *Journal of Information Security and Applications*, vol. 61, 2021, article 102896, https://doi.org/10.1016/j.jisa.2021.102896

GeeksforGeeks. "Types of Machine Learning." *GeeksforGeeks*, https://www.geeksforgeeks.org/types-of-machine-learning/

Grünwald, Peter D. *The Minimum Description Length Principle*. MIT Press, 2007

Jiang, Hui. *Machine Learning Fundamentals*. Cambridge University Press, 2021

IBM. "What Is the k-Nearest Neighbors Algorithm?" *IBM*, https://www.ibm.com/topics/knn

GeeksforGeeks. "What Is Isolation Forest?" *GeeksforGeeks*, https://www.geeksforgeeks.org/what-is-isolation-forest/. Accessed 28 May 2024

Serokell. "K-Means Clustering in Machine Learning." *Serokell Blog*, https://serokell.io/blog/k-means-clustering-in-machine-learning

Freie Universität Berlin. "Decision Tree Classification." *Freie Universität Berlin*, https://www.geo.fu-berlin.de/en/v/geo-it/gee/3-classification/3-1-methodical-background/3-1-1-cart/dectree.png?width=1000

Gunning, David, et al. "XAI—Explainable Artificial Intelligence." *Science Robotics*, vol. 4, 2019, article eaay7120, https://doi.org/10.1126/scirobotics.aay7120

The World Bank. "Key Data from Regulatory Sandboxes across the Globe." *The World Bank*, https://www.worldbank.org/en/topic/fintech/brief/key-data-from-regulatory-sandboxes-across-the-globe

Appendix A: Google Colab Notebook

This notebook contains the code and experiments conducted for this study. It includes the implementation of various algorithms, data preprocessing steps, and results of the analysis.

Access the notebook here: Prova Finale AML – Vitale.ipynb