Ministry of Science and Higher Education of the Republic of Kazakhstan

Suleyman Demirel University

Amankossova Aruzhan

Automating banking sector monitoring procedures for exceptional situations

THESIS

Presented in Partial Fulfilment for the

Master of Technical Sciences Degree in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Associate Professor, Ph.D. Cemil Turan**

Kaskelen 2023

Suleyman Demirel University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty

Associate Professor

PhD Zhamanov A.

06 06 2023

Topic of the thesis:

Automating banking sector monitoring procedures for exceptional situations

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" SDU, 2021-2023

Head of Department _____ Assistant Professor, PhD Mukash Zh.

Academic Supervisor _____ Associate Professor, PhD Cemil T.

Master student _____ Amankossova A.

Kaskelen 2023

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Aruzhan Amankossova

2023

Acknowledgements

I would like to express gratitude to my supervisor, Cemil Turan, for giving me clear direction for my early efforts. My deepest gratitude goes to my parents, brother, and sisters, who supported me. I would not be where I am today without them.

I would like to thank my friends and coworkers for their encouragement and support, who offered useful feedback and direction during the study process. Their ideas and insights have been quite helpful to me. I also want to express my gratitude to the teachers and staff at my university, who gave me the tools and chances I needed to finish my paper. Their dedication to study and teaching has served as a continuing source of encouragement and inspiration.

I want to express my heartfelt gratitude to everyone who has helped me with both my academic and personal efforts. I am so glad that you are in my life because your continuous support and motivation have helped me accomplish my objectives.

Dedication

This thesis is dedicated to:

My parents, Cemil Turan, Kamila Orynbekova, Gulnar Baktybayeva, Ualikhan Sadyk and many other for their support, help, sense of humour and useful comments for improvingthis project.

Abstract

The need to detect anomalous events and react to them immediately in real time is becoming increasingly important in the banking sector. The main objective of this thesis is to propose the development of a real time alert notification system that uses outlier detection algorithms to discover unexpected trends in the key performance indicators of a financial industry. In order to enable real-time monitoring of data streams and notify users of anomalous occurrences as they happen, the system will take advantage of the capabilities of cloud computing and big data technologies. The proposed system will be evaluated against traditional outlier identification techniques. The efficacy of the outlier detection algorithms for the banking dataset is assessed using precision, recall, and F1 score measurements. The approaches of sending alerts are evaluated, with the strengths and weaknesses of each method taken into account. This thorough evaluation approach aims to emphasise the advantages and disadvantages of the suggested system as well as identify potential areas for improvement. The suggested system will allow users to take proactive action to lessen the consequences of abnormal events, reduce the risk of costly downtime and other adverse effects.

Аңдатпа

Аномальды оқиғаларды анықтау және оларға нақты уақыт режимінде дереу әрекет ету қажеттілігі банк секторында барған сайын маңызды болып келеді. Бұл дипломдық жұмыстың негізгі мақсаты қаржы саласының негізгі тиімділік көрсеткіштеріндегі күтпеген тенденцияларды анықтау үшін шектен шығуды анықтау алгоритмдерін пайдаланатын нақты уақыттағы ескерту хабарландыру жүйесін әзірлеуді ұсыну болып табылады. Деректер ағындарын нақты уақыт режимінде бақылауды қосу және пайдаланушыларды орын алған ауытқулар туралы хабардар ету үшін жүйе бұлттық есептеулер мен үлкен деректер технологияларының мүмкіндіктерін пайдаланады. Ұсынылған жүйе дәстүрлі сәйкестендіру әдістеріне қарсы бағаланады. Банктік деректер жинағы үшін шектен шығуды анықтау алгоритмдерінің тиімділігі дәлдік, қайта шақыру және F1 ұпай өлшемдері арқылы бағаланады. Ескертулерді жіберу тәсілдері бағаланады, әрбір әдістің күшті және әлсіз жақтары ескеріледі. Бұл мұқият бағалау тәсілі ұсынылған жүйенің артықшылықтары мен кемшіліктерін атап көрсетуге, сондай-ақ жақсартудың ықтимал бағыттарын анықтауға бағытталған. Ұсынылған жүйе пайдаланушыларға әдеттен тыс оқиғалардың салдарын азайту, қымбат тұратын тоқтап қалу қаупін және басқа да жағымсыз әсерлерді азайту үшін белсенді әрекет етуге мүмкіндік береді.

Аннотация

Необходимость обнаружения аномальных событий и немедленного реагирования на них в режиме реального времени становится все более актуальной в банковской сфере. Основная цель этой диссертации - предложить разработку системы оповещения в реальном времени, которая использует алгоритмы обнаружения выбросов для обнаружения неожиданных тенденций в ключевых показателях эффективности финансовой отрасли. Чтобы обеспечить мониторинг потоков данных в режиме реального времени и уведомлять пользователей об аномальных событиях по мере их возникновения, система будет использовать возможности облачных вычислений и технологий больших данных. Предлагаемая система будет оцениваться по сравнению с традиционными методами идентификации выбросов. Эффективность алгоритмов обнаружения выбросов для набора банковских данных оценивается с использованием измерений точности, отзыва и оценки F1. Оцениваются подходы к отправке оповещений с учетом сильных и слабых сторон каждого метода. Этот тщательный подход к оценке направлен на то, чтобы подчеркнуть преимущества и недостатки предлагаемой системы, а также определить потенциальные области для улучшения. Предлагаемая система позволит пользователям принимать упреждающие меры для уменьшения последствий нештатных ситуаций, снижения риска дорогостоящих простоев и других неблагоприятных последствий.

Abbreviations

SDU Suleyman Demirel University
ML Machine Learning
LOF Local Outlier Factor
OCSVM One-class Support Vector Machine
IF Isolation Forest
IQR Interquartile range
SMTP Simple Mail Transfer Protocol
AAMAS Automated Attendance Management and Alert Systems
SMS Short Message Service
IETF Internet Engineering Task Force
LGBM Light Gradient Boosting Machine
GBDT Gradient Boosted Decision Tree
GBoost Gradient Boosting
XGBoost Extreme Gradient Boosting
PII Personally Identifiable Information
API Application Programming Interface
SMOTE Synthetic Minority Oversampling Technique
GMail Google Mail
BSD Berkeley Software Distribution
AWS Amazon Web Services
SES Simple Email Service
EC2 Elastic Compute Cloud
MRI Magnetic resonance imaging

Table of Contents

Chapter 1

Background and motivations

1.1 Introduction

Big data and real-time data streams have multiplied exponentially in the banking sector in recent years. This increase in data has given data experts and managers a variety of new opportunities as well as entirely new challenges to overcome. One of the most serious difficulties these bank data professionals face is the need to notice abnormal events in real-time in order to take early actions to prevent or reduce the impact of such incidents.

This is due to the increasing extent of banking as well as the sheer number of financial transactions that occur every day. Banks are struggling with identifying and stopping dishonesty and other illegal activities. Therefore, with the growth of digital banking and the rising use of technology in financial transactions, the significance of real-time alert notification in the banking sector has become more and more clear.

Real-time alerting of abnormal or critical bank performance is important to bank safety and security. The prevention of financial misconduct can be aided by the detection of unusual financial behaviour. Additionally, it may assist in locating the financial system's risks and weaknesses as well as shield against any interruptions. Banks frequently use approval metrics to assess consumer creditworthiness and decide whether to accept loan applications. It may be a sign of

mistakes in the decision to approve a loan if there is a rapid rise or fall in the number of loans given. For instance, if there is a significant increase in high-risk loan applications that are authorised, this may indicate that the bank is taking on too much risk and improperly evaluating the creditworthiness of the applicants. Customers expect quick and effective loan processing times, and any mistakes or delays in the approval procedure may result in customer dissatisfaction and even lost revenue.

In order to increase the precision of credit risk assessments, it is crucial in the banking industry to spot discrepancies in approval rates. Outlier identification can assist banks in identifying potential loan defaulters in the context of credit risk assessment. Banks can modify their risk scoring models to better account for these higher-risk borrowers by spotting variances in approval levels, which can help reduce default losses. It can also be employed to look for probable wrongdoing. An indication of falseness might be seen, for instance, if a bank's acceptance rate for credit card applications suddenly increases dramatically and then maintains a high level for an extended period of time. Banks can analyse falsehood errors and take the necessary steps to avoid similar behaviours by recognising these anomalies. If a bank doesn't adhere to business key performance identificators like approval levels, it runs a lot of risks. One of the main worries is the risk of making poor judgements. If a bank's loan acceptance rate is too high, it can discontinue making loans to high-risk customers, which might lead to increased defaults on loans and financial losses for the company. On the other hand, if the bank's loan acceptance rate is too low, it can end up rejecting creditworthy clients, missing out on potential revenue streams.

Another risk linked to noncompliance with business objectives is inadequate planning and budgeting. A bank may overestimate or underestimate its financial performance when predicting income and spending, which might result in a budget deficit or surplus. This may occur if the bank's loan approval rate is overlooked. If the business metric is not achieved, the bank's reputation may ultimately be in jeopardy. A bank may lose the confidence of its investors and clients if it gains a reputation for bad credit judgement, which might eventually cause operations to decrease.

Real-time anomaly event detection in data streams is a challenging problem with many facets that calls for sophisticated computational and analytical techniques. Data professionals may fully use the capabilities of real-time anomaly detection systems in the financial sector by utilising the most recent developments in big data technology. The combination of statistical analysis and machine learning algorithms that can find patterns and trends in huge and complicated datasets is necessary for the timely identification of such occurrences. These technologies can help financial organisations respond to unusual occurrences promptly and efficiently, lowering the likelihood of financial loss and damage to reputation.

A possible approach for spotting outliers is to employ statistical or machine learning approaches that detect abnormalities in univariate time series data. Although these techniques have demonstrated potential for identifying abnormal occurrences, traditional approaches have a high incidence of false positives and are not appropriate for real-time monitoring. Furthermore, the complex and diverse nature of financial data flows necessitates the deployment of modern data pre-processing and data cleaning techniques to assure the quality and trustworthiness of the insights gained.

To address this issue, the dissertation recommends implementing a real-time warning and notification system that integrates the traditional way of detecting deviations with the major indications of the bank's activity.

The main goal of this dissertation is to develop a system for detecting anomalous events and compare its performance with outlier detection approaches.

This research paper aims to investigate and provide a comprehensive analysis of the research question posed "How to effectively and quickly identify abnormal behaviour in a critical banking indicator?", drawing on relevant sources and data to offer a well-supported answer. The research question has been chosen based on its significance and relevance to the field of study, and it has been carefully constructed to be clear, succinct, and researchable.

1.2 Literature Review

The first step in determining a response to this issue is to thoroughly review the relevant scientific literature in order to demonstrate proficiency in this field. It is important to undertake research and stay current with best practises and industry trends in order to exhibit a high degree of expertise and competence in the subject matter of the request. A literature review is a method for gathering information from previously published scientific works and applying it to one's own work and decision-making. This technique helps researchers produce new insights and ideas that may be utilised to direct their own study as well as get a thorough understanding of the current state of the art in their particular subject. The thesis will first review the body of knowledge about outlier detection and real-time data stream monitoring in order to accomplish this. The review will cover both the latest developments in statistical approaches and algorithms for machine learning used to find abnormalities in time series data, as well as warning notification systems.

My interest in this subject has been sparked by P. Robert's research work [1] on setting up an email-based notification system for large-scale system resources. P. Robert details how he successfully developed an alerting system using Python, Filebeat, and Watchdog in the article he published. This innovative approach has the potential to increase company efficiency and simplify resource management by sending out timely warnings and alerts. Future research in this field, building on Robert's work, may look into the integration of machine learning, which might improve the system's performance and expand its ability to recognise anomalies. Businesses may achieve operational excellence and gain a competitive advantage in today's fast-changing business environment by leveraging the most recent technological breakthroughs and developing innovative solutions.

In addition to the work already stated, Mohammad Braei and Dr.-Ing. Sebastian Wagner's experimental study [2] on anomaly identification in univariate time series offers useful information to businesses looking to streamline their processes. Braei and Wagner used numerous domain-specific datasets in their study to compare the performance of deep learning methods to statistical and classical

machine learning procedures. Their failure to assess their approaches using financial data was one of the study's flaws. Our dissertation seeks to further this work by investigating how these techniques are applied in the banking industry. The financial sector, which generates enormous volumes of complex and varied data, provides unique possibilities and problems for data analysis.

The paper "A simple heat alert system for Melbourne, Australia" by Neville Nicholls, Carol Skinner, and Margaret Loughnan [3] presents a heat alert system designed for the city of Melbourne, Australia. The authors begin by outlining the significance of heatwaves as a problem for public health, particularly in urban areas. The next section of the article gives a thorough explanation of the heatwave warning system, which is based on a set of objective standards that sends warnings when certain thresholds are surpassed. The method takes humidity, wind speed, cloud cover, and maximum and lowest temperatures into account. The authors also discuss the communication methods employed to inform the general public and significant stakeholders of heatwave warnings. The use of objective criteria to include heat warnings and the use of communication tactics to spread heat warnings to the general public and important stakeholders are both covered in the article. The authors do not go into specifics about how the notifications were disseminated, but they do say that they were spread via a number of channels, including media reporting, social media, and email alerts. The examination of temperature and fatality data for the city of Melbourne over the summers of 2009–2010 is then used to evaluate the heat wave warning system. The outcomes demonstrated that the heat wave warning system was successful in lowering heat-related fatalities.

The article "A Comparison of Outlier Detection Algorithms for Machine Learning" by H. Jair Escalante [4] is a survey of the literature that contrasts several outlier identification techniques for machine learning. The relevance of outlier identification in machine learning and the numerous applications of outlier detection methods are discussed by the author in the first paragraphs. The paper then goes into detail about a number of outlier identification techniques, including statistical techniques, distance-based techniques, clustering-based techniques, and density-based techniques. The author discusses the benefits and drawbacks

of each type of algorithm and provides examples of how they are used in various situations. He employed the principal component analysis of the kernel to decrease the dimensionality of the spectral data. This method allowed him to transform the data into a new set of variables that captured most of the original information while greatly reducing the amount of data that needed to be analysed. The outcome was a more organised set of data from which valuable insights and patterns could be drawn. Data that was both noisy and not noisy was used to conduct the research. The author also offers a comparison of outlier detection techniques based on performance indicators including accuracy, repeatability, and F1 score. The comparison's findings demonstrate that no outlier identification method is better than another and that the right strategy to use depends on the characteristics of the data and the specific application. The paper also discusses interpretability, the curse of dimensionality, and unbalanced datasets as problems and challenges with outlier identification in machine learning. The author also discusses the most recent advancements in outlier identification, such as ensemble approaches and deep learning-based algorithms. The key findings are outlined in the article's conclusion, which emphasises the importance of detecting outliers in machine learning and the demand for more study in this field. However, this paper provides a comprehensive examination of machine learning outlier identification techniques, making it a significant resource for researchers and industry professionals.

A notable addition to the field of anomaly identification is made by Zhangyu Cheng, Chengming Zou, and Jianwei Dong's study on outlier detection utilising isolation forests and local outlier factors [5].The authors offer a two-layer progressive ensemble method for finding outliers that uses the isolation forest and local outlier factors. The study also offers experimental results that demonstrate how effective this approach is in detecting anomalies. The approach that is suggested combines the advantages of both LOF and IF machine learning algorithms to achieve high accuracy in outlier detection while minimising false positives. In the context of business and finance, this is essential since false positives can have costly repercussions. This is crucial in a corporate and financial environment because false positives may be expensive. Financial organisations may avert losses

and reduce risk by rapidly spotting potential deviations. The article provides helpful details on utilising ensemble methods to look for abnormalities in time series data. Effective outlier detection is an essential aspect of the analysis of data, as businesses depend steadily on it to make wise decisions. By building on the work of Cheng, Zou, and Dong and providing insights that will aid organisations of all types in increasing their efficiency and remaining competitive in an industry that is changing quickly, our thesis seeks to further this important field of study.

The article on Automated Attendance Management and Alert Systems, authored by A. Rahim and P. Ismail [6], is a valuable contribution to the field of automation and sending alerts. The authors created an AAMAS system utilising MS Visual Basic, MS Access, and MS Excel, which provides enterprises with an automated and efficient manner of controlling staff attendance. The system can only send SMS alerts, as indicated in the article, and email notifications are not supported. This may be a significant financial factor for enterprises because SMS notifications can be pricey and may not always be the most economical option. As a result, it is critical to analyse the cost and efficacy of various notification techniques, such as email notifications, to decide the optimal solution for a certain organisation. This can assist businesses in streamlining their management procedures and reducing expenses without compromising precision or effectiveness. The article stresses the necessity for organisations to carefully consider the costs and advantages of various notification techniques in order to make educated decisions about attendance management and offers insightful information on the design and implementation of real-time alerts. This paper, however, aims to build on this work by investigating the usage of various notification systems for automated alerts in order to provide organisations new concepts and tactics to improve their operations.

The purpose of this work is to design, construct, and evaluate a real-time alert-notification system that identifies outliers in univariate time series data using machine learning techniques and statistical computing, taking into account the shortcomings of previous comparable research papers. The primary objective will be to concentrate on proactive monitoring and warning of anomalous occurrences, allowing users to take prompt and efficient action to reduce the effects of such

situations.

The core ideas and findings of the dissertation were featured in two papers that were published in the journal "SDU Bulletin". The research conducted in this dissertation incorporates some of the results and experiments presented in those papers in order to comprehensively examine all of the methods under investigation. We list here the publications written during this Master's thesis. The thesis is based on the following articles, listed in order of appearance in the text:

1. Amankossova A.; Turan C. "Implementation of a real-time alert-notification system for data monitoring in the financial industry", Suleyman Demirel University Bulletin: Natural and Technical Sciences, [S.l.], v. 62, n. 1, p. 132 - 141, mar. 2023. ISSN 2709-2631 [7].

2. Amankossova A.; Turan C. "An evaluation of unsupervised outlier detection methods for univariate time series data in financial transactions", Suleyman Demirel University Bulletin: Natural and Technical Sciences, [S.l.], v. 62, n. 1, p. 178 - 188, mar. 2023. ISSN 2709-2631 [8].

The research is innovative in that it combines machine learning algorithms and statistical methodologies with a real-time warning and notification system to identify outliers in a financial dataset. This strategy may make proactive monitoring and communication of unusual occurrences possible, which may be very helpful in important industries like banking. The project seeks to contribute to the development of more accurate as well as effective ways for recognising and managing risks related to unusual occurrences in real-time data streams by applying this system. The findings of this work will aid in the development of strong and practical outlier identification systems with potential financial sector applications. As a result, this research study aims to add something worthwhile to the body of knowledge and deepen our grasp of the topic.

The architecture and design of the suggested real-time alert notification system will then be given in the dissertation's remaining pages. The system will be built to handle massive amounts of data streams, and it will combine big data technologies with computing to enable real-time monitoring and alerting.

The thesis will then describe the implementation of the system, including the integration of statistical and machine learning methods to detect outliers in time series data. The implementation will also include a practical example to demonstrate the effectiveness of the system in detecting anomalous events and alerting users in real time.

Subsequent to it, the thesis will present the results of an evaluation of the proposed system, including a comparison of its performance with traditional outlier detection approaches. The evaluation will be carried out using metrics such as accuracy, recall, and F1 score and will demonstrate the effectiveness of the proposed system in detecting anomalous events in real time.
The results of the research include the development of an online anomaly notification system that was implemented using Python and a library for working with data and calculations. The experiment utilised data from the Home Credit Bank database, and the precision and recall evaluation metrics of various anomaly detection methods were compared, along with their respective advantages and disadvantages. Also the study analysed and compared the pros and cons of different methods for sending notifications. This dissertation will contribute to the development of more efficient and effective outlier identification systems with potential applications across a wide range of industries. The suggested method will enable users to take preventative action to lessen the consequences of unexpected circumstances, boost overall productivity, and lower the possibility of expensive downtime or other negative effects.

1.3 Overview of Chapters

It is essential to provide a clear summary of the thesis at the end of the introduction. Six chapters make up the structure of the thesis. This helps the reader have a clear understanding of the subsequent chapters, which provide the context and goals in Chapter 1. We gave a concise summary of our system and the procedures used in our work in Chapters 2, 3, and 4, highlighting the essential components and processes used within the scope of our processes. The formula and results for getting an input test using templates are provided in Chapter 5 of the study.

This chapter also contains the code's outcomes and additional information on the experiment. The results of our investigation were given in Chapter 6, which also included potential areas for further research.

Chapter 2

Methodology

A dissertation's methodology section describes the research strategies we employed. It is essential to carry out a comprehensive examination of suitable methods for spotting outliers and include mechanisms for prompt notice distribution prior to the construction of this system.

Figure 2.1 depicts the three-level architecture that was adopted to ensure the system's functionality. There is no direct connection between the thin client layer and the database. The data is kept in a database, and the application logic is developed using a programming language. Scalability, maintainability, and adaptability are just a few advantages of using a three-level design. Additionally, it enables the division of concerns, which makes it simpler to update or alter one layer of the system independently of the others.

To achieve this objective, our algorithms were executed on the Home Credit Bank dataset. To overcome this challenge, we generated synthetic timestamp using python libraries. As our data set's target value was not evenly distributed, we applied the SMOTE technique to balance it.
The synthetic minority resampling method is an algorithmic approach used to balance skewed datasets. The process of developing synthetic samples of the minority class entails using each sample from the minority class individually to produce fresh synthetic examples along the line segments linking the minority class's closest neighbours [9]. This aids in balancing the data distribution and

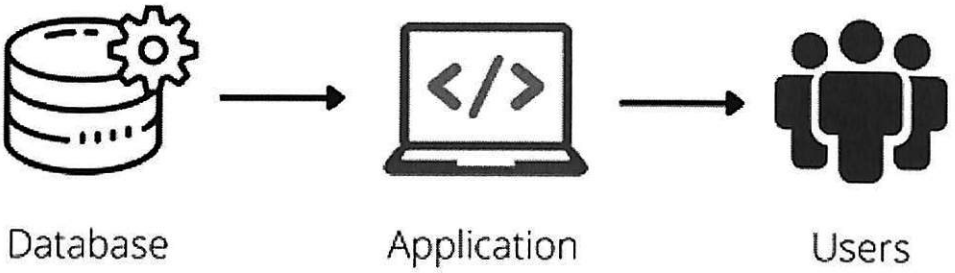


Figure 2.1: Three-level architecture of the alert system

Table 2.1: Labels

City
Almaty
Moscow
Kaskelen
London

guards against model bias in favour of the dominant class.

It is crucial to do data preprocessing if the dataset contains categorical data types. Encoding a dataset entails converting the raw data into a format that is appropriate for machine learning. The properties of the dataset may increase or decrease depending on the encoding technique used.

It is necessary to create a distinct binary variable for each group in the source variable in order to implement a one-time encoding. City names may be transformed into binary columns using one-time encoding.
For example, if the original categorical variable is city and has classifications shown in Table 2.1 such as Almaty, London, Kaskelen, and Moscow, then one-hot encoding would create three new variables: city_Almaty, city_London, city_Kaskelen, and city_Moscow. The values for each of these variables would be either 0 or 1, depending on whether that specific value for the original variable was present in a particular observation. One-hot encoding is useful when there is no inherent order to the categories in the original variable. Almaty can be encoded as [1, 0, 0, 0],Kaskelen as [0, 0, 1, 0] , London as [0, 0, 0, 1] and Moscow as [0, 1, 0, 0] (Table 2.2).

Table 2.2: Labels after one-hot encoding

City	city_Almaty	city_Moscow	city_Kaskelen	city_London
Almaty	1	0	0	0
Moscow	0	1	0	0
Kaskelen	0	0	1	0
London	0	0	0	1

Contrary to this, label coding necessitates giving a unique numerical value to each category included in the original data. This indicates that, for purposes of model building, each category is represented by a distinct integer number. In this example, label encoding may be used to assign the values 1, 2, and 3 to the categories. When the categories in the source variable have a natural order, label encoding is advantageous. We should minimise the number of features in the data set while maintaining as much information as feasible.

We are working with a categorical variable in this situation that reflects several occupational groups. This variable was label encoded to make it machine readable. Each occupation is given a label by the label encoder, which in this case uses the values 0 to 5 from the Table 2.3. This enables the use of a person's employment status as a feature by machine learning algorithms that need quantitative input. This, however, is not the ideal application for label encoding. The use of number labels suggests an ordinal connection between categories, where one category is more significant or superior than another, which is the rationale for this. In reality, this is not the case; rather, the categories differ and should be handled as such. Direct coding, where each category is represented by a binary value, would be a preferable method for encoding this categorical data. This assures that there is no ordinal link between categories and that each category is treated equally by the machine learning algorithm.

Consider the case when the categorical variable designates the several categories of different professions (Table 2.3). In this example, the label encoder assigns the label from 0 to 5 (Table 2.5).This allows us to use the gender of a person's data in machine learning algorithms that require numerical input.

Feature selection is a strategy for improving the performance of machine learn-

Table 2.3: Label encoding

Profession
Teacher
Unemployed
Farmer
Doctor
Painter
Hairdresser

After label encoding

Profession	Encoded
Teacher	0
Unemployed	1
Farmer	2
Doctor	3
Painter	4
Hairdresser	5

Table 2.4: Labels before label encoding

Labels
Men
Women

ing algorithms by detecting and deleting unnecessary variables while maintaining just the most significant ones. To assure the accuracy of the classifier while dealing with large datasets, feature selection must be done beforehand. The feature selection process tries to find out a subset of features $a \subseteq b$, where a is the optimal subset and b is the set of features. One of the primary research inquiries in feature selection involves identifying the optimal subset.

Four key phases are involved in feature selection (see Figure 2.2):

- Formation of subsets
- Evaluation of subsets
- Criteria for stopping
- Validation

Subset generation is a search procedure that uses a certain search methodology. A produced subset feature is compared to the most recent best subset feature using a specific evaluation criterion. If it turns out that the new feature subset is superior to the prior best feature subset, it will take its position. This method keeps going until a predetermined stopping requirement is satisfied. Validation is required after the creation of the ideal feature subset.

The feature selection process employs the LightGBM approach. The one-way

Table 2.5: After label encoding

Labels	Encoded
Men	1
Women	0

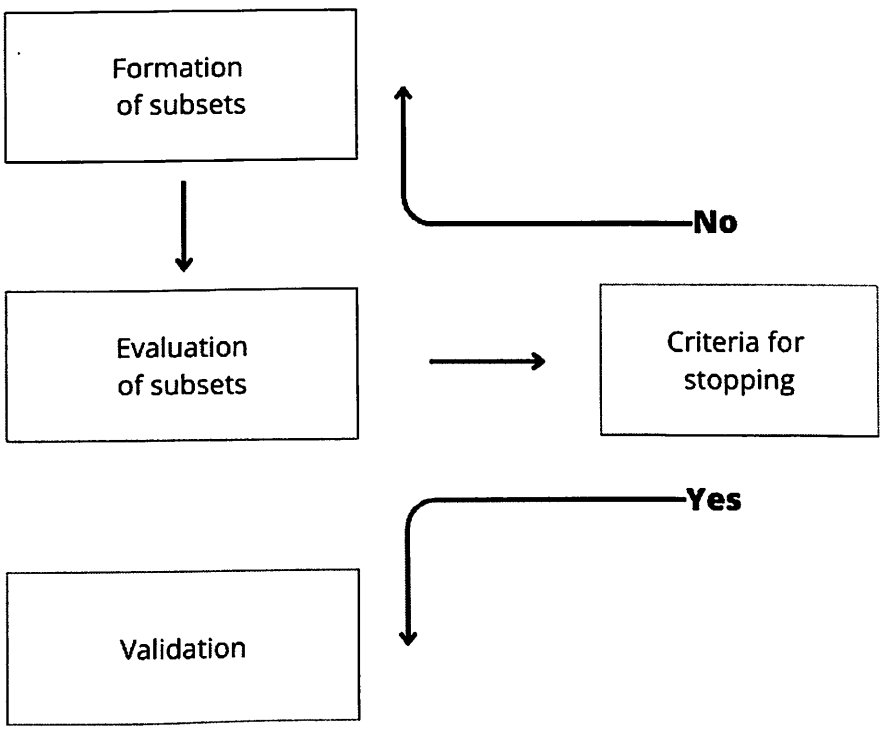


Figure 2.2: Steps of Feature Selection

gradient-based sampling decision tree approach is the foundation of the GBDT design. Although the GBDT technique may transform continuous functions into discrete values, it only makes use of first-order derivative data when optimising loss functions. Because each tree learns from the inferences and residuals of prior trees, decision trees can only be employed as regression trees. A decision tree strategy based on histograms has been used by LightGBM. As a result, XGBoost uses less memory and processes enormous volumes of data more quickly and accurately, with fewer false positives and missed detections. In order to reduce overfitting and boost performance, LightGBM utilises multi-threaded optimisation and maximum tree depth. This removes duplication and irrelevant data from the data collection, increasing the model's overall predictive power. This method is effective for data

visualisation and accelerating the training of machine learning models.

Outlier detection systems are often assessed based on precision, recall, and F1 scores. It is usual and useful to assess the efficacy of outlier identification systems using these measures.

The ratio of genuine hits to all anticipated hits is known as precision, whereas the ratio of true hits to all actual hits is known as recall [10].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2.0.1}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2.0.2}$$

The F1 score is the harmonic mean of precision and recall and provides a single measure of algorithmic performance [11].

$$F1\ Score = \frac{True\ Positive}{True\ Positive + \frac{1}{2}(False\ Positive + False\ Negative)} \tag{2.0.3}$$

or

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.0.4}$$

These metrics are used to measure how well the outlier detection algorithm performs in identifying data abnormalities. Researchers and practitioners may evaluate many techniques for outlier identification and select the one that is best suited for their specific use case by scoring the algorithm using these metrics.

Chapter 3

Outlier Detection

The term "anomaly detection" refers to the detection of outliers or abnormalities in data. This technique is used to find substantial shifts in data points that might signal errors or other unusual behaviour. Both statistical and unsupervised techniques may be used to find outliers.

Time series data cannot be used to discover anomalies in non-temporal data, such as location-based information. One of the primary methods for finding anomalies in geographic data is to calculate the difference between anomalous points and the rest of the data. The dataset may also be aggregated to identify all locations situated in less populated areas as anomalies. It is crucial to keep in mind that the fundamental tenet of geographic data is that the data points are independent of one another.

Time series data, on the other hand, reacts differently due to the interdependence of the data points. It is presumable that a sequence's later timestamps are influenced by its earlier data points. As a result, the sequence's values either undergo progressive modification or take on a predictable pattern. Any abrupt deviations from the pattern are viewed as abnormalities.
For instance, consider an example of a time series showing the percentage of approved applications, recorded every 10 minutes:

49 50 47 53 95 51 51 45

The majority of approaches would instead find two evenly distributed clusters if the data points were assumed to be independent, which would prevent them from seeing any unique patterns. In the context of time series, the sudden increase in the approval rate from 53% to 95% should be identified as an anomaly. Due to the interdependence of timestamps, it is possible that abnormalities in time series data are frequently contextual or collective in character.

The difference between the expected value x_{i-1} and the actual value x_i is used to calculate the anomaly score in time series:

$$\varepsilon_i = d(x_{i-1}, x_i)$$

where d is a distance function. In univariate time series, usually the euclidean distance is used. The deviation ε_i – is anomaly score. If this score is above a threshold δ, it is classified as an outlier.
There are several unsupervised machine learning methods that can be used for outlier detection.

3.1 Density-based method

Density-based clustering is an important problem of research for data scientists and has been investigated with interest in the past. These methods identify outliers based on the density of the data points in a particular region. Points that are in low-density regions are more likely to be outliers than points in high-density regions. Example of density-based methods include LOF, One-Class SVM.

3.1.1 Local Outlier Factor

The local outlier factor is a frequently utilised unsupervised machine learning technique for detecting outliers. Breunig et al. first introduced it in 2000 as a density-based method that quantifies the local deviation between a data point and its neighbours [12]. LOF identifies data points with a substantially lower density than those of their neighbours as outliers [13]. The following are the main definitions for the LOF:

Definition 3.1.1 (Distance k from a data point). A mathematical notion known as Euclidean n-dimensional space may be used to calculate the separation between two data points, p and o.

$$d(p,o) = \sqrt{\sum_{i=1}^{n}(p_i - o_i)^2} \tag{3.1.1}$$

Definition 3.1.2 (K nearest neighbors). A data point q is regarded as p's k nearest neighbour if the distance between them is less than or equal to k-distance(p). The k-nearest neighbours of q make up the k-distance neighbourhood of p according
to the above equation [14].

$$N_{k\_distance(p)}(p) = q \in A\ p | d(p,q) \leqslant k\_distance(p) \tag{3.1.2}$$

Definition 3.1.3 (The distance between p and o at which anything is reachable). We will assign a positive integer value to k. The equation explains how to calculate the reachability distance from data point p to data point o.

$$reachability\_distance_k(p,o) = max(k\_distance(o), d(p,o)) \tag{3.1.3}$$

Definition 3.1.4 (Density of data points for local reachability). Density-based clustering algorithms rely on two parameters to determine density:

- the minimum number of data points (MinD)
- volume

$$Lrd_{MinD}(p) = 1/\frac{\sum_{o \in N_{MinD}(p)} reachability\_distance_{MinD}(p,o)}{MinD(p)} \tag{3.1.4}$$

To determine the local reachability density (Lrd) of data point p, the average reachability distance based on its MinPts nearest neighbors is calculated first. Then, the inversion of this value produces the local reachability density (Lrd) of data point p.

Definition 3.1.5 (Local Outlier Factor of data point p). The equation shown below is used to calculate the LOF (local outlier factor) estimate for data point p based on the work mentioned above [13].

$$LOF_{MinD}(p) = \frac{\sum_{n \in N_{MinD}(p)} \frac{Lrd_{MinD}(o)}{Lrd_{MinD}(p)}}{|N_{MinD}(p)|} \tag{3.1.5}$$

The Local outlier factor algorithm works by first defining a k-distance neighbour hood of each data point, which consists of the k nearest neighbours of the point and their distances from the point [8]. The k-distance is the distance to the k-th nearest neighbour and serves as a measure of the local density of the point. Points with a higher k-distance are in less dense regions and are more likely to be outliers.

The next stage is to calculate each point's reachable distance, which quantifies its reachability from neighbouring points. The highest value between two points' distances and the second point's k-distance is used to calculate the accessible distance. The accessible distance for a place that is close to its k nearest neighbours is the larger of the k-distance or the distance to the k-th nearest neighbour.
A visual depiction of the algorithm's operation is presented in Figure 3.1. Once the k nearest neighbours of each data point have been identified, the local reachability density can be determined by dividing the inverse by the average reachability distance of its neighbours. By comparing the local reachability density of each point with the average local reachability density of its k nearest neighbours, the LOF value can be derived [14].

3.1.2 One-Class SVM

The One-Class SVM is a type of support vector machine that can be utilised for detecting anomalies in an unsupervised setting [8]. A one-class SVM, unlike standard SVMs, attempts to detect patterns in an unlabeled dataset rather than learning from a labelled dataset. In multidimensional space, this approach seeks a hyperplane that divides the data from the origin. The hyperplane should be as close to the data points as is feasible while being as far away from the origin

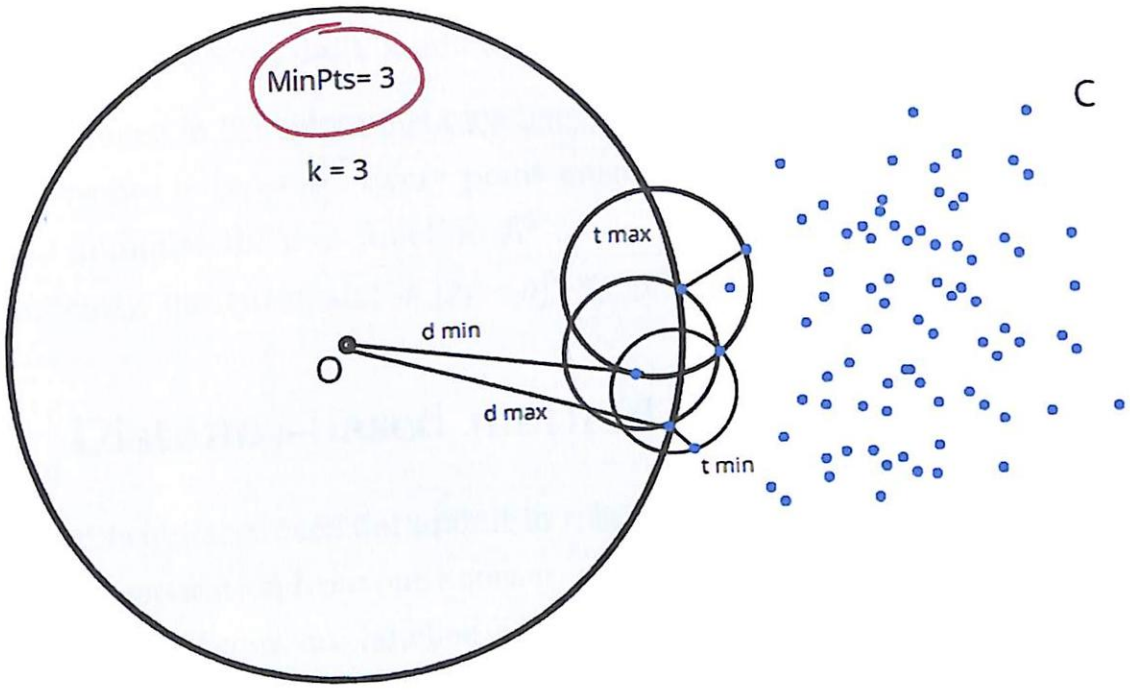


Figure 3.1: Local Outlier Factor

as is feasible. This method may be applied to tasks like novelty identification, which includes finding unique or previously unidentified patterns in an untagged dataset. According to their location in reference to the hyperplane, fresh data points may be classified as normal or anomalous using the technique, which builds a hyperplane.

The ocsvm function is used by one-class support vector machines to detect abnormalities in training data. The ocsvm function, which also offers anomaly scores and indications for the training set of data, is used to train one-class SVM objects. By giving clean training data (data devoid of outliers) to OCSVM, it produces a OneClassSVM object and then uses the object and the new data to detect abnormalities in older data. For the new data, the isanomaly function offers scores and anomaly markers.

There are two variations of one-class SVMs. The initial formulation, introduced by Schölkopf, entails employing a hyperplane as the decision boundary, whereas the second formulation, developed by Tax and Duin, involves using a hypersphere as the decision boundary. In this regard, let's delve more into the latter for-

mulation. The hypersphere has a radius-based centre $R > 0$ called a. Let's also imagine that we have n data points that are provided by x_i, where $i = 1, 2, \cdots, n$.

The distance in Euclidian distance between a specific data point and the hyper sphere's centre is $|x_i - a|$. Every point must be on or within the hypersphere in order to minimise the cost function R^2 .
Consequently, our constraint is $|x_i - a|^2 \leq R^2 \; \forall i$.

3.2 Distance-based method

This approach places each data point in relation to its closest neighbours by calculating their separation from one another, and data points that deviate significantly from their neighbours are labelled as outliers. The simplicity of implementation and ability to perform effectively on high-dimensional data are two benefits of distance-based outlier identification approaches. Additionally, they don't demand any presumptions regarding the fundamental distribution of the data. They may not perform well on datasets with variable densities or atypical distributions, and they can be sensitive to the parameters chosen, such as the number of nearest neighbours to take into account. The isolation forest is an illustration of a distance-based approach.

3.2.1 Isolation Forest

The isolation forest approach is another popular unsupervised machine learning method for finding outliers. The Isolation Forest algorithm selects a feature and a split value at random for each partition in order to partition the data. This indicates that the algorithm builds a tree structure, with each branch of the tree isolating the aberrant data points. Anomalies are positioned nearer to the tree's base because they may be separated with fewer splits than regular points. This approach allows the computer to detect abnormalities more quickly and precisely than other, more conventional approaches [8]. After that, the algorithm repeatedly divides the data into smaller and smaller subsets, eventually reducing each subset to just one data point. As an indicator of how readily a data point can be isolated from the rest of the data, the depth of each partition is kept track

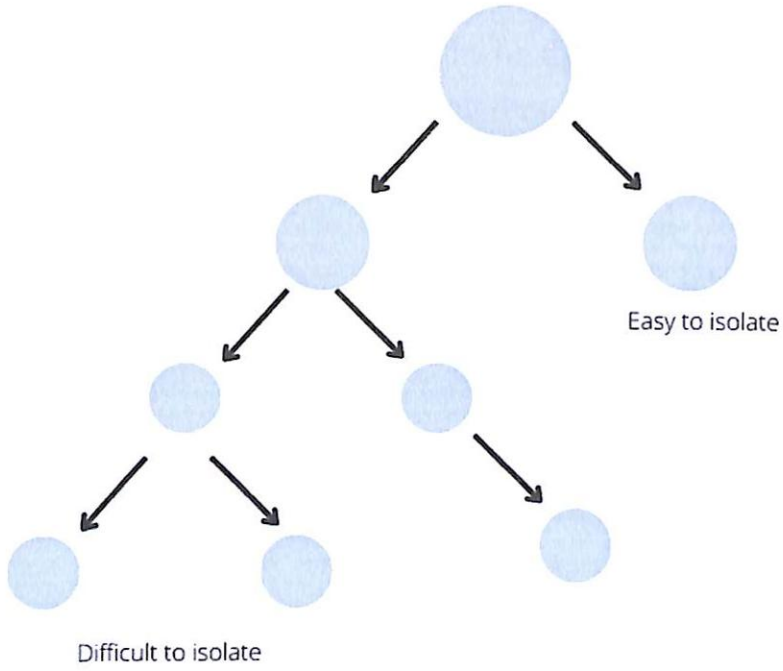


Figure 3.2: Isolation Forest

of.

Outliers are more easily isolated and therefore require fewer partitions to be separated from the bulk of the data. This means that outlier data points will have shorter route distances from the tree's root to their respective leaf nodes than normal data points. Figure 3.2 displays a graphical representation of how the algorithm operates:

The Isolation Forest algorithm uses path lengths to determine the degree of "isolation" of each data point and identifies points with shorter path lengths as outliers [15]. The Isolation Forest technique has several advantages over other outlier identification methods. It can scale large data sets with millions of data points and handle very high-dimensional data. Because the approach may be parallelized due to the random splitting of the data points, it is also computationally efficient. The Isolation Forest approach is more adaptable to many types of data since it does not rely on any presumptions about the underlying distribution of the data.

The Isolation Forest approach may not be as efficient, however, when dealing

with datasets that have a lot of overlap or when there are lots of clusters of data points that are similar, since these might be mistakenly classified as outliers. Additionally, the selection of tuning parameters, such as the number of trees in the forest and the subsampling rate utilised for each tree, might affect how well the isolation forest method performs [16]. The Isolation Forest algorithm is an adaptable outlier identification technique that may be used with a variety of datasets. When utilised correctly, the method can help find anomalous data points that may be of interest or suggest data mistakes.

The algorithm's performance on each unique dataset should be thoroughly assessed, and its shortcomings and possible hazards should be understood.

3.3 Statistical method

A statistical model may be fitted to the data using one of these techniques, and outliers—points that the model is unlikely to produce—can be identified. Statistical techniques are fast at processing vast volumes of data and are computationally efficient. Banks can now analyse huge datasets and find outliers more quickly and effectively than they could before using manual techniques.
Statistical techniques are adaptable and may be tailored to meet the special requirements of a given data collection or application. This implies that banks may select the statistical techniques that best suit their unique requirements and modify them as necessary to secure the best outcomes. One of several statistical techniques that may be used to identify outliers is the interquartile range approach.

3.3.1 Interquartile Range

The interquartile range (IQR) is a statistical method used for detecting outliers in a dataset. This range is obtained by determining the difference between the 75th percentile (Q3) and the 25th percentile (Q1) of a set of data [8]. This measure provides an effective way to gauge the middle 50% of a dataset and is considered to be more robust against extreme values than other popular measures of spread, such as the range or standard deviation. The use of the IQR method has several

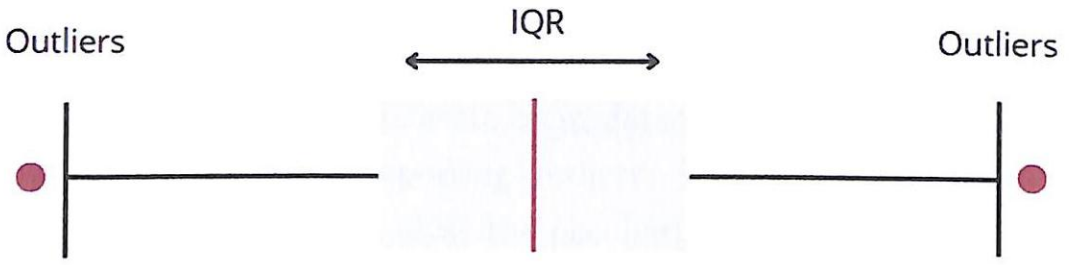


Figure 3.3: Interquartile Range

advantages in detecting outliers. The IQR method is more resistant to outliers than other measures of spread, such as the range or standard deviation, since it focuses on only the middle 50% of the data.As a result, it is a more trustworthy and accurate way of locating outliers. The IQR approach is adaptable and simple to modify to meet the unique requirements of a data collection or application. The IQR can be calculated as follows [17]:

$$IQR = Q3 - Q1$$

To use the IQR to detect outliers, one common approach is to define a "boundary" based on the IQR, and identify any data points that fall outside this fence as potential outliers. One common method is to use the following rule:

$$Lower\ boundary = Q1 - 1.5 * IQR$$

$$Upper\ boundary = Q3 + 1.5 * IQR$$

Potential outliers are any data points that are outside of the lower boundary or inside of the upper boundary. These potential outliers should be further examined to determine if they are legitimate data points or if they are data errors or anomalies. Figure 3.3 illustrates a graphical representation of the IQR method employed by the algorithm in question.

The use of IQR for outlier detection provides a number of benefits. It is more resistant to extreme values than other variance measurements like range or standard deviation, making it a robust technique. It is widely used in numerous

scientific fields and is also quite simple to compute and analyse.

Additionally, the IQR approach has significant drawbacks. For instance, in extremely small or very big datasets or in datasets with a skewed distribution, it could not be successful in spotting outliers. When employing the approach, researchers should carefully analyse the peculiarities of their data because using a single criterion to create the fence may not be acceptable for all datasets. In conclusion, the interquartile range is a popular technique in several study areas for identifying outliers in a dataset. To guarantee proper implementation of the approach while utilising it, researchers should be aware of its benefits and drawbacks as well as carefully analyse the peculiarities of their data.

Each of these approaches has advantages and disadvantages of its own. The best approach will be determined by the details of the data and the issue at hand. Some techniques could work better with high-dimensional data, while others might work better for locating outliers. Since some approaches need more processing power than others, the choice of method may be influenced by the computational resources available. Additionally, it is important to note that unsupervised machine learning algorithms for outlier detection are not perfect and may result in false positives or miss certain outliers. To ensure the validity and use of the discovered outliers, it is crucial to carefully assess the findings of outlier detection and integrate them with domain expertise and human judgement.

Chapter 4

Automated email notifications

The banking industry is one of several businesses that may use automated email alerts, which are a common automation option. They may be used for a variety of things, such as order confirmations, welcome messages, reminders about abandoned shopping carts, and more. Automated email alerts, when properly set up and optimised, may promote sales, foster client loyalty, and raise brand recognition. This technique may be used effectively to notify key players who are in charge of monitoring important performance data for the bank. For example, automated email notifications can be used to advise management of the progress of loan applications, the performance of various investment portfolios, or any potential fraud or security breaches. They can be customised and subdivided for greater efficacy, but they should be pertinent, timely, and beneficial to the receiver. Banks may use automatic email notifications in this case to improve communication, speed up response times, and assist stakeholders in making decisions based on data quickly. Automated email notifications may ultimately help banks improve customer service, reduce operational costs, and promote company growth.

Businesses may find it difficult and costly to develop email systems for the mass dissemination of transactional and marketing information. To ensure a high

rate of email delivery success, organisations must maintain email servers, network configurations, and strict Internet Service Provider rules for email content.

Here are some of the techniques that can be used to send automated email notifications:

4.1 SMTP

The Simple Mail Transfer Protocol is an essential and widely used communication protocol that ensures the smooth transmission of email messages across different servers and clients on the internet [18]. Its primary function is to ensure the delivery of emails from the sender's email client to the recipient's email server by following a set of guidelines and standards [7]. These guidelines include the message format, error handling, and message encoding. SMTP enables users to send and receive email messages from various email clients and service providers in a standardised manner. It was developed by the IETF, and most email clients, such as Microsoft Outlook, Gmail, and Apple Mail, provide support for it.

The way the SMTP protocol operates is by connecting the email clients of the sender and the receiver, then formatting the email message into the required format before transmission. Once the message is received by the recipient's email server, which then sends it to the recipient's email client, thereby completing the email transmission process, An SMTP session consists of several phases:

- connecting to the server;
- receiving the synchronous messages;
- transmitting sender, recipients and contents of a message
- disconnecting from the server

Establishing a connection between the sender's email client and the recipient's email server is how SMTP operates. The email message is then formatted in the appropriate format and transmitted to the server via the SMTP protocol. The message is obtained by the recipient's email server, which then sends it to the recipient's email client [7].

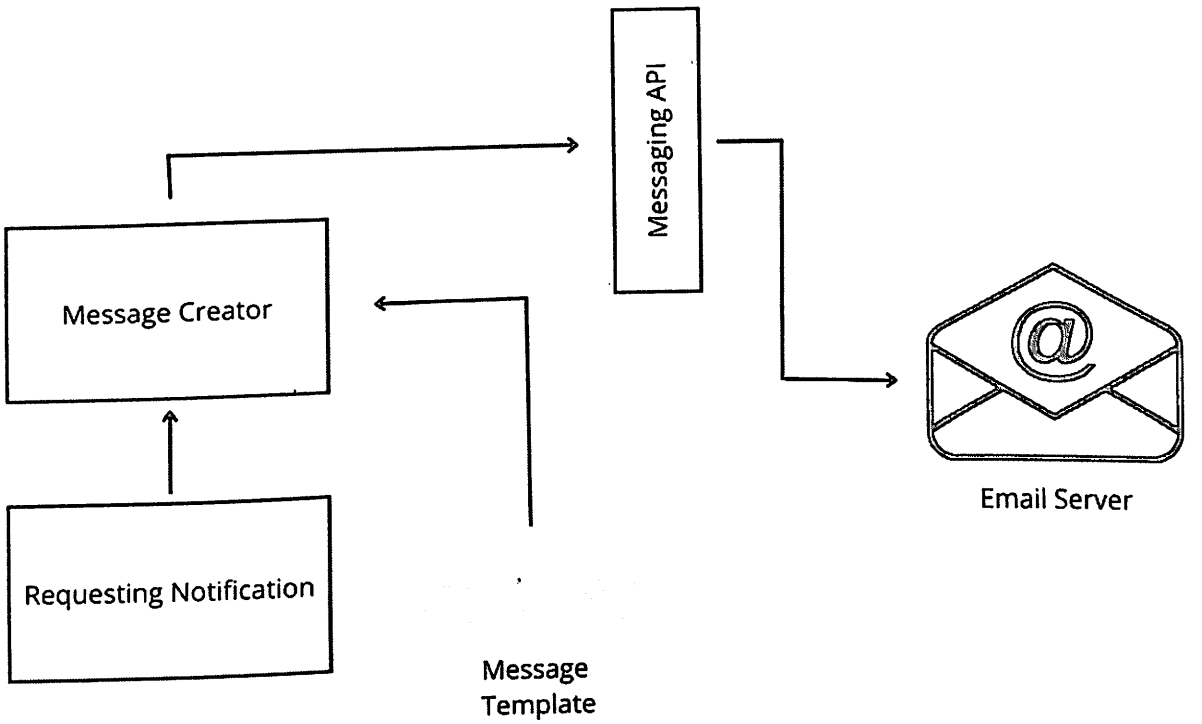


Figure 4.1: Architecture of Sending notifications

One of the advantages of using the SMTP method is that it provides a reliable and efficient means of transmitting email messages across different networks and devices. It ensures that email messages are delivered promptly and securely to the intended recipient. It also provides error checking and handling mechanisms to ensure that any errors or issues encountered during the transmission process are addressed in a timely manner.

Architecture of Sending notifications can be shown in Figure 4.1.

The picture appears to be a visual representation of a system architecture for sending notifications. This system architecture is designed to streamline the process of sending notifications by breaking it down into smaller, more manageable components. By separating the various tasks involved in notification delivery, this architecture allows for greater flexibility and scalability and can help to ensure that notifications are delivered in a timely and efficient manner. It features four components, including a message template data store, a messaging API, an email

server (SMTP server), and a message creator. The Message Template data store holds the raw message templates, which can be used to create customised messages for notifications. These templates can be easily modified or updated as needed, allowing for greater flexibility and customization in the messaging process. The messaging API is responsible for handling the delivery of notifications to their intended recipients. It serves as the bridge between the Message Template data store and the various channels through which notifications may be sent, such as email, SMS, or push notifications. The email server, or SMTP server, is specifically responsible for sending out email notifications. This server utilises the email protocol to deliver messages to recipients' email addresses. It is optimised for handling high volumes of email traffic and ensuring that messages are delivered in a timely and efficient manner. The last, the Message Creator, is responsible for loading the message template from the Message Template data store and creating the actual message that will be sent to the recipient. This component is essential for ensuring that notifications are created and sent in a consistent and accurate manner.

4.2 Cron Jobs

Cron could be a prevalent time-based work scheduler that's primarily utilised in Unix-like working frameworks such as Linux, BSD, and macOS. This open-source utility is broadly utilised by framework directors, designers, and control clients to computerise dreary assignments such as reinforcements, system maintenance, and information preparation. The Cron daemon runs within the foundation and permits clients to plan assignments at particular intervals, counting minutes, hours, days, weeks, and months. This implies that clients can arrange for Cron to run a specific command or script at a particular time or on a customary basis, depending on their necessities. Cron underpins a wide range of commands and scripts, which makes it a flexible instrument that can be redone to suit different needs. Python and R scripts can be planned as cron jobs to send mail notices at indicated intervals. This procedure can be used to mechanise e-mail notices from applications and workflows. It requires knowledge of Unix-based working frameworks and cron setup, as well as programming abilities.

Cron tasks require supervision. Consequently, it is critical to create software that keeps track of cron job execution. But this programme is not infallible. A cron job may send emails when it succeeds, but it is also possible for it to fail and prevent sending emails. Because users seldom check the latest update date, the sites will eventually become out-of-date, which users will realise when reading certain data. There is absolutely no version control. There is no straightforward method to determine who deleted a job and why. Additionally, adding more and more cron tasks to a single system over time may cause the machine to become overloaded.

4.3 APIs

A way to automate the process of sending emails using a programmable interface is by using APIs. Using APIs (application programming interfaces) is a powerful way to send notifications programmatically and in real-time [7]. Without having to create a separate email client or interface, it enables developers to include email functionality within their apps. APIs may be used to deliver a variety of emails, including newsletters, transactional emails, and marketing emails. Businesses that send a lot of emails on a daily basis might use this strategy to automate the process and save time [7]. Leveraging APIs enables businesses to measure and monitor email performance and make required modifications to ensure emails are not flagged as spam or banned by email filters, which helps businesses increase the deliverability of their emails. APIs allow developers to integrate their applications with third-party services and leverage their functionalities, including sending notifications via email. Many email services Providers offer APIs using services like Mailgun, SendGrid, and Amazon Simple Email Services. that can be used to send email notifications programmatically.

The process of developing an internal email solution or utilising a third-party email service is made easier and less expensive with the help of Amazon SES, an email service. For businesses who need to send emails from apps hosted on services like Amazon EC2, this service integrates smoothly with other AWS services, which is handy [19]. The fact that there are no discussions, minimum expenditure

restrictions, or long-term commitments makes Amazon SES flexible and affordable above all else. A free use tier is furthermore offered, and small extra costs are assessed based on the volume of emails sent and data transferred.

SendGrid is a powerful and flexible email delivery and management service that can help companies enhance the effectiveness of their email advertising operations and achieve better outcomes. SendGrid's email delivery infrastructure is optimised to ensure that messages are delivered to the recipient's inbox rather than being flagged as spam or landing in the junk folder. This helps increase the likelihood of email engagement and conversion rates.

Mailgun is a trustworthy organisation that provides email API services that allow you to send, confirm, and receive enormous volumes of emails using your domain [20]. It gives you access to a thorough tracking system that includes information on opens, clicks, bounces, and delivery, enabling you to keep track of the efficacy of your sent emails. Mailgun does not have a mobile app, while other platforms do, making them more attractive to businesses and individuals who are always on the go.

Python libraries such as Requests, urllib, and R packages such as httr, sendgridr, and aws.ses can be used to make API requests and send email notifications. This technique can be used to send email notifications from scientific applications that are hosted on cloud platforms.

Chapter 5

Results

The present section shows the findings of experimental evaluations that aimed to compare the effectiveness of the aforementioned methods.

To achieve this objective, our algorithms were executed on the Home Credit Bank dataset. Timestamps are critical for time-series analysis and modeling, as they provide insight into when events occurred and allow predictions to be made for future events. Data privacy is a crucial consideration when working with client data, and obtaining an appropriate dataset with a timestamp can be a challenging task. The dataset utilised in this study comprises a total of 307,511 observations and 122 features. As our data set's target value was not evenly distributed, we applied the SMOTE technique to balance it. The results of the synthetic minority over-sampling technique with more samples in the minority class are shown below in Figure 5.1. This improved the performance of a machine learning model by reducing bias towards the majority class and increasing the accuracy of predictions for the minority class. Among these features, 106 are numerical variables, while the remaining 16 are categorical variables. Numerical variables can be discrete or continuous and have values that can be expressed as numbers. Continuous numerical variables can take on any value within a range, but discrete numerical variables can only take on a countable set of values. Age, income, and score total are a few examples of numerical variables. Variables that accept values from a predetermined, constrained set of categories or groups are known as categorical variables. They are frequently expressed using labels or strings since they cannot

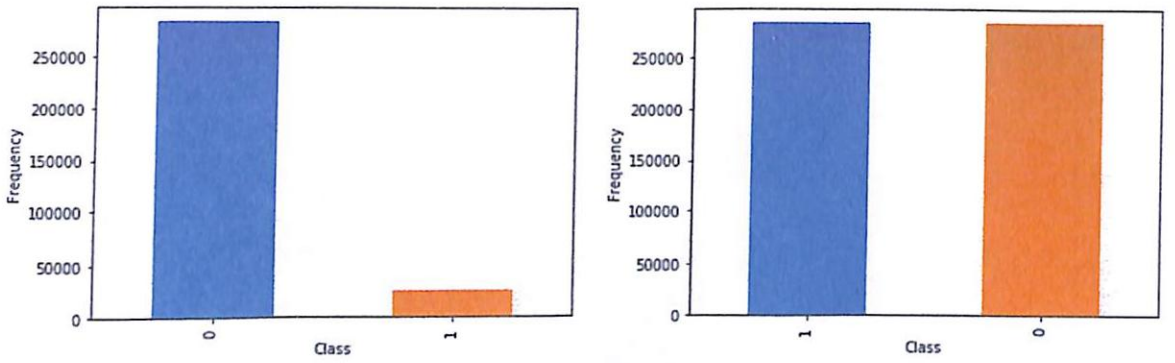


Figure 5.1: SMOTE technique

be quantified using numbers. The categorical variables gender, education, address, and city are a few examples. The key difference between numerical and categorical variables is that the former can be quantitatively measured while the latter cannot. Contrary to category variables, numerical variables may be added, subtracted, multiplied, and divided. For instance, adding "male" and "female" will never result in a result that makes sense, yet adding 5 and 7 would result in 12.

The visualisation shown in Figure 5.2 shows the distribution of different data types. The categorical variables were encoded using one-hot and label-encoding techniques. In this particular case, after encoding the dataset, the number of attributes increased, meaning that there are now twice as many attributes as there were before encoding. The quantity of characteristics decreased by a factor of 2 after applying the LGBM method. We can gain insights into which characteristics are most significant for generating predictions by utilising LGBM for feature importance analysis. The Figure below 5.3 displays the top 15 most important features as a consequence of this method. The results of the feature importance analysis show that certain characteristics are more important than others in predicting the target variable. We use this knowledge to improve the model by focusing on the most important features and deleting the least important ones to simplify it. Even though the correlation coefficient may not be the most efficient method to demonstrate the significance of a feature, it can still offer valuable information regarding potential connections within the data. By exploring the most prominent correlations, including the "Days of Birth" parameter that displays the highest positive correlation, we can gain important insights. Hence, it is essential to prioritise the analysis of this variable initially. To achieve this, we can create

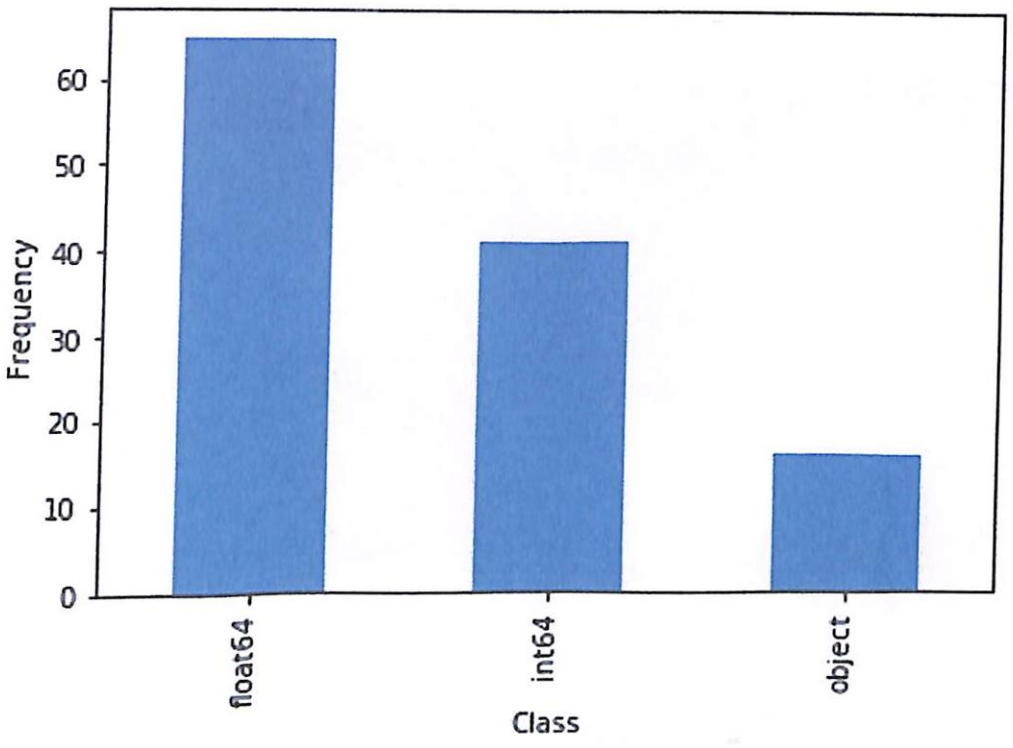


Figure 5.2: Distribution of data types of dataset

a histogram of the age, with the x-axis in years for clarity (Figure 5.4).

The age distribution alone does not reveal much, except for the absence of outliers, as all the ages appear to be reasonable. To gain a better understanding of how age impacts the target variable, we will create a kernel density estimation plot, which will be color-coded based on the target values. This plot is a smoothed histogram that displays the distribution of a single variable. We will utilize the seaborn kdeplot to generate a visualization that will show the distribution of ages based on target values (Figure 5.5). The curve representing the cases where the target equals one indicates a tendency towards younger age groups. This variable is probably useful in a machine learning model despite having a negligible correlation value of -0.07 since it affects the aim.
Let's examine that relationship from a different angle, as represented in Figure 5.6: average loan default rates by age group.

We then separated the age range into bins, with each bin including a 5-year gap, to make this graph. We then computed the average goal value for each of these bins, which gives us an idea of the proportion of unpaid debts in each age group.

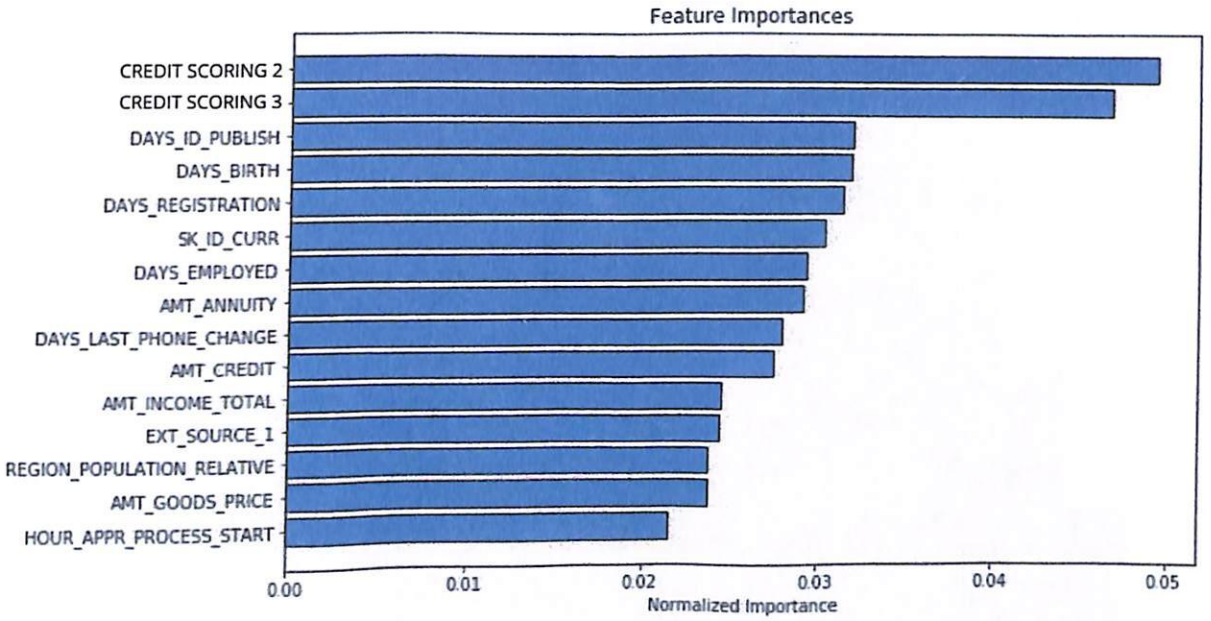


Figure 5.3: Feature importance

The likelihood of loan default is higher for younger individuals, according to the statistics, which show a definite trend. For the three youngest age groups, loan repayment failure rates are more than 10%, while for the oldest group, they are lower than 5%. The bank can immediately use this vital information. Although the bank shouldn't exclude younger customers, it could wish to offer additional advice or financial planning suggestions to help younger customers pay off their loans in a timely manner. Implementing precautionary measures like these is a prudent move.

Three variables — Credit_scoring_1, Credit_scoring_2, and Credit_scoring_3 — have strong negative correlations with the target variable. These parameters stand for a "normalised score from an external data source". In the beginning, we can demonstrate the correlation between credit-rating characteristics and the target variable as well as between them. All three credit scoring attributes have a negative correlation with the goal, indicating that the chance of the borrower repaying the loan rises as the Credit_scoring value rises. We may deduce that Credit_scoring_1 has a positive association with Days_Birth, indicating that one of the elements utilised to determine this score may be the client's age (Figure 5.7).

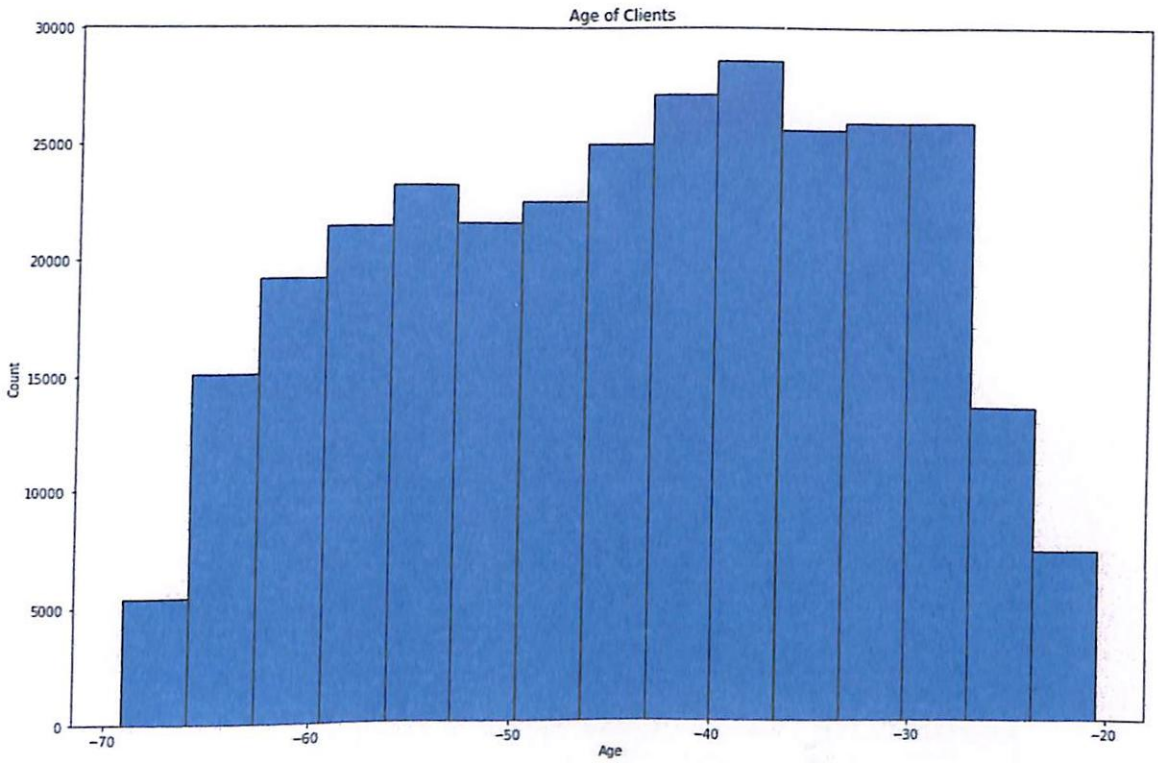


Figure 5.4: Age of Clients

We could proceed by examining the dispersion of each of these characteristics, with the added feature of colour-coding each dispersion according to the target's value. This approach would allow us to gain a visual understanding of how this variable affects the target (Figure 5.8, 5.9, 5.10).

We can summarise the exploratory analysis by producing a pair plot that displays the correlation between the Credit_scoring variables and Days_birth. The pair plot is a valuable method for examining the relationship between multiple pairs of variables and the distribution of individual variables. We are using the PairGrid function and the Seaborn visualisation package to build the Pairs Plot. The PairGrid function generated the diagonal histograms, the bottom triangle's 2D kernel density plots with correlation coefficients, and the top triangle's scatterplots. In this plot demonstrated in Figure 5.11, red denotes debts that are still owing money, whereas blue denotes loans that have already been paid. We can find various links between the variables by carefully scrutinising the visual data. Our research shows that Credit_scoring_1 and Days_Birth have a somewhat positive linear association, proving that this property takes the client's age into consideration.

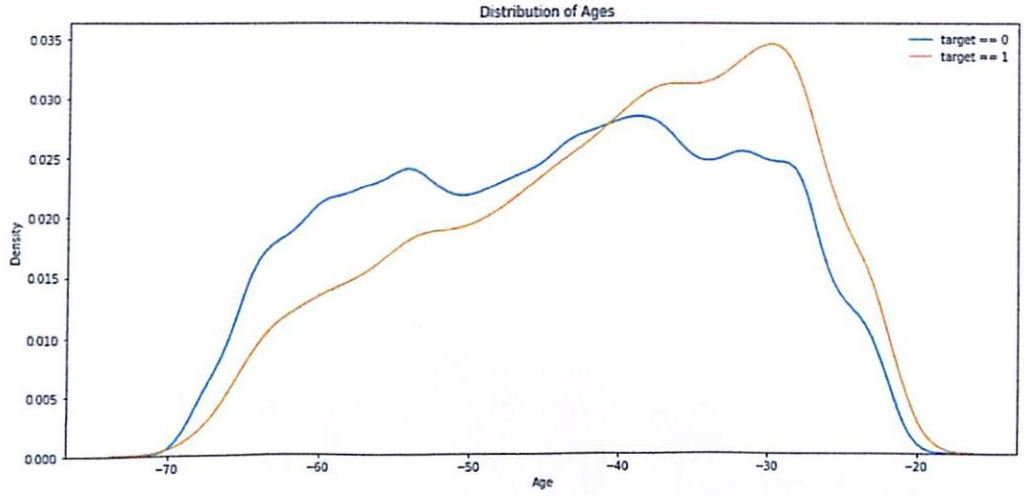


Figure 5.5: Distribution of Ages

The subsequent step is to calculate the approval level for every request.

$$Approval\ Rate = \frac{Number\ of\ Approved\ Applications}{Number\ of\ Total\ Applications}$$

At the outset, we need to group them by timestamp to find the approval level of each time.

Here's an algorithmic representation by step:

Step 1: Group the DataFrame *df* by the *Time_stamp* and Find the size of each group

Step 2: Filter *df* to only include rows where the *Target* column = 0

Step 3: Group the resulting DataFrame by the *Time_stamp* column and find the size of each group

Step 4: Calculate the approval rate for each *Time_stamp* group by dividing the size of the filtered DataFrame from step 2 to the size of the original DataFrame from step 1, multiplying the result by 100, and storing the resulting Series in the *app_rate* variable

The first step is to group the data in the 'df' data set based on timestamp intervals and determine the number of observations in each group. This information is stored in a variable called 'a'. Next, a new data frame is created where the 'target'

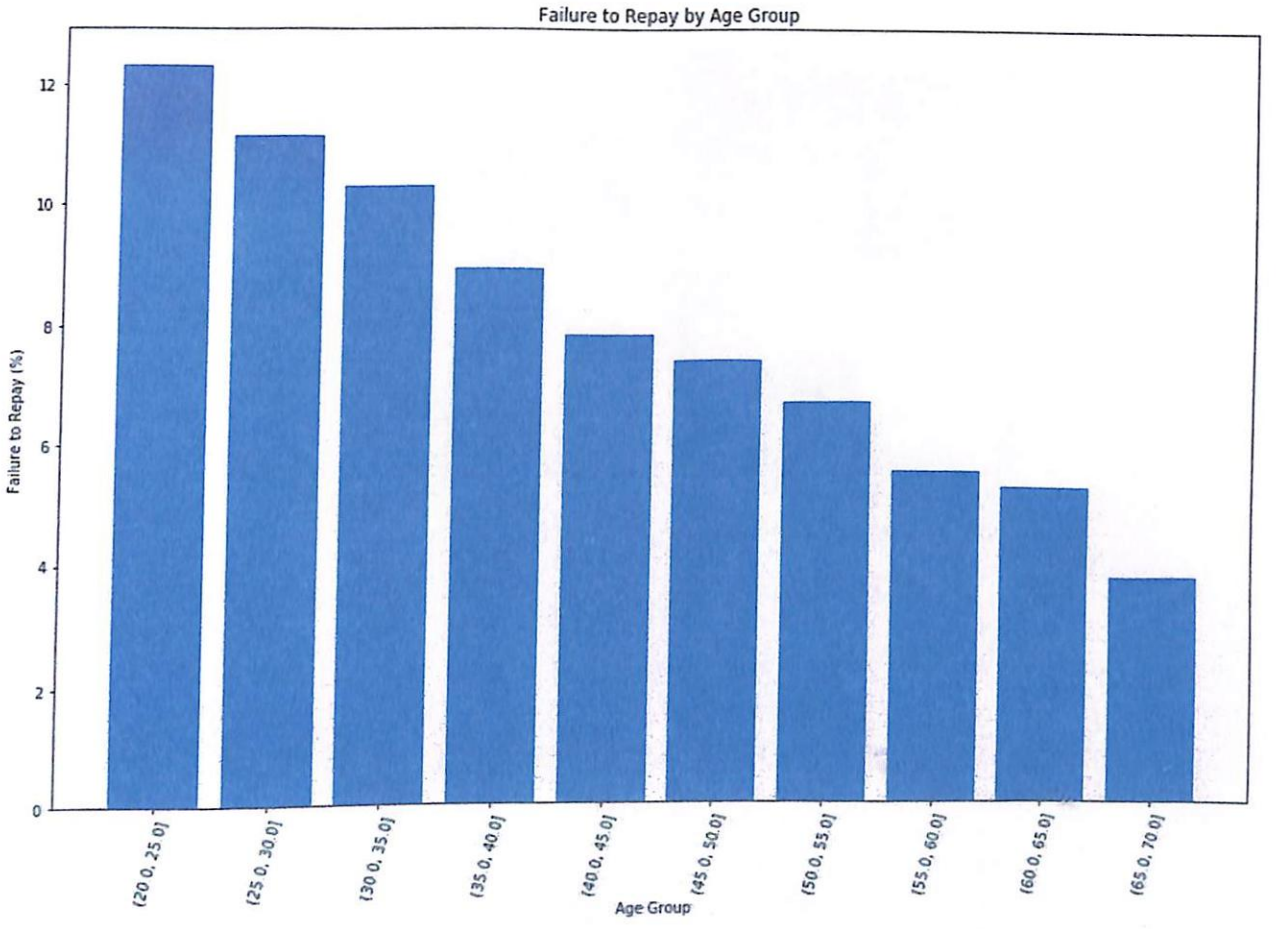


Figure 5.6: Age Group Failure to Repay

column is 0, indicating that no defaults have occurred. This data frame is also grouped by timestamps, and the size of each group is calculated. These results are stored in another variable. A new variable called 'app_rate' is then created by dividing the number of observations without defaults (stored in variable 'b') by the total number of observations (stored in variable 'a'). This value is then multiplied by 100 to get a percentage, and the data is grouped again by timestamp. Finally, a new column called 'app_rate' is created in the 'df' data frame. This column maps the values from the 'app_rate' variable to the 'time_stamp' column in 'df', allowing the 'app_rate' column to display the percentage of observations without default values for each timestamp.

Following exploratory data analysis and data preparation, we will employ machine learning algorithms on our obtained dataset. The current study investigated several outlier identification approaches utilised in the banking industry. The trials were carried out in order to determine the optimum possible configuration for

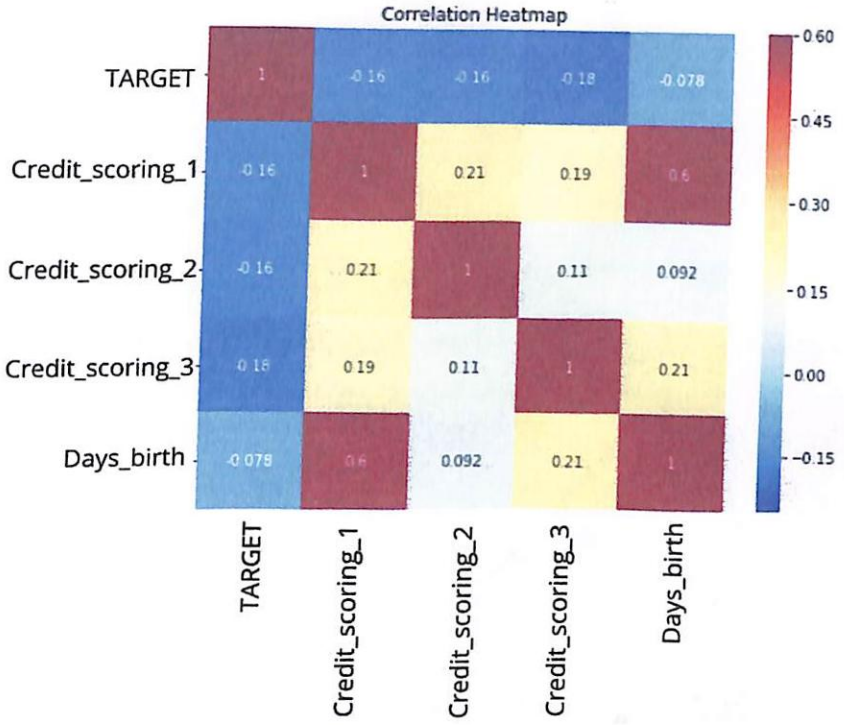


Figure 5.7: Correlation

each approach.

The study involved the global optimisation of all algorithms. Finding the hyperparameters that needed adjusting was the initial emphasis. The main hyperparameters, for instance, in the case of the isolation forest were the number of trees and the maximum sample size utilised to construct each tree. On the other hand, the primary hyperparameter for the local outlier factor was the number of neighbours used to generate the outlier score. A one-class support vector machine's key hyperparameters were the kernel function (kernel) and the regularisation parameter. Only a few of the strategies employed for hyper-parameter optimisation were grid search, random search, and Bayesian optimisation. Depending on the issue and search space, each approach offers advantages and downsides. Grid Search is easy to understand and use, does an exhaustive search over all possible hyperparameter combinations, and can be applied to any type of model. It cannot, however, handle continuous hyperparameters and is computationally expensive for large hyperparameter spaces. If the grid is not fine enough, it may also miss the optimum hyperparameters. We use it when the number of hyperparameters is

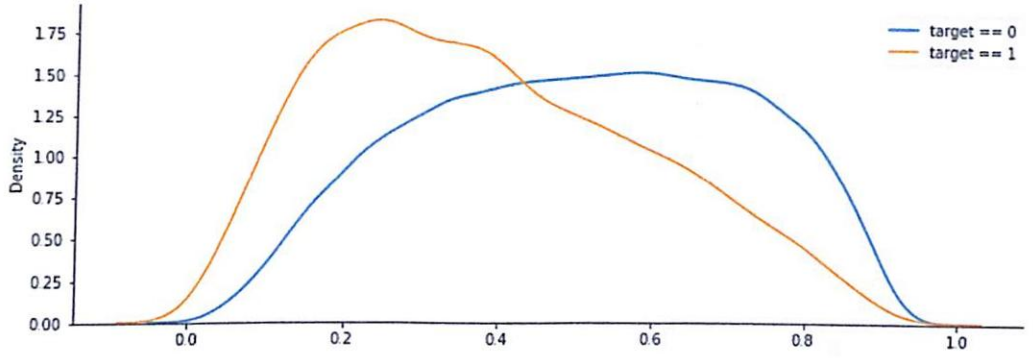


Figure 5.8: Distribution of credit scoring_1 by Target

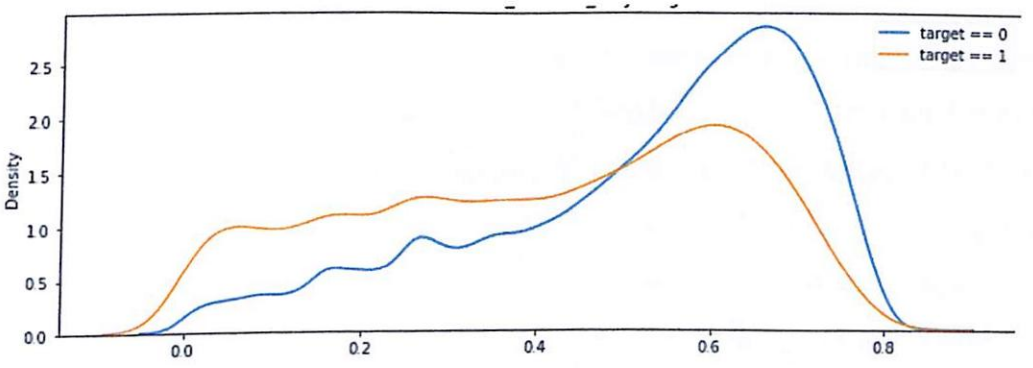


Figure 5.9: Distribution of credit scoring_2 by Target

small, the search space is small, and computational resources are not a constraint. Random search, like grid search, is a technique for hyperparameter optimisation. Random search, on the other hand, instead of painstakingly searching through all the hyperparameters in a preset grid, randomly samples hyperparameters from the search space. Exploration of huge hyperparameter spaces is computationally efficient. This approach is quicker than grid search in finding optimal hyperparameters and can handle continuous hyperparameters. Its key drawbacks are that randomization can sometimes produce suboptimal results and that it may not investigate all portions of the search space. Its key drawbacks are that randomization can sometimes produce suboptimal results and that it may not investigate all portions of the search space. It is crucial to highlight that while random search cannot always identify the global optimum, it may frequently find a good local optimum. Random search is best employed when there are a large number of hyperparameters, the search space is complicated, and computational resources are limited. By adjusting to past assessments, Bayesian optimisation efficiently searches the hyperparameter search space. It is capable of dealing with both continuous and discrete hyperparameters and has the benefit of discovering opti-

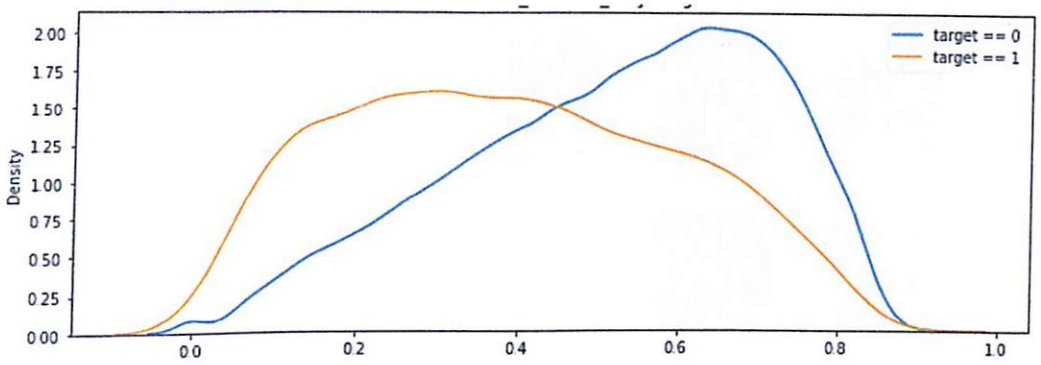


Figure 5.10: Distribution of credit scoring_3 by Target

mal hyperparameters with fewer evaluations. It does require previous knowledge of the search space and model, and for complicated models, it can be computationally expensive. Bayesian optimisation is most effective when the number of hyperparameters is large, the search space is complicated, computing resources are not limited, and prior knowledge about the search space and model is accessible. Bayesian optimisation constructs a probabilistic model of the goal function that depicts the relationship between the hyperparameters and the model's performance measure. Based on the projected gain in performance, this model is then used to recommend the next set of hyperparameters to assess. This enables Bayesian optimisation to concentrate on the most promising areas of the search space while avoiding those that are unlikely to provide excellent results.

The evaluation of the results was carried out utilising precision, recall, and F1 score metrics, as demonstrated in Table 5.1. The table provides the performance metrics (recall, precision, and F1-score) for three different anomaly detection algorithms applied to a specific dataset. The three algorithms are Isolation Forest, Local Outlier Factor, and One Class SVM. For each algorithm, the table shows its recall, precision, and F1-score. According to the table, the two top-performing algorithms in terms of F1-score are Isolation Forest (87.05%) and Local Outlier Factor (86.54%), with Isolation Forest having slightly higher scores in both precision and recall. The One Class SVM algorithm has a lower F1-score (60.01%) and lower recall (50.4%), indicating that it identifies fewer true anomalies and has a relatively high rate of false negatives. The interquartile range is a descriptive statistic used to assess a dataset's dispersion. Because it is not an algorithmic approach for detecting anomalies, it cannot be assessed using metrics like F1-score,

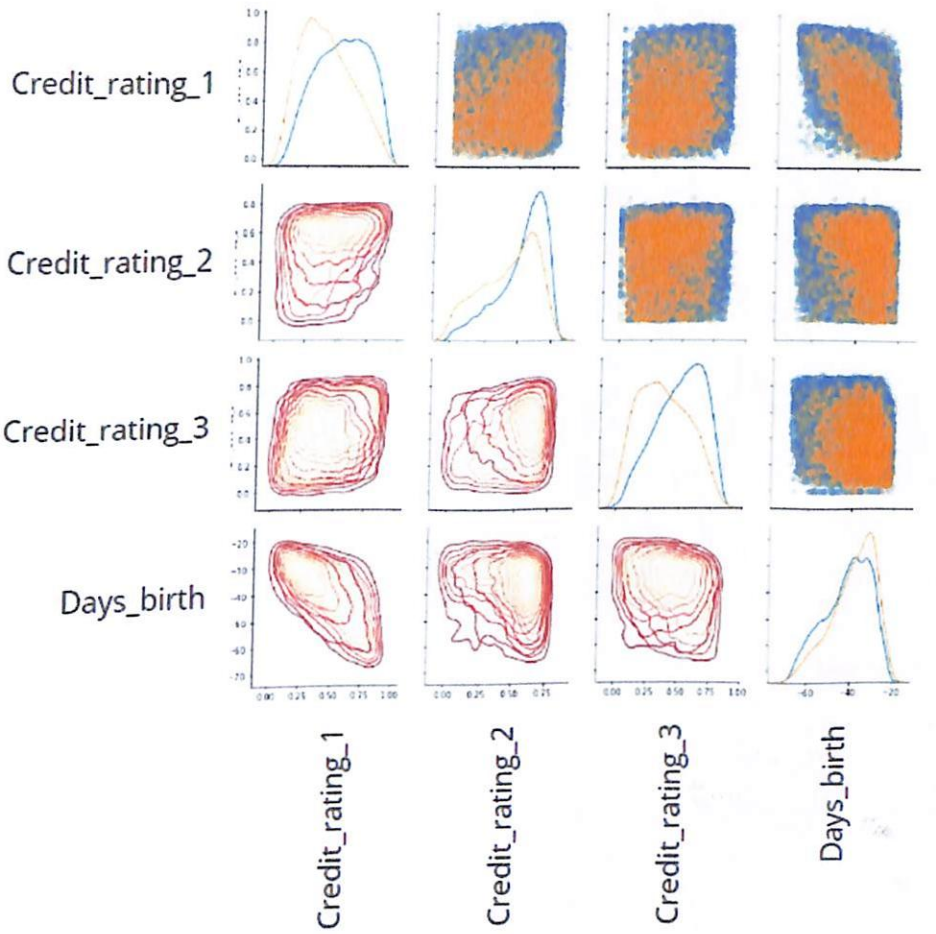


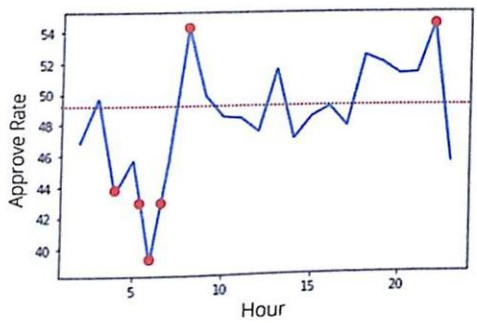
Figure 5.11: Pairs plot

recall, or accuracy. Instead, the IQR is a measure of variability derived from the difference between a dataset's third (Q3) and first (Q1) quartiles. The IQR can help you find values that are exceptionally high or low in comparison to the rest of the dataset. It is frequently used in conjunction with box plots to show data distribution and detect potential outliers.

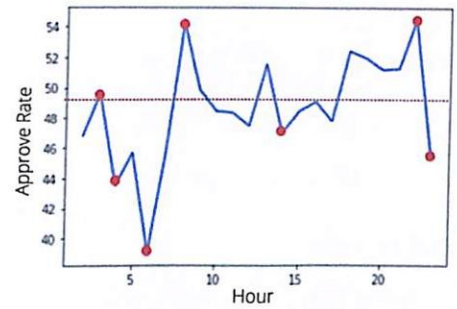
Visualisation showing the results of outlier detection algorithms such as one-class Support Vector Machine, Isolation Forest, Local Outlier Factor, and Interquartile Range are presented in Figure 5.12. Outlier detection methods for time series data can assist in identifying unexpected trends or abnormalities in the data over time. In this situation, the data being examined is the approval rate, which is depicted on the graph's y-axis. The x-axis depicts the time of day, while the blue line reflects the usual approval rate at that time. Dashed lines indicate a regular level of approval. The red dots on the graph indicate the outlier detection algorithms' observed abnormalities. These anomalies represent cases in

Table 5.1: Evaluation of Time Series Outlier Detection

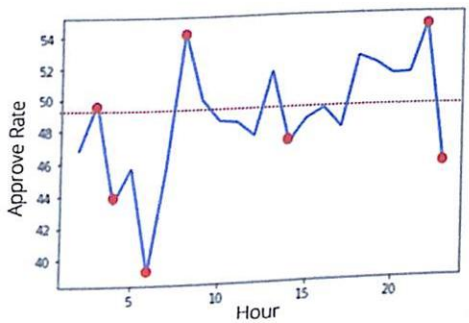
	Method	Recall	Precision	F1-Score
1	Local Outlier Factor	83.94%	89.2%	86.54%
2	Isolation Forest	84.06%	90.1%	87.05%
3	One Class SVM	50.4%	74.6%	60.01%



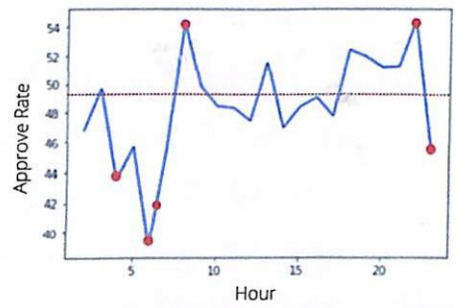
(a) IQR



(b) One-Class SVM



(c) LOF



(d) IF

Figure 5.12: Time Series Outlier detection

which the approval rate deviates considerably from the typical rate, which might be attributable to external influences influencing the approval rate. Line graphs were chosen because they are particularly useful for time-series data.

By detecting these outliers, we can investigate what caused them and take appropriate action to address the issue. Based on the results of these algorithms, Table 5.2 contrasts their benefits and drawbacks.

After conducting the experiments, we received valuable information about the problems that we are investigating. This led to new discoveries, insights and a deeper understanding of the subject under consideration. As a result of the experiments, we collected data, analyzed them and drew conclusions. Isolation Forest

Table 5.2: Comparison of Timer Series Outlier Detection Methods

	Methods	Advantages	Limitations
1	Isolation Forest	Low memory usage	Datasets with trends or seasonality
			Designed to handle datasets with complex structures
		Scalability	Not for skewed distribution
2	One-Class Support Vector Machines	Handle data with missing values	Computationally intensive
			Perform not well when the dataset has a large number of outliers
		Robust to noise	Sensitivity
3	Inter-Quartile range	Simple and easy to understand	Unable to handle missing values
		No affect of extreme values/outliers	It ignores some of the data
4	Local Outlier Factor	can be used with different distance metrics	Sensitive to the choice of parameters
		easy to implement	Computationally expensive
		can be used in high-dimensional datasets	Produce false positives if the dataset has clusters of similar data points that are not anomalous

is a machine learning approach designed specifically to handle datasets with intricate topologies. However, it may not do as well when it comes to discovering subtle correlations in time series data. The IQR approach, on the other hand, is used to find outliers in a dataset. It works by replicating the distribution of the data and identifying any values that fall outside of a predefined range of expected values. This strategy is especially useful when the data is dispersed frequently and the underlying distribution is well understood. Any result outside the interquartile range, which is 1.5 times the value below the lower quartile or above the upper quartile, is regarded as an outlier in the IQR technique. One potential drawback of the interquartile method is that it only considers the middle 50% of the data and ignores the extremes. This means that if a data set contains outliers or extreme values, the IQR may not be a suitable measure of variability since these values are not taken into consideration. Other measurements, such as the range or standard deviation, may be more appropriate in such instances. Another possible shortcoming of the interquartile approach is that it gives no information about the shape of the data distribution. Two data sets with the same IQR, for example, may have quite different forms: one with a symmetric distribution and the other with a skewed distribution. Other measurements, such as skewness or kurtosis, may be more instructive in such instances. The interquar-

tile range is a simple and reliable approach that may be used for datasets with simple distributions. SVM with only one class is computationally intensive. This method assumes that the dataset only includes one type of data and labels each observation that goes outside of that class as an anomaly. Our dataset is uneven in real-world circumstances, which means that the class of interest is underrepresented. As a result, one-class SVM did not perform well since it misclassified some normal data as anomalies. Isolation Forest and LOF are suitable for datasets with complicated forms and high-dimensional data, while One-class SVM is more suitable for datasets with non-linear separable data [8]. The local outlier factor is computationally demanding, and its performance degrades as the size of the dataset grows. This makes applying LOF to huge datasets in real-time applications difficult. Isolation Forest is a fast method that handles enormous datasets well. It operates by randomly choosing features and dividing points, making it less susceptible to outliers than other algorithms. It is a reliable and efficient method that may be applied to a wide range of anomaly detection activities. Its capacity to handle massive datasets, scalability, and flexibility made it a good fit for our situation.

Our next objective is to promptly inform the appropriate individuals of the unusual amounts of approval that we have discovered. Manually issuing such messages is an impossible task. This makes automating the message delivery procedure difficult. A better alternative is to send a message automatically to alert important stakeholders of the business indicator of concern. Several approaches were considered for this purpose, including sending notifications via an API, scheduling alerts with cron tasks, and using SMTP for message delivery. The outcomes of the comparison have been presented in tabular format, as shown in Table 5.3.

Comparing the results in the table 5.3, we can conclude that SMTP is suitable for large-scale email campaigns and industries with strict legal requirements because it is inexpensive and offers better data control. Cron jobs can run predefined scripts at predetermined intervals, but they require fewer resources and are more challenging to troubleshoot than other options. Real-time triggers and customizability are offered through APIs, but they also require technical know-how

Table 5.3: A comparison of sending notifications methods

Methods		Cost	Time	Reliability	Limitations	Advantages
SMTP		Free	<1 second	High	1. Authentication 2. Deliverability 3. Scalability	1. Immediate delivery 2. Aviability 3. Cost
Using cron jobs		Free	<1 second	Medium	1. Limited functionality 2. Dependence on the operating system 3. Debugging	1. Cost
Using APIs	Amazon SES	$24.95 per month	<1 second	High	1. Learning curve 2. Limited support 3. Cost	1. Scalability 2. Deliverability
	Mailgun	$35 per month			1. Learning curve 2. Limited support 3. Cost 4. Deliverability issues	1. Easy Integration 2. Detailed analytics 3. Advanced Features
	Send Grid	$14.95 per month			1. Learning curve 2. Limited support 3. Cost 4. Deliverability issues	1. Scalability 2. Advanced features

and may be expensive to use. The organization's specific objectives and resources dictate the strategy that is used.

As we indicated early in our research work about the implementation of a real-time alert-notification system for data monitoring in the financial industry [7], SMTP offers significant cost-saving advantages compared to other methods such as Mailgun, SendGrid, and Amazon SES API. We chose to use the SMTP method for delivering email notifications in our work after examining the various notification techniques that were accessible. Although this approach has certain drawbacks, such as a propensity for security problems and delivery failures, it is economical and gives users more control over their data, making it appropriate for massive email campaigns and sectors with stringent regulatory requirements. We can have complete control over our email data while avoiding paying for pricey email marketing firms or other third-party software by employing an SMTP server. This is crucial for regulatory compliance since it guarantees that we follow

Ответить Ответить всем Переслать Мгновенные сообщения

AA Аманкосова Аружан

RE: Urgent: Anomaly detected

Кому Аманкосова Аружан

The approval rate appears to be deviating from the norm.

This suggests a need to review and evaluate the effectiveness of our current strategy to ensure that it aligns with our business objectives

ID	TIME_STAMP	Approval_Rate	Is_Anomaly
1086	2023-01-26 20:15	49	0
1087	2023-01-26 20:30	48	0
1088	2023-01-26 20:45	51	0
1089	2023-01-26 21:00	50	0
1090	2023-01-26 21:15	47	0
1091	2023-01-26 21:30	46	0
1092	2023-01-26 21:45	22	1

See the attachment for more information.

Figure 5.13: Example of notification by email

stringent industry requirements. The choice to adopt SMTP was made after giving serious thought to the organisation's unique requirements and available resources. Additionally, we noted in a prior phase of our investigation that the Python programming language would be suitable for our situation. The SMTP functionality in Python may be implemented using the built-in module "smtplib". We may connect to an SMTP server and send emails by utilising that server thanks to this module. We can simply send emails from within your Python code without relying on any other libraries thanks to the "smtplib" package.

The steps of the process that involve monitoring and identifying any anomaly in the bank's approval rate and promptly communicating with the relevant parties to prevent any potential harm to the bank are shown in the appendix section 6.2.

The results of the conducted experiment are presented in Figure 5.13. The figure aids in successfully conveying the outcome by giving a visual representation, resulting in increased comprehension and more transparency.

The table 5.4 compares the time it took to manually send alert notifications to

employees versus the time it took to automatically send them. The names of the activities taken are included in the table, along with how long it took to finish each step. Creating the alert message, selecting the recipients, and sending the message by email or another channel are some of the stages involved in manually sending alerts to employees. According to the comparison table, these manual processes took an average of 30 minutes to complete, sometimes even up to 45 minutes. Automated alerts, on the other hand, can be set off by a software programme when specific criteria are satisfied. The table demonstrates how much quicker this procedure is, taking only 5 minutes on average for each notice. The automated procedure may be modified to give notifications to certain people or groups based on predefined criteria, requiring minimal input from human operators.

It turns out that the system for giving warnings in real time has been successfully implemented as a result of this effort. Previously, this required manual labour, and every time they checked the settings, it consumed time and distracted them. On occasion, people could disregard the significance of double-checking or carefully studying particulars. It implies that, compared to manual techniques, automating alarm notification might result in considerable time savings. Employees may respond to significant occurrences more swiftly by sending warnings faster, which can increase safety, productivity, and overall job satisfaction.

The study's findings demonstrated how well the models for real-time alert-notification systems worked at spotting outliers in datasets and sending timely warnings and alerts to the appropriate persons. The algorithms could identify outliers in the data in real time, allowing staff to respond quickly to any problems or abnormalities. The models were able to constantly learn from their mistakes and increase their precision over time, which increased their efficiency even further.

The investigation's findings suggest that implementing real time alert notification system models with the capacity to recognise outliers represents a significant development in the field of data analytics and has the potential to increase the precision and dependability of data-driven decision-making processes in a variety of fields and applications. To fully realise the promise of these models and discover fresh uses for them, more study in this area is required. Employees can therefore avoid doing tiresome duties due to this data monitoring. They will be

Table 5.4: A comparison of sending notifications methods

Working time		
Name of Procedures	Manually (in seconds)	Automatically (in seconds)
Read data from database	10	less than 1
Find the anomaly	30	less than 1
Create a report	200	less than 1
Open email	5	less than 1
Enter subject	15	less than 1
Mark recipients	120	less than 1
Write message body	200	less than 1
File attachment	15	less than 1
Send a report	5	less than 1

alerted automatically in the event of data errors.

Chapter 6

Conclusions and future work

6.1 Conclusions

This paper presents the implementation of real-time alerts for outliers in bank approval level indicators using the Python programming language. The present dissertation has been successful in providing an extensive amount of knowledge and insight towards automating banking sector monitoring processes. Thanks to our thorough investigation, we have successfully discovered a solution to our research question.

Initially, the hypothesis predicted that a static IQR method would work best because this identifier does not have such complex logic. However, the drawback of this approach was that some outliers went unnoticed, which negatively affected the way outliers were detected. Examining the outcomes of each approach allowed us to determine that the IF method was more effective in our particular instance. Other algorithms were less precise and required more time for computation. Several common evaluation methods for algorithms and a comparison of the advantages and disadvantages of each technique are used to give a comprehensive assessment. SMTP technique was selected for alert notification sending because it offers more benefits in our situation. Because their email infrastructure control improves data privacy and lowers the danger of sensitive information exposure. Additionally, it is cost-effective, especially in comparison to some third-

party email service providers. The price of third-party email service providers with extra features like analytics is expensive.

In conclusion, an automated system has been developed that can recognize errors in crucial financial metrics, particularly the approval level, and immediately alert the necessary stakeholders of these anomalies in real-time. With the help of this system, banks and financial institutions will be able to monitor important banking parameters efficiently and respond to problems as they develop. This technology represents a significant advancement in banking automation since it can quickly and precisely identify and warn employees of any deviations from expected norms in crucial metrics like approval rates.

It's no secret that an efficient automation system is critical to an organization's success. Employing dynamic solutions enriched with Python development tools allows banks to minimize resource allocation aimed at manual data processing procedures while optimizing work processes for enhanced productivity results. Customizing processes as per their reality trends enhances organizational operations while improving overall efficiency levels dramatically.

An organization's ability to leverage advanced technologies determines its competitiveness in today's dynamic banking sector, where quick-thinking is essential for growth. Machine learning algorithms and statistical methods have made it possible for systems to instantly identify problems, resulting in timely intervention measures being taken. The incorporation of customized ML applications and other automation tools using Python-based languages enhances accuracy further by enabling real-time analysis of patterns applied during decision-making processes, resulting in sustained success over time. Without this integration, significant opportunities could be lost, and needed optimization could also help reach newly improved heights by providing valuable insight with an edge over the competition through strategy implementation, giving banks the appropriate toolset necessary when operating in such times.

We believe this dissertation will give readers a clear understanding of the challenges associated with and approaches to automating monitoring procedures, as well as point them in the right direction for further study in this area.

6.2 Future work

Methodologies for detecting outliers have a lot of potential to improve in the future in various spheres of life. The use of these approaches in various spheres of life, such as health care, education, and public administration, among others, is one of the probable future options for the subject's development.

Anomaly discovery can be utilised in the healthcare sector for illness detection and diagnosis, locating broken medical equipment, and spotting phoney insurance claims. For instance, Isolation Forest may be used to spot irregularities in X-rays and MRI scans to identify potential health risks. Finding irregularities can help healthcare professionals identify unexpected patient data patterns, such as abrupt changes in vital signs, and notify staff about patients who need immediate treatment. Strange occurrences can be used to spot pupils who need more attention or who are in danger of leaving the education sector. It may be used, for instance, to keep tabs on students who suddenly change their attendance patterns or academic performance and need further academic help.The capacity to see anomalies may be used to spot fraud and corruption in the public administration sector by spotting unusual patterns in publicly available data, such as tax records.

The potential uses for anomaly detection techniques in different spheres of life are numerous and diverse. These approaches will probably be utilised even more frequently in the future as they develop and get better, contributing to improvement.

Bibliography

[1] Robert Poenaru. Implementation of an email-based alert system for large-scale system resources. 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), pages 1–6, 2021. doi: 10.1109/RoEduNet54112.2021.9638277. URL https://ieeexplore.ieee.org/document/9638277.

[2] Mohammad Braei and Sebastian Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art. arXiv preprint arXiv:2004.00433, 04 2020. URL https://arxiv.org/pdf/2004.00433.pdf.

[3] Neville Nicholls, Carol Skinner, Margaret Loughnan, and Nigel Tapper. A simple heat alert system for melbourne, australia. International journal of biometeorology, 52:375–84, 06 2008. doi: 10.1007/s00484-007-0132-5. URL https://www.researchgate.net/publication/5790370_A_simple_heat_alert_system_for_Melbourne_Australia.

[4] Hugo Jair Escalante. A comparison of outlier detection algorithms for machine learning. Programming and Computer Software, 01 2005. URL https://www.researchgate.net/publication/228728521_A_comparison_of_outlier_detection_algorithms_for_machine_learning.

[5] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. Outlier detection using isolation forest and local outlier factor. RACS '19: Proceedings of the Conference on Research in Adaptive and Convergent Systems, pages 161–168, 09 2019. doi: 10.1145/3338840.3355641. URL https://www.researchgate.net/publication/337204025_Outlier_

detection_using_isolation_forest_and_local_outlier_factor#:~:text=As%20well%2Dknown%20outlier%20detection,it%20has%20high%20time%20complexity.

[6] Siti Khatijah Nor Abdul Rahim, Ismail Nor Rashidah Paujah, Fadzilah Abdol Razak, I. Zulkifli, Nurul Jamian, N. Razi, and Nor Mohammad. Automated attendance management and alert system. Journal of Fundamental and Applied Sciences, 9:59, 02 2018. doi: 10.4314/jfas.v9i6s.6. URL https://www.ajol.info/index.php/jfas/article/view/165540.

[7] Aruzhan Amankossova and Cemil Turan. Implementation of a real-time alert-notification system for data monitoring in the financial industry. Suleyman Demirel University Bulletin: Natural and Technical Sciences, 62(1):132–141, 2023. ISSN 2709-2631. doi: 10.47344/sdubnts.v62i1.891. URL https://journals.sdu.edu.kz/index.php/nts/article/view/891.

[8] Aruzhan Amankossova and Cemil Turan. An evaluation of unsupervised outlier detection methods for univariate time series data in financial transactions. Suleyman Demirel University Bulletin: Natural and Technical Sciences, 62 (1):178–188, 2023. ISSN 2709-2631. doi: 10.47344/sdubnts.v62i1.911. URL https://journals.sdu.edu.kz/index.php/nts/article/view/911.

[9] Juanjuan Wang, Mantao Xu, Hui Wang, and Jiwu Zhang. Classification of imbalanced data by using the smote algorithm and locally linear embedding. In 2006 8th international Conference on Signal Processing, volume 3. IEEE, 2006. URL https://ieeexplore.ieee.org/document/4129201.

[10] Brendan Juba and Hai S. Le. Precision-recall versus accuracy and the role of large data sets. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01):4039–4048, Jul. 2019. doi: 10.1609/aaai.v33i01.33014039. URL https://ojs.aaai.org/index.php/AAAI/article/view/5193.

[11] Schweser. 2020 CFA Level II Schweser Notes eBook 1. Kaplan, 2020. ISBN 9781475495515. URL https://dokumen.pub/qdownload/2020-cfa-level-ii-schweser-notes-ebook-1-9781475495515.html.

[12] Boddu L V Siva Rama Krishna, V Mahalakshmi, and Gopala Krishna Murthy

Nookala. A comprehensive analysis on outlier prediction using learning approaches. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pages 566–573, 2022. doi: 10.1109/ICAAIC53929.2022.9792877. URL https://ieeexplore.ieee.org/document/9792877.

[13] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. A review of local outlier factor algorithms for outlier detection in big data streams. Big Data and Cognitive Computing, 5(1), 2021. ISSN 2504-2289. doi: 10.3390/bdcc5010001. URL https://www.mdpi.com/2504-2289/5/1/1.

[14] Mahsa Salehi, Christopher Leckie, James C. Bezdek, Tharshan Vaithianathan, and Xuyun Zhang. Fast memory efficient local outlier detection in data streams. IEEE Transactions on Knowledge and Data Engineering, 28(12):3246–3260, 2016. doi: 10.1109/TKDE.2016.2597833. URL https://www.researchgate.net/publication/305825504_Fast_Memory_Efficient_Local_Outlier_Detection_in_Data_Streams.

[15] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. IEEE Transactions on Knowledge and Data Engineering, PP:1–14, 04 2023. doi: 10.1109/TKDE.2023.3270293. URL https://arxiv.org/pdf/2206.06602.pdf.

[16] Lars Doorenbos, Stefano Cavuoti, Massimo Brescia, Antonio DIsanto, and Giuseppe Longo. Comparison of Outlier Detection Methods on Astronomical Image Data, pages 197–223. Springer International Publishing, 2021. ISBN 978-3-030-65867-0. doi: 10.1007/978-3-030-65867-0_9. URL https://doi.org/10.1007/978-3-030-65867-0_9.

[17] Shabbir Ahmad, Zhengyan Lin, Saddam Akber Abbasi, Muhammad Riaz, et al. On efficient monitoring of process dispersion using interquartile range. Open journal of applied sciences, 2(04):39–43, 2012. URL https://www.scirp.org/journal/paperinformation.aspx?paperid=26464.

[18] Vladimir Riabov. SMTP (Simple Mail Transfer Protocol), pages 388–406. John Wiley , Sons, 12 2007. ISBN 978-0-471-78459-3. doi:

10.1002/9781118256114.ch26. URL https://www.researchgate.net/publication/273830243_SMTP_Simple_Mail_Transfer_Protocol.

[19] Sajee Mathew Jinesh Varia. Amazon Web Services – Overview of Amazon Web Services. Advanced Software Engineering at the Department of Engineering of Roma Tre University, 01 January 2014. URL http://cabibbo.dia.uniroma3.it/asw-2014-2015/altrui/AWS_Overview.pdf.

[20] Tony Phillips. Asset management software. Technical Library, June 2017. URL https://scholarworks.gvsu.edu/cistechlib/274/.

Appendix A

Steps of the methodology for detecting an anomaly of approval rate in the bank and notifying the responsible persons in the event of an abnormality to avoid possible risky circumstances:

1. Import the necessary python libraries
   - from email.message import EmailMessage
   - import smtplib
   - from sklearn.ensemble import IsolationForest
2. Define the sender and recipient email addresses as strings
3. Create an 'EmailMessage' instance and set the sender, recipient, subject, and content of the email.
4. Set up an SMTP server (in this case, Outlook) and login to the sender's email account using password.
5. Read the data from database
6. Group by timestamp 15 minutes
7. Calculate the approval rate by timestamp
8. Identify outliers by using IF algorithm
9. Save the result of the algorithm in another table containing the columns:
   1) id
   2) time_stamp
   3) approval_rate
   4) is_anomaly
10. Automatically check if there is a record in the table with the value of one column is_anomaly

SQL code: select * from table_research
where is_anomaly = 1

11. If is_anomaly = 1, then send an email alert.
Send the email using the SMTP server and the 'sendmail()' function to pass the sender email, recipient email, and email message as arguments.
- 'smtp.sendmail(sender, recipient, email.as_string())'

12. Close the SMTP connection using the 'quit()' function.
- 'smtp.quit()'

Appendix B

Function to calculate missing values by column:

```
def missing_values_table(df):
    # Total missing values
    mis_val = df.isnull().sum()
    # Percentage of missing values
    mis_val_percent = 100 * df.isnull().sum() / len(df)
    # Make a table with the results
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
    # Rename the columns
    mis_val_table_ren_columns = mis_val_table.rename(
    columns = {0 : 'Missing Values', 1 : '% of Total Values'})
    # Sort the table by percentage of missing descending
    mis_val_table_ren_columns = mis_val_table_ren_columns[
        mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
    '% of Total Values', ascending=False).round(1)
    # Print some summary information
    print ("Your selected dataframe has " + str(df.shape[1]) +
    " columns.There are "+ "str(mis_val_table_ren_columns.shape[0]) +
          " columns that have missing values.")
    return mis_val_table_ren_columns

# Missing values statistics
missing_values = missing_values_table(app_train)
missing_values.head(20)
```

Code of Label Encoder to convert categorical variables into numerical format.

```
# Create a label encoder object
le = LabelEncoder()
le_count = 0

# Iterate through the columns
for col in app_train:
    if app_train[col].dtype == 'object':
        # If 2 or fewer unique categories
        if len(list(app_train[col].unique())) <= 2:
            # Train on the training data
            le.fit(app_train[col])
            # Transform both training and testing data
            app_train[col] = le.transform(app_train[col])
            app_test[col] = le.transform(app_test[col])
            # Keep track of how many columns were label encoded
            le_count += 1
# one-hot encoding of categorical variables
app_train = pd.get_dummies(app_train)
app_test = pd.get_dummies(app_test)
```