

University of Khartoum



Faculty of Mathematical Sciences and  
Informatics

Department of Statistics

## Data Pre-processing Methods: Review and illustration

### Group Members:

Mohammed Salah Aldeen Ahmed	14-206
Mohammed Kamal Shams Aldeen	14-207
Mohammed Almeeraf Nazar	14-208

### Supervisor:

Mr. Raad Shaaban

## الخلاصة:

يهدف هذا المشروع إلى التحقيق في تقنيات تحضير البيانات الخام للتحليل، مع التركيز على أهمية معالجة البيانات الأولية في مجالات مثل استخراج البيانات والتعلم الآلي. يناقش المشروع مختلف طرق المعالجة الأولية مثل تنظيف البيانات، التحويل، الاختزال، والتمثيل التكراري. سيتم فحص كل تقنية من حيث هدفها، ومزاياها، وعيوبها المحتملة، وغالبًا ما تكون مصحوبة بأمثلة حقيقية أو دراسات حالة. علاوة على ذلك، يتضمن المشروع التنفيذ العملي ومقارنة هذه التقنيات على مجموعات البيانات لتقييم تأثيراتها على التحليل أو أداء النموذج.

## Abstract:

This project aims to investigate techniques for preparing raw data for analysis, emphasizing the importance of data preprocessing in fields like data mining and machine learning. It discusses various preprocessing methods such as Data

Cleaning, Transformation, Reduction, and Discretization. Each technique will be examined in terms of its purpose, advantages, and potential drawbacks, often accompanied by real-world examples or case studies. Furthermore, it involves practical implementation and comparison of these techniques on datasets to evaluate their effects on analysis or model performance.

## Acknowledgment:

We sincerely thank Mr. Raad Shaaban for his invaluable guidance and support throughout this project. We are also grateful for our Friends for their support. We deeply appreciate the unwavering support and encouragement from our families throughout this process.

## Table of Content:

Title	Page
Abstract	1
Acknowledgment	1
Table of Content	2
List of Tables and Figures	4
<b>Introduction</b>	5
introduction	5
Background	5
<b>1.Proposal</b>	7
1.1.Objectives	7
1.2.Scope of Work	7
1.3.Implementation Plan	8
1.4.Expected Outcome	8
1.5.Conclusion	8
<b>2.Literature Review</b>	9
2.1.Introduction	9
2.2. Outlier Detection and Handling	10
2.3.Overview of Data Pre-processing Techniques	12
2.3.1.Data Cleaning	12
2.3.2.Handling Missing Value Material and Technique	14
2.4.Current approaches used by practitioners	17
2.5.Conclusion	22
<b>3.Methodology</b>	23
3.1 introduction	23
3.2 Research Design	23
3.3 Research Scope	23
3.4 The Data overview	25
3.5 Data preprocessing steps	26
3.6 Real Case Studies for Preprocessing and Model Performance Evaluation	33

<b>4. Implementation</b>	36
4.1 Introduction	36
4.2 Some Important Definitions	36
4.3. Case Study 1	38
4.4 Case Study 2	45
4.5. Case Study 3	47
4.6. Case Study 4	51
Conclusion and Recommendations	69
<b>Bibliography</b>	70

## List of Tables and Figures

<b>Table / Figure</b>	<b>Page</b>
Table of Content	2
<b>Figure1:</b> Research on Data Preprocessing	12
Figure2: Data Cleaning	12
Figure3: Data Transformation	13
Figure4: Data Balancing	13
Figure 5: Preprocessing on ML	26
Figure 6: Powerful ML Model	27
Figure 7: Data Cleaning Steps	28
Figure 8: Feature Engineering	30
Figure 9: Statistical vs ML Technique	39
Figure 10: MVI Methods	39
Figure 11: MVI Metrics	40
Figure 12: Missing Value Imputation	43
Figure 13: ML Performance Improvements	44
Figure 14: (MLP) prediction model	41
Figure 15: SVD Prediction quality	48
Figure 16: SVD prediction generator	48
Figure 17: Top 10 (movie)	49
Figure 18: Top 10 (commerce)	49
Figure 19: Pearson correlation	53
Figure 20: Oversampling summary	53
Figure 21: Performance plot (NearMiss)	54
Figure 22: Performance plot(Smote)	55
Figure 23: Random oversampling technique.	55
Figure 24: classification(undersampling)	56
Figure 25: classification(oversampling)	56
Figure 26: IQR	60
Figure 27: distribution After using IQR	61
Figure 28: Standard Deviations	62
Figure 29: distribution After using SD	63
Figure 30: z-Score Method	64
Figure 31: distribution After z-score	65

## **Introduction**

### **Introduction:**

In recent years, the Web has transitioned from a focus on sharing and broadcasting to emphasizing individual data tracking and analysis. Various applications now enable people to monitor their habits, behaviors, and surroundings, providing insights into personal aspects such as eating habits, exercise, online activity, and finances.

Historically, personal data collection, like the efforts of the 1930s British social research group Mass Observation, relied on manual methods. Today, advanced technologies allow for automatic, real-time data collection via mobile phones and handheld devices, resulting in vast and continuous data streams.

The challenge lies not in the sheer volume of data but in making this data comprehensible and meaningful to individuals without statistical or computational expertise. This underscores the need for efficient and accurate data processing systems that transform raw data into clear, understandable information.

Data preprocessing is a critical step in data analysis and machine learning. It involves transforming raw data into a format that is suitable for analysis. Effective preprocessing ensures that the data is clean, consistent, and ready for modeling.

This project aims to review and illustrate various data preprocessing methodologies, highlighting their importance in preparing data for analysis and interpretation. By examining existing methodologies and showcasing their practical applications, the project will address the current demands for effective data preprocessing solutions that cater to non-professionals.

### **Background:**

Before any research or project activity can commence, the analyst or researcher must collect and prepare the data. The data available is often not in the ideal form for use and may require significant operations before processing. Our project focuses on this crucial stage of research.

Specifically, we discuss the properties of high-quality data, common issues that datasets can present, and methods for data preparation.

Key questions to address before collecting data include: Is the data (accessible, sizeable, usable, understandable, and reliable?)

By exploring this question, our project aims to provide a comprehensive review of data preprocessing methodologies and illustrate their importance in ensuring that data is ready for analysis.

## **Chapter 1: Proposal**

### **Objectives:**

#### **1\ Review:**

To conduct a comprehensive review of current data preprocessing methodologies, highlighting their strengths, weaknesses, and areas of application.

#### **2\ Identify Best Practices:**

To identify and document best practices in data preprocessing to guide practitioners in choosing the most effective methods for various types of data.

#### **3\ Illustration:**

To demonstrate the differences after application of these techniques.

### **Scope of Work:**

#### **1\ Literature Review:**

Conduct a comprehensive review of existing literature on data preprocessing methods. This will include techniques such as:

- Handling missing data
- Data normalization and scaling
- Handling outliers
- Data transformation

#### **2\ Analysis of Data Quality Issues:**

Analyze typical data quality issues that necessitate preprocessing. Such as missing values, outliers, noise, and Inconsistencies.

Provide examples of datasets with these issues. Explain how these issues impact data analysis and decision-making processes.

#### **3\ Case Study:**

Select a real-world dataset from a specific domain (healthcare, finance, social media ...etc.) To illustrate how preprocessing technique can be applied:

- Describe the dataset and its characteristics.
- Discuss the impact of each technique on the data quality and analysis outcomes.



#### **4\ Documentation and Presentation:**

The report includes detailed chapters on each aspect of the project, including methodology, analysis and case studies. Ensure that all illustrations, tables, and charts are clear and informative.

#### **Implementation Plan:**

- Data Collection: Gather diverse datasets from publicly available sources or use standard datasets from machine learning libraries.
- Evaluation: Evaluate the effectiveness of each preprocessing technique based on:
  - Improved data quality metrics (reduced error rates, enhanced model performance)
  - Comparison of results before and after preprocessing.

#### **Expected Outcome:**

**A comprehensive review project summarizing various data preprocessing methods.**

- A guide with illustrations of Differences occurred after the techniques applied to the real-world dataset.
- Glance of data preprocessing in improving data analysis and machine learning outcomes.

#### **Conclusion:**

Data preprocessing is crucial for ensuring the quality and reliability of data analysis. By reviewing and illustrating various methods, this project aims to provide a clear understanding of how different techniques can be applied to prepare data effectively for analysis and modeling.

This proposal outlines a structured approach to explore and demonstrate key concepts in data preprocessing, offering both theoretical insights and practical comparison.

## **Chapter 2: Literature Review**

### **Introduction:**

In any data driven discipline, the quality of the data is fundamental to the accuracy and reliability of the models or analyses that rely upon it.

Data preprocessing is a critical step in any data analysis data science or machine learning project.

However, this phase presents numerous challenges due to the inherent imperfections of the real-world datasets, which are often incomplete, inconsistent, imbalance, and noisy. AS such, effective data preprocessing is essential to ensuring that data is well suited for model performances.

Data preprocessing involves cleaning, transforming, and organizing raw data to make it suitable for analysis. The methodologies for data preprocessing are diverse, encompassing techniques for handling missing data, normalizing and scaling features, data quality, detecting and addressing outliers, and transforming data into formats that better suit analytical models. This literature review draws on many famous research papers and books, highlighting old methodologies and practices with their flaws and short comings and last cutting-edge methodologies and how many combinations of these methods could be the key to improvement and finding better solutions to this important topic.

Fundamental Problems in Data Preprocessing and Their Definitions:

### **Handling Missing Values:**

Handling missing values refers to the process of identifying, addressing, and treating gaps in a dataset where certain observations or features lack values. These missing values can result from data collection errors, equipment malfunction, or the absence of information.

The techniques for handling missing data include deletion methods (removing rows or columns with missing data), imputation methods (replacing missing values with substituted values like the mean, median, or a predicted value), and advanced methods such as multiple imputation and K- nearest neighbors (KNN) imputation.

The goal is to minimize bias and retain the dataset's integrity while ensuring that analyses or models built on the data remain robust [1].

### **Outlier Detection and Handling:**

Outliers are data points that deviate significantly from other observations in the dataset. These points can skew statistical results and negatively affect machine learning models, especially in sensitive algorithms like linear regression and k-nearest neighbors. Outlier detection involves identifying these anomalous points using techniques such as Z-scores, interquartile range (IQR), or distance-based methods (e.g., K-nearest neighbors or clustering). Once identified, outliers can either be removed, transformed, or retained depending on their cause and impact on the analysis [1].

### **Data Normalization and Standardization:**

Normalization and standardization are techniques used to scale the features of a dataset, so they are within a similar range, which is particularly important for algorithms that are sensitive to the magnitude of feature values.

Normalization typically rescales data into a range of [0, 1] using Min-Max scaling, while standardization transforms data so that it has a mean of 0 and a standard deviation of 1 (Z-score standardization). These methods help prevent certain features from dominating the model due to their larger scale [1].

### **Noise Removal:**

Noise removal is the process of smoothing noisy data through methods such as binning or applying statistical filters to reduce variability that may obscure underlying patterns [1].

Data Transformation:

Data transformation involves modifying the original data to make it suitable for analysis or modelling. Common transformations include:

- One-hot encoding:** Converting categorical variables into numerical formats (get\_dummies in Pandas).
- Logarithmic transformations** for dealing with skewed distributions.
- Polynomial transformations** to capture non-linear relationships. Transforming data helps ensure that machine learning algorithms can interpret the data correctly and efficiently [1].

### **Data Imbalance:**

Data imbalance refers to situations where the distribution of classes within a dataset is skewed, with one class being much

more prevalent than others. This can lead to biased model predictions, favoring the majority class. Common techniques for addressing imbalance include over-sampling the minority class, under-sampling the majority class, or generating synthetic data using techniques like SMOTE (Synthetic Minority Over- sampling Technique) [1].

### **Data Reduction:**

Data reduction is the transformation of numerical or alphabetical digital Data derived empirically or experimentally into a corrected, ordered, and simplified form. The purpose of data reduction is to reduce the number of data records by eliminating invalid data or produce summary data and statistics at different aggregation levels for various applications [1]. Data reduction does not necessarily mean loss of information. Data reduction has three types: Dimensionality reduction, Numerosity reduction and Statistical modeling [1].

### **Data Discretization:**

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss.

## Overview of Data Preprocessing Techniques:

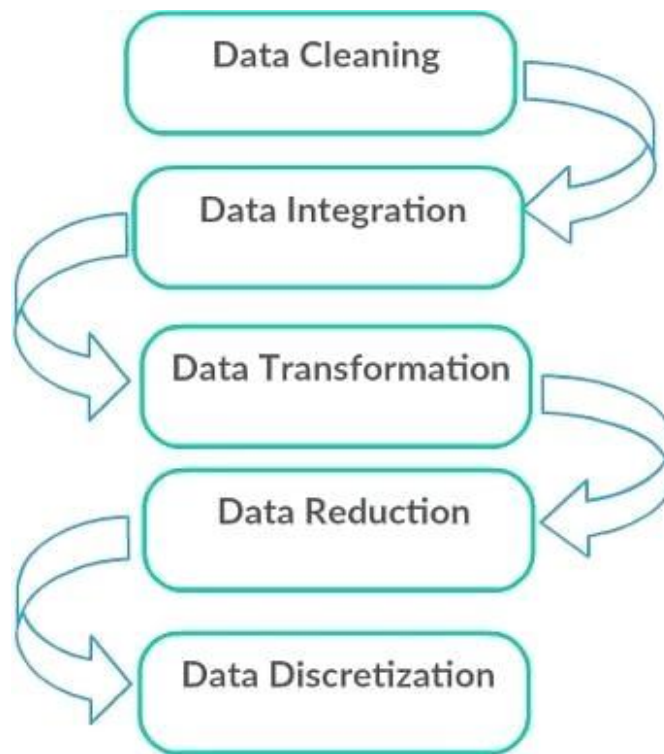


Figure 1

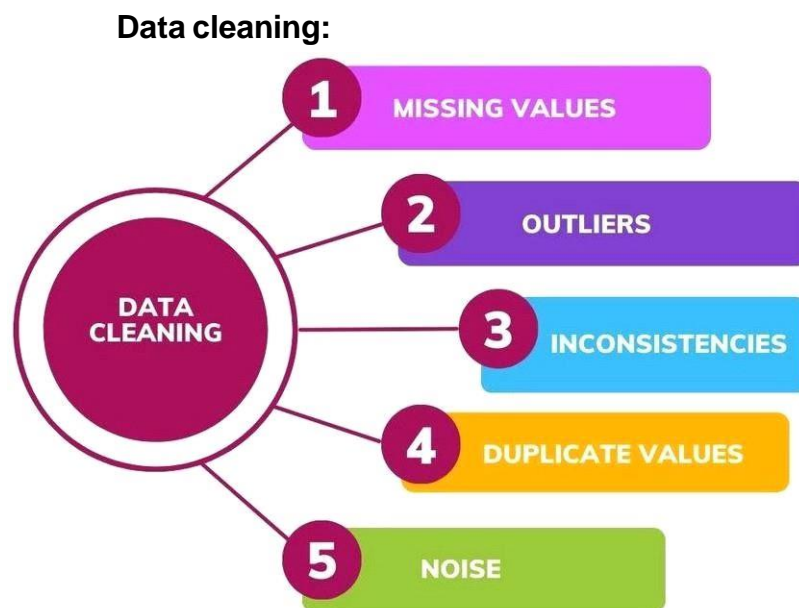
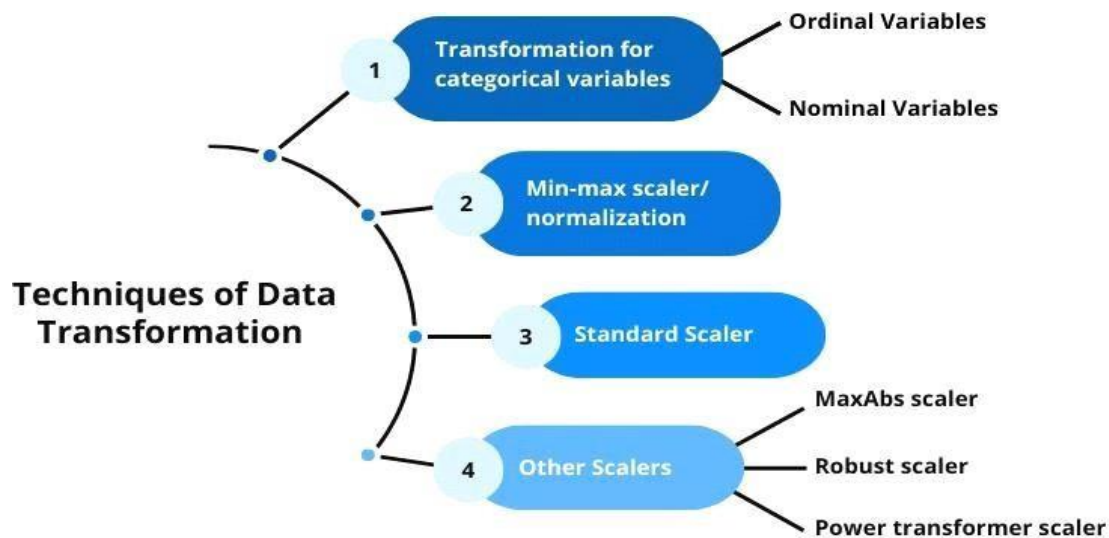


Figure 2

Data cleaning is the process of addressing missing data, noise and outliers in the datasets, which improves the overall data quality and reliability of any subsequent analysis or Modelling [2].


Data Transformation:

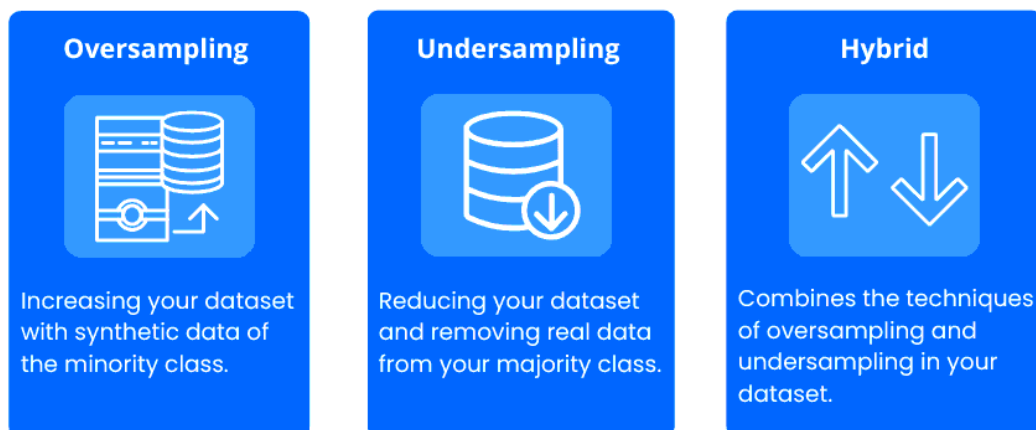


**Figure 3**

Involves converting the raw data into a form that is more suitable for analysis or modeling. This process ensures that data features are compatible the techniques of data transformation involves Normalization and Standardization Encoding Categorical variables and Discretization and Binning [2].

**Data Balancing:**

 | **Techniques to address Imbalanced Data**



**Figure 4**

Refers to the techniques that address issue of class imbalance datasets where one class (typically the majority class) dominates the datasets leading to biased predictions.

Techniques to address this like over sampling and under sampling and cost sensitive learning [3].

Now we will look at these common data preprocessing topics and highlights famous approaches and techniques used by researchers and analyst in the fields of data science and machine learning, many research papers and books shows different methodologies .we pick the most famous of these resources to give a well-rounded and accurate review on these topics.

### **Handling Missing Value Material and Technique:**

So many methods, approach, and techniques are proposed by the previous researchers, from the simplest one to the complicated one and their own advantage and drawbacks. Some basic methods are proposed in [5], such as ignoring, deleting, zero or mean or mode estimation methods. These methods above have the simplicity but only effective for low percentage of missingness, but in bigger percentage of missingness, the result will affect the result of the analysis and even biased result [6]. The main disadvantage of discarding incomplete observation is the loss of efficiency and

Biased estimation result especially when the data missingness is systematic [6]. The quality of data mining or analysis is influenced by the quality of the data.

Therefore, the data that contain missing values should be estimated to provide complete case of data to get expected result from the data [6].

Nowadays, there many studies conducted and resulting some methods or approaches to solve missing values problem. The next section will provide an explanation about the methods and approaches to estimate missing values.

- **Conventional Method:**

Many methods have been invented to deal with missing values, several of them are simple methods. Some of the methods using statistical principle as their base.

- **Ignoring** the first way to deal with missing values as pointed in [6] is ignoring the missing value, which is the simplest way to

deal missing values. The missing values is completely ignored as the analysis is carry on. Though it is a simple method yet very risky if the missingness percentage of the data are big enough to disrupt the result of the analysis [5].

- **Deletion** As what it says, deletion method is simply delete the missing variable, or the instance of observation data to continue the analysis or data mining process [5,6].
- **Mean/mode** Imputation While ignoring and deletion did not give a good result of an analysis or data mining process caused by missing values (Ignoring) and less data to be proceed (Deletion), Mean/Mode Imputation method comes as a solution to give a better result, it solves the missing values problem and the number of

Data that expected to be proceed is remain to be the same number. But the drawback of this method is the bias caused by so many values on the data have a similar value [5, 6].

- **Imputation Procedures:**

Imputation procedures can produce a complete sample of data but ignores the consequences of the imputation methods side effect. Imputation or estimation is a procedure that handling missing values problem by replacing each of the missing values by some certain values (the values source is different for each method).

From the principle of statistical field, several methods have been proposed, such as Hot and Cold Deck Imputation, Mean and Mode Imputation, and Multiple Imputation. In this section we will give a brief explanation about these methods.

- **Hot and Cold Deck Imputation:**

In hot deck the imputation is done through replacing the missing values X by matching the values from different observed data Y with similar variables X and take the values. The cold deck is in contrast from hot deck method, the imputation is done through replacing the values by the external source, like the values from the previous observation dataset with the same domain of observation. These types of method is simple, but it has a bad effect when the dataset have large amount of dataset, moreover when the assumption of missing data are MAR [7].



- **Mean Imputation:**

One of the easiest ways to impute or estimate missing values to get a complete sample is replacing each of the missing values with the mean of the observed data for that variable or known as unconditional mean imputation. Aside of mean imputation, there is median and mode that can be used for replacing missing values [7]. Kevin strike et al. [8] did an evaluation of three missing data handling techniques: leastwise deletion, mean imputation, hot deck imputation for the certain purpose.

Leastwise deletion (one of deletion method) can handle missing values for below 15%, however its accuracy is different to the increasing number of missingness percentage. Euclidean distance and z-score standardization are employed by hot deck method and proves to provide a consistent and accurate compared to the other methods. But the three methods are less effective for non-ignorable missing values compared to MCAR and MAR [7].

- **Multiple Imputation:**

Other statistical method of missing values imputation is multiple imputation which proposed by rubin [5]. In this method missing values imputed  $n$ - times to represent the uncertainty of possible values that are to be imputed. The  $n$  times values then analyzed to get a single combined estimates [7]. The result that produced by statistical method is somewhat questionable because just effective in small amount percentage of the missing values. For example, for mean imputation, the data that is imputed by this method will suffer from high bias because the newly imputed data are the same with the mean of the observed data.

In the implementation chapter, we will provide a comprehensive overview of different imputation techniques and their evaluation metrics. Most importantly, we will demonstrate which techniques are popular and yield significant improvements.

Additionally, we will address the crucial question of how to choose the most appropriate imputation technique.

## Current approaches used by practitioners:

In his famous practical book **python for data analysis orielly 3rd edition (2022)** [9] the creator of the pandas library Wes McKinney offered three ways to deal with the problems of missing value :

**1/Dropping Missing Values:** Removing rows or columns with missing data, which is appropriate when the amount of missing data is small.

**2/Imputation:** Filling missing values with a specific value, such as the mean, median, or a constant. This is particularly useful in maintaining the dataset's structure without losing valuable information.

**3/Forward/Backward Fill:** Filling missing data by propagating the next or previous value forward or backward.

Another famous author and former googler in his well acclaimed book **Hands on machine learning with sickit-learn keras and tensorflow orielly 3rd edition (2022)**[10] which consider the bible for Machine learning practitioners He highlights the importance of imputing missing data, particularly in the context of machine learning pipelines. He illustrates how the **Simple Imputer** class in Scikit- Learn can be used to automate the process of filling in missing values. Géron emphasizes that the choice of imputation strategy can significantly impact model performance and suggests experimenting with different strategies to find the most effective one.

In his well-acclaimed book, Geron introduces well-known methods for machine learning practitioners. As we mentioned earlier, many considerations must be taken into account when choosing the appropriate imputation techniques.in the last chapter . Our review will provide insights and guidelines for selecting the best techniques.

## HANDLING OUTLIERS MATERAIL AND TECHNIQUES:

Outliers, or extreme values that deviate significantly from other observations in a dataset, are a crucial issue in data preprocessing. They can distort statistical analyses and machine learning models if not appropriately addressed. Definition and Importance of Outliers can be caused by variability in the data or measurement errors.

**Aggarwal (2016)** [11] in **Outlier Analysis** explains that outliers can either represent real but rare events, or they could be errors in data collection, both of which must be carefully considered during

preprocessing . **Barnett and Lewis (1994)** [12] define outliers as data points that significantly diverge from the rest of the dataset, often leading to erroneous results in statistical analyses if not detected and handled correctly.

#### **Detection Methods Z-score Method:**

One of the simplest ways to detect outliers is by using the Z-score, where values with a Z-score greater than a specific threshold (commonly 3) are considered outliers. This method is widely discussed in **Kim and Kim (2016)** [13], where they show how Z-scores are used to identify outliers in normally distributed data.

#### **Boxplot Method:**

**Tukey (1977)** [14] introduced the boxplot, a graphical tool where outliers are those values falling outside 1.5 times the interquartile range (IQR) from the quartiles. This method is effective for detecting univariate outliers and remains popular due to its simplicity.

#### **Isolation Forests:**

Liu, Ting, and Zhou (2008) [15] developed the isolation forest algorithm for outlier detection, which isolates anomalies by creating random partitions in the data. This unsupervised learning technique is particularly useful in high- dimensional datasets.

### **Handling Outliers:**

Removal of Outliers:

1/ **Hodge and Austin (2004)** [16] recommend removing outliers when they are the result of data collection errors or noise, as they can skew the results of machine learning models.

#### **2/ Transformation Techniques:**

In cases where outliers are important, transformation techniques like log transformations can be applied to reduce their influence. **Osborne and Overbay (2004)**

[17] Suggest using log or square root transformations to minimize the effect of extreme values without removing them.

### 3/ Imputation of Outliers:

**Little and Rubin (2019)** [18] in **Statistical Analysis with Missing Data** discuss imputing outliers when they result from errors or missing values. Imputation can be a useful technique for replacing outliers with more reasonable values, thereby maintaining the dataset's structure.

Since handling outliers involves many techniques and approaches, we have limited this project to three: two statistical approaches and one graphical approach. we will cover **z score**, **tukey(1977)** and **Standard deviation method**. We will also demonstrate when and how each of these techniques is relevant to machine learning model performance.

### Impact on Machine Learning:

Outliers can negatively impact the performance of many machine learning algorithms. **Hawkins (1980)** [19] explains that outliers in regression models can disproportionately affect the fit, leading to biased predictions. Similarly, **Aggarwal (2016)** [11] mentions that algorithms like k-nearest neighbors (KNN) and support vector machines (SVM) are particularly sensitive to outliers, as they rely on distance measures.

### The Techniques of Data Normalization and Standardization:

To understand the main reason for Data Normalization and Standardization we must understand Feature scaling.

#### Feature scaling:

Feature scaling is a technique to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.

#### Why use Feature Scaling?

Gradient descent converges much faster with feature scaling than without it.

Many classifiers (like KNN, K-means) calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. So the range of features should be scaled so that each feature contributes approximately proportionately to the final distance.

However, every dataset does not require features scaling. It is required only when features have different ranges.  
**This can be achieved using two widely used techniques.**

### **1/Normalization:**

Normalization refers to the process of scaling data to fit within a specific range, typically  $[0, 1]$  or  $[-1, 1]$ . This is particularly useful when dealing with models that rely on distance metrics, such as k-nearest neighbors (KNN) or support vector machines (SVMs).

**Han, Kamber, and Pei (2011)** in their book, **Data Mining: Concepts and Techniques** [20], introduce normalization as a standard technique to transform attributes to a uniform scale. The authors emphasize its importance in distance-based algorithms, which can be sensitive to varying scales of features. **In Pattern Recognition and Machine Learning, Bishop (2006)** [21] also highlights the importance of normalization in distance-based classifiers. He discusses its utility in algorithms that rely on Euclidean distance measures, where varying scales of features can significantly impact model performance.

**Sammur and Webb (2010)** in **Encyclopedia of Machine Learning** [22] describe min- max normalization as a simple, yet effective way of scaling features, especially in scenarios where the range of data is well understood.

### **2/Standardization:**

Standardization, also known as Z-score normalization, transforms data to have a mean of 0 and a standard deviation of 1. This is particularly useful in models that assume a normal distribution of the data, such as linear regression and principal component analysis (PCA).

- James, Witten, Hastie, and Tibshirani (2013) in an Introduction to Statistical Learning Describe standardization as essential in linear models and Regularization techniques like ridge and lasso regression. They note that without standardizing the data, features with larger scales can disproportionately influence the model.
- Friedman, Hastie, and Tibshirani (2001), in the **Elements of Statistical Learning** argue that standardization is critical when variables are measured in different units, as this prevents bias in algorithms that rely on weight updates based on feature values.

- **Kuhn and Johnson (2013) in Applied Predictive Modeling**  
Discuss the necessity of standardization in the context of PCA, noting that if variables are not standardized, those with larger scales dominate the analysis.
- Historical Papers and Key Contributions one of the foundational papers on the concept of standardization was written by **Fisher (1936)**, titled **The Use of Multiple Measurements in Taxonomic Problems** [26]. In this paper, Fisher highlighted the need to standardize variables when dealing with multiple measurements in discriminant analysis.
- **LeCun, Bottou, Bengio, and Haffner (1998) in Gradient-Based Learning Applied to Document Recognition** [27] emphasized the use of normalization techniques in deep learning architectures, especially for image processing tasks where input pixels have widely varying ranges. The concept of Batch Normalization, introduced by **Ioffe and Szegedy (2015)** in **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**, [28] marks a significant advancement in the field. This paper addresses the importance of normalization within deep learning models, helping to stabilize and accelerate training.
  - In the case study, we will not focus solely on the effects of normalization and standardization on machine learning model performance. Instead, we will explore how combining these techniques with other data preprocessing methods can lead to significant improvements.

### **Comparative Studies:**

**Zhang and Ma (2012)** in their paper, **Comparison of Data Normalization Methods for Principal Component Analysis** [29], found that standardization is generally preferable when working with PCA, as it helps to align features of varying scales. However, normalization can be useful when PCA is applied to datasets where all features are already on a similar scale. In their study, **Garcia, Luengo, and Herrera (2015)**, **Data Preprocessing in Data Mining** [1], evaluated the impact of normalization and standardization on classification algorithms. They concluded that normalization generally improved the performance of algorithms that rely on distance measures, while standardization was more appropriate for statistical and probabilistic models.

## **Conclusion:**

Data preprocessing techniques have undergone remarkable evolution, driving advancements across a spectrum of applications in modern statistical research, data analysis and machine learning practices. This literature review has highlighted the pivotal role of different approaches and techniques in data preprocessing their advantages and also their weaknesses .further researches will publish techniques will continue to evolve and more ideas will enhance this very important topic.

## **Chapter 3: METHODOLOGY**

### **Introduction:**

This chapter outlines the methods and procedural steps used to achieve the objectives of this research. It provides a structured approach to reviewing and illustrating various data preprocessing techniques, emphasizing their crucial role in improving data quality and enhancing machine learning model performance. By presenting each step in a clear and systematic manner, this chapter ensures that the methodology is both transparent and replicable.

We also explore how machine learning models serve as the foundation for evaluating the effectiveness of different preprocessing techniques. For instance, we assess dataset characteristics before and after preprocessing to measure its impact on model performance. Additionally, we discuss the datasets used in this study, followed by an overview of the tools and technologies employed throughout the project.

### **Research Design:**

This research adopts a theoretical approach, providing a comprehensive review of various data preprocessing methods with illustration using famous use cases and research papers and also describing how these methods applied in practices, their drawback and effectiveness. With that in mind our research can be considered review based project, for example how outlier's detection method differ conceptually and their effect on machine learning model performance.

### **Research Scope:**

This project aims to provide a comprehensive review and analysis of data preprocessing methodologies, emphasizing their importance in data analysis and machine learning workflows. The scope includes the following:

#### **1/Review of Preprocessing Techniques:**

A detailed exploration of data preprocessing methods, including data cleaning, scaling, transformation, and dimensionality reduction.

Theoretical explanations of these techniques.



## **2/Comparison of Methods:**

Comparative analysis of preprocessing techniques based on their advantages, limitations, and impact on machine learning model performance.

Discussions on the suitability of methods for specific datasets or applications.

## **3/Evaluation Framework Using Model Performance:**

Demonstrating the impact of preprocessing on model performance through theoretical insights and practical examples.

Focus on metrics such as accuracy, precision, recall, F1-score, and mean squared error as evaluation criteria.

## **4/Case Studies and Example:**

Real-world examples to illustrate the application and outcomes of various preprocessing techniques.

Insights into how preprocessing decisions affect modeling results, supported by visual representations.

## **5/Educational Value:**

Aimed at providing a clear, structured guide for students, researchers, and practitioners to understand and apply preprocessing techniques effectively.

## **6/Implementation Chapter (Scope Expansion):**

The implementation chapter will use case studies and illustrative scenarios to validate the theoretical concepts outlined in the methodology chapter.

The focus will be on demonstrating workflows, comparative outcomes, and best practices without involving actual coding.

## The Data overview:

### 1/The Credit Card Fraud Detection dataset:

On Kaggle is a widely used resource for developing and evaluating fraud detection models. This dataset contains transactions made by European cardholders over two days in September 2013, totaling 284,807 transactions, with 492 classified as fraudulent. The dataset is highly imbalanced, with fraudulent transactions accounting for only 0.172% of all transactions.

#### Key Features:

Time: Seconds elapsed between this transaction and the first transaction in the dataset.

V1 to V28: Principal components obtained from PCA transformation to protect confidentiality.

Amount: Transaction amount.

Class: Binary indicator of fraud (1) or legitimate (0) transaction.

#### Considerations for Use:

**Data Imbalance:** The significant imbalance requires careful handling, such as resampling techniques or appropriate evaluation metrics like Precision-Recall curves.

**Feature Scaling:** While PCA-transformed features are standardized, the 'Amount' and 'Time' features may require scaling.

**Anonymized Features:** Due to anonymization, domain-specific feature engineering is limited.

**This dataset is suitable for experimenting with various preprocessing techniques and evaluating model performance in fraud detection scenarios**

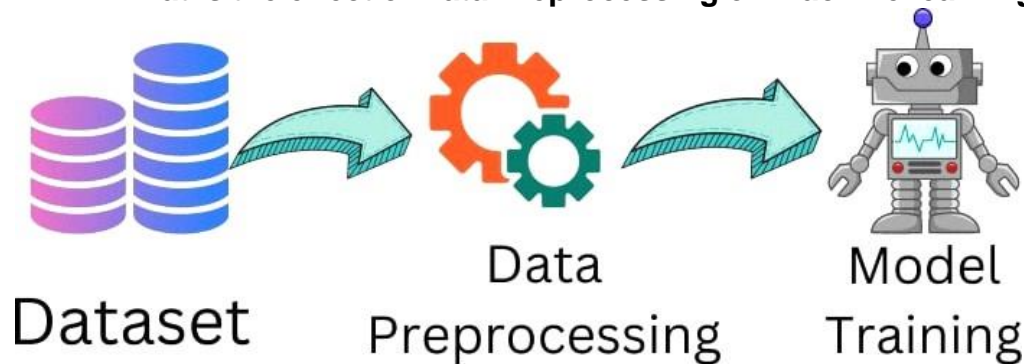
- **Data preprocessing steps:**

### **What is Data Preprocessing?**

Data preprocessing is the process of evaluating, filtering, manipulating, and encoding data so that a machine learning algorithm can understand it and use the resulting output. Data processing techniques are important. They are crucial in the ever-changing fields of data science, machine learning, and data analysis.

The major goal of data preprocessing is to eliminate data issues such as missing values, improve data quality, and make the data useful for machine learning purposes.

### **What is the effect of Data Preprocessing on Machine learning Models?**



**Figure 5**

Machine learning algorithms are statistical equations that operate on database values. As the adage goes, “If garbage goes in, garbage comes out.” Your data project will only be as successful as the input data you feed into your machine learning algorithms. Machine learning and deep learning algorithms perform best when data is presented in a way that streamlines the solution to a problem. Data wrangling, data transformation, data reduction, feature selection, and feature scaling are all examples of data preprocessing approaches teams use to reorganize raw data into a format suitable for certain algorithms. This can significantly reduce the processing power and time necessary to train a new machine learning or AI system or perform an inference against it.



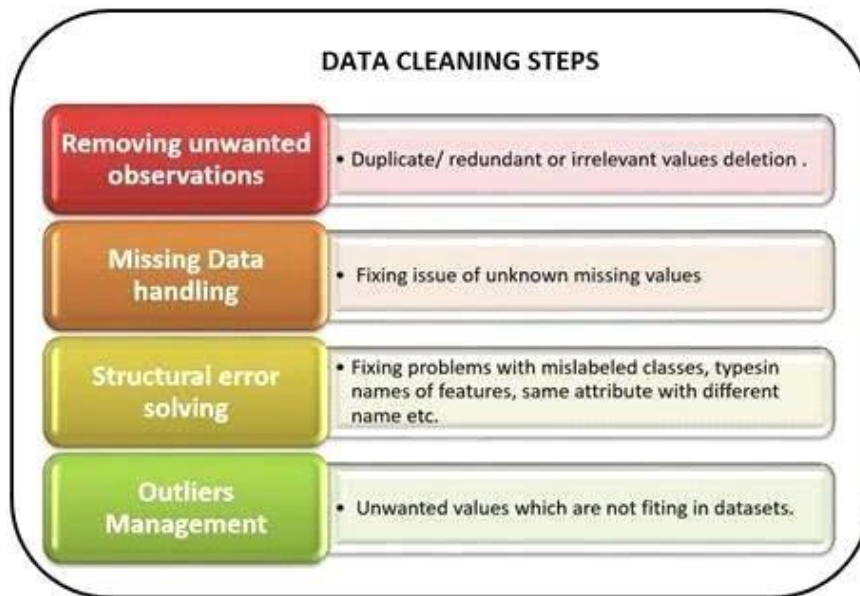
**Figure 6**

### **The Preprocessing Workflow:**

We will start by giving an introduction to the data preprocessing techniques which we will discuss in this project our focus will be on those techniques that are encounter very frequently by practitioners these techniques are:

- 1/Data Cleaning – Handling missing data, outliers, etc.
- 2/Feature Engineering – Creating meaningful features (e.g., encoding, interaction terms).
- 3/Scaling and Transformation – Normalization, standardization, logarithmic transformations.
- 4/Dimensionality Reduction – Techniques like PCA or feature selection.
- 5/Data Balancing

## 1/Data cleaning:



**Figure 7**

### Why Data Cleaning?

In data Analysis and machine learning, the quality of input data is paramount. It's a well-established fact that data quality heavily influences the performance of machine learning models and the finding results of the analysis this makes data cleaning, detecting, and correcting (or removing) corrupt or inaccurate records from a dataset a critical step in the data analysis practice.

Data cleaning is not just about erasing data or filling in missing values. It's a comprehensive process involving various techniques to transform raw data into a format suitable for analysis. These techniques include handling missing values, removing duplicates, data type conversion, and more. Each technique has its specific use case and is applied based on the data's nature and the analysis's requirements.

We will focus on the main problems in data cleaning which are:

### A/Handling missing values:

Missing data can occur for various reasons, such as errors in data collection or transfer. There are several ways to handle missing data, depending on the nature and extent of the missing values.

## **B/Outlier Detection and Removal:**

There is more than one way to detect outliers and remove them from data, we will compare different methods .Including outliers in data driven models could be risky. The existence of an extreme single misleading value has the potential to change the conclusion implied by the model. It is therefore important to manage that kind of risk.

**This project explains three of the most popular outlier detection methods:**

1/Tukey's IQR method

2/Standard deviation method

3/Z-score method

Later in the implementation chapters, we will provide recommendations and guidelines on how to choose the best outlier detection techniques depending on the specific situation the practitioner encounters. We will also demonstrate how these techniques are implemented, discussing their advantages and drawbacks.

## **C/removing Duplicates**

### **The Duplicates Dilemma?**

Duplicates are outside the scope of this project because their impact on model performance and evaluation introduces complexities that are not the focus of our study. While duplicate data may be relevant in certain contexts, it can also distort model evaluation metrics, making it difficult to interpret results accurately. For example, accuracy and other efficacy metrics can be misleading without a clear understanding of how many duplicates exist and whether they reinforce or contradict existing patterns in the dataset.

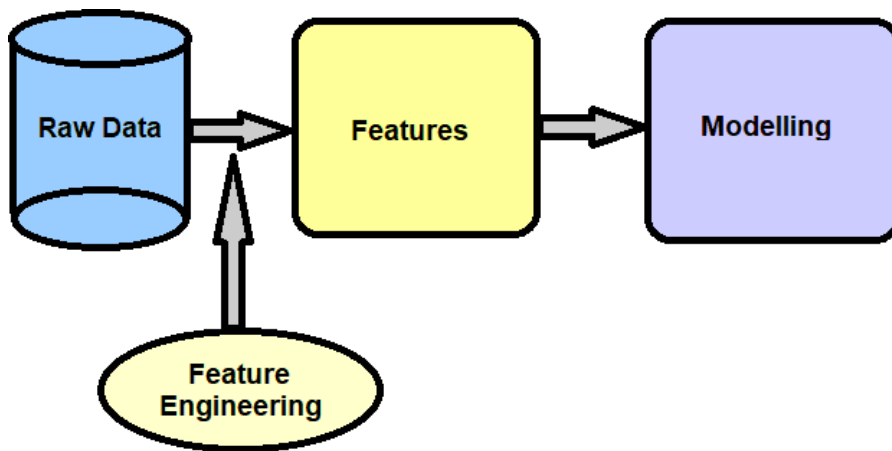
## **2/Data transformation:**

Data transformation is a critical part of data preprocessing in which raw data is converted into a unified format or structure. Data transformation ensures compatibility with target systems and enhances data quality and usability. It is an essential aspect of data management practices including data wrangling, data analysis and data

warehousing. We will focus on Feature Engineering. , particularly feature extraction and feature selection. We will explore a couple of use cases to illustrate how these two techniques can significantly affect machine learning model performance. For example, we will examine how feature selection and transformation by machine learning can improve predictions for heart failure.

Now we will give definitions and explanations of these techniques:

### 1/Feature Engineering:



**Figure 8**

### What is Feature Engineering?

Feature engineering is the process of transforming raw data into features that are suitable for machine learning models. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models. The success of machine learning models heavily depends on the quality of the features used to train them. Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.

Feature Engineering are divided into:

## 1. Feature Transformation:

Feature Transformation is the process of transforming the features into a more suitable representation for the machine learning model. This is done to ensure that the model can effectively learn from the data.

Types of Feature Transformation:

1/Normalization: Rescaling the features to have a similar range, such as between 0 and 1, to prevent some features from dominating others.

2/Scaling: Scaling is a technique used to transform numerical variables to have a similar scale, so that they can be compared more easily. Rescaling the features to have a similar scale, such as having a standard deviation of 1, to make sure the model considers all features equally.

3/Encoding: Transforming categorical features into a numerical representation. Examples are one-hot encoding and label encoding.

4/Transformation: Transforming the features using mathematical operations to change the distribution or scale of the features. Examples are logarithmic, square root, and reciprocal transformations.

From these we will review scaling and Transformation and how combining them yielded the best results, particularly for SVM, which is sensitive to feature scales.

## 2/Feature Extraction:

Feature Extraction is the process of creating new features from existing ones to provide more relevant information to the machine learning model. This is done by transforming, combining, or aggregating existing features. When it comes to **Feature Extraction we will focus on Dimensionality Reduction**: Reducing the number of features by transforming the data into a lower-dimensional space while retaining important information. Examples are PCA.

As mentioned earlier, we will see the implementation of these powerful techniques in the use case of predicting heart failure. We will use feature selection and extraction to illustrate how these techniques can enhance model performance.



### 3/ Feature Selection:

Feature Selection is the process of selecting a subset of relevant features from the dataset to be used in a machine-learning model. It is an important step in the feature engineering process as it can have a significant impact on the model's performance.

#### Types of Feature Selection:

1/Filter Method: Based on the statistical measure of the relationship between the feature and the target variable. Features with a high correlation are selected.

2/Wrapper Method: Based on the evaluation of the feature subset using a specific machine learning algorithm. The feature subset that results in the best performance is selected.

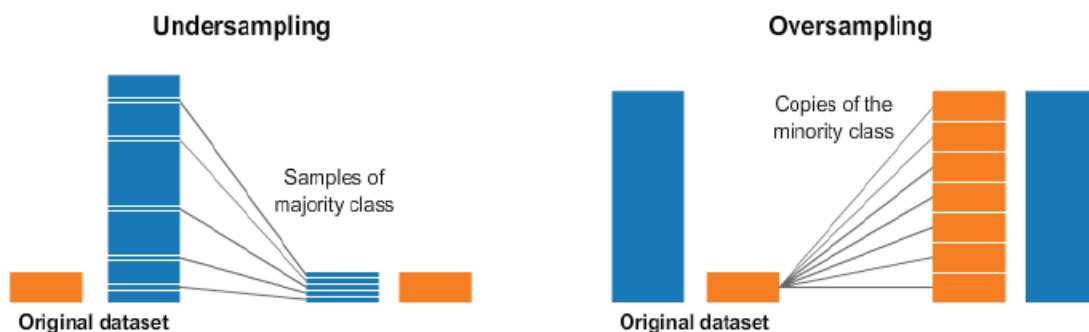
3/Embedded Method: Based on the feature selection as part of the training process of the machine learning algorithm.

In the implementation chapter, we will explore both statistical and wrapper methods in the study of feature selection and variable transformation for predicting heart failure.

### 3/Data Balancing:

#### What is Imbalanced Data and How to handle it?

Imbalanced data pertains to datasets where the distribution of observations in the target class is uneven. In other words, one class label has a significantly higher number of observations, while the other has a notably lower count. When one class greatly outnumbers the others in a classification, there is imbalanced data. Machine learning models may become biased in their predictions as a result, favoring the majority class. Techniques like oversampling the minority class or undersampling the majority class are used in resampling to remedy this.



The figure above show how data balancing work

**There are mainly two mainly algorithms that are widely used for handling imbalanced class distribution.**

1/SMOTE

2/Near Miss Algorithm

In the implementation chapter, we will demonstrate how data balancing works in practice.

### **3.6. Real Case Studies for Preprocessing and Model Performance Evaluation:**

#### **Case 1: Research papers overview of the effect of imputations technique on model performance:**

As we will explain in the case study, handling missing values using imputation techniques is quite tricky and depends on various perspectives, such as the percentage of missing data and the methodological approach. When we talk about the methodological perspective, we consider whether to use statistical approaches, machine learning-based approaches, or a hybrid approach. Therefore, instead of providing a single case study, we choose to give an overview of the state of the art by referencing some recently published papers. We will focus on machine learning-based techniques and their impact on model performance.

#### **Case 2: Feature selection and transformation by machine learning reduce variable numbers and improve prediction for heart failure readmission or death:**

The prediction of readmission or death after a hospital discharge for heart failure (HF) remains a major challenge. Modern healthcare systems, electronic health records, and machine learning (ML) techniques allow us to mine data to select the most significant variables (allowing for reduction in the number of variables) without compromising the performance of models used for prediction of readmission and death. Moreover, ML methods based on transformation of variables may potentially further improve the performance.

### **Case Study 3: Dimensionality Reduction in an E-commerce Recommendation Dataset**

Dataset: User-item interaction data with hundreds of features representing product categories, demographics, and user behavior.

Issue: High dimensionality led to overfitting and poor generalization in the model.

Preprocessing Techniques:

Principal Component Analysis (PCA): Reduced 300 features to 50 components, retaining 95% of variance.

Feature Selection: Retained only the top 20 features based on mutual information scores.

Model Used: Gradient Boosting Classifier for predicting purchase likelihood.

### **Case Study 4: Class Imbalance Handling in a Fraud Detection Dataset**

Dataset: Transaction dataset with a fraud class representing only 2% of the total instances.

Issue: Imbalanced classes caused the model to predict the majority class almost exclusively.

Preprocessing Techniques:

Oversampling: Used SMOTE (Synthetic Minority Oversampling Technique) to balance the classes.

Under sampling: Randomly reduced the majority class size for balance.

Model Used: Logistic Regression for fraud prediction.

### **Challenges and Limitations:**

#### **1/Data Availability and Quality:**

Access to diverse, real-world datasets that demonstrate the impact of preprocessing may be limited.

Public datasets might lack sufficient documentation, leading to potential misinterpretation of features and labels.

## **2/Applicability of Preprocessing Techniques:**

Techniques such as PCA or predictive imputation depend on assumptions (e.g., linear relationships, normality) that may not hold for all datasets.

Results from preprocessing strategies may not generalize across domains, reducing the practical utility of findings.

## **3/Model Dependency:**

The effectiveness of preprocessing is often model-dependent. A method that improves performance for one algorithm (e.g., SVM) may not work for another (e.g., tree-based models).

Different preprocessing strategies require varied evaluation criteria, increasing complexity.

## **4/Theoretical Scope vs. Practical Implementation:**

Since no code implementation is required, demonstrating preprocessing impact through theory and examples alone may not fully convey the depth of its influence on model performance.

Real-world implementation challenges, such as computational costs or scalability, are excluded from this study.

## **5/Evaluation without Model Retraining:**

Theoretical evaluations using case studies rely on previously reported outcomes, which might not reflect preprocessing impact in novel or customized use cases.

## **Chapter 4: Implementation**

### **4.1 Introduction:**

In this chapter, we delve into the theoretical foundations of this project and explore how various data preprocessing techniques are applied in practice. We present four case studies that highlight the impact of these techniques on machine learning models, demonstrating their effectiveness in improving data quality and model performance.

We begin by introducing key definitions and concepts essential for understanding the results and findings of these studies. Following this, we discuss four case studies previously outlined in the methodology chapter, analyzing their implications in detail. Finally, we introduce an additional article on outlier detection to provide a broader perspective on the role of data preprocessing in different machine learning scenarios.

### **4.2 Some important definitions:**

#### **Feature (variable) extraction**

This is the process of creating a new and a smaller set of variables, with the aim to capture the most useful information that is present in the original variables, to predict the outcome. The new variables are produced by applying a transformation to the original variables. The transformed variables represent projections of the original variables onto a new variable space, where the distinct outcome groups have a better separation compared to the original variable space.

**Principal Component Analysis.** (PCA) is a popular feature extraction method that creates a linear transformation of the input variables. The new variables, called the principal components, are the projections of the original variables to a new variable space.

#### **SMOTE (Synthetic Minority Oversampling Technique) – Oversampling**

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesizes new minority instances between existing minority instances.

## **NearMiss Algorithm – Undersampling**

NearMiss is an under-sampling technique. It aims to balance class distribution by randomly eliminating majority class examples. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes. This helps in the classification process

## **Receiver Operating Characteristics (ROC) Curve:**

ROC stands for Receiver Operating Characteristics, and the ROC curve is the graphical representation of the effectiveness of the binary classification model. It plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.

## **Area under Curve (AUC) Curve:**

AUC stands for the Area under the Curve, and the AUC curve represents the area under the ROC curve. It measures the overall performance of the binary classification model. As both TPR and FPR range between 0 to 1, So, the area will always lie between 0 and 1, and A greater value of AUC denotes better model performance. Our main goal is to maximize this area in order to have the highest TPR and lowest FPR at the given threshold. The AUC measures the probability that the model will assign a randomly chosen positive instance a higher predicted probability compared to a randomly chosen negative instance.

## **Handling missing values:**

Causes and Types of Missing Data. Data can be missing due to various reasons such as sensor failures, human errors, or privacy concerns. There are three primary types of missing data:

1/Missing Completely at Random (MCAR): The probability of missing data is independent of observed or unobserved variables.

2/Missing at Random (MAR): Missingness depends on observed data but not on the missing values themselves.

3/Not missing at Random (NMAR): The probability of missing data depends on the missing values, making it more challenging to address.

## **Case 1: Papers:**

### **Introduction:**

Providing a single successful study on imputation techniques is challenging since many factors must be considered, and the sheer number of papers and articles on this topic is overwhelming. That's why we have chosen to present a comprehensive review of state-of-the-art methods using various research papers and articles. This approach allows us to highlight current practices and emerging trends, with a particular focus on machine learning-based imputation rather than traditional statistical methods. Many articles and papers have been written on this important topic. In this overview we used, most notably two comprehensive research papers that covers all imputation techniques. The first one is :

**"Missing Value Imputation Affects the Performance of Machine Learning: A Review and Analysis of the Literature (2010–2021)"** This paper reviews the

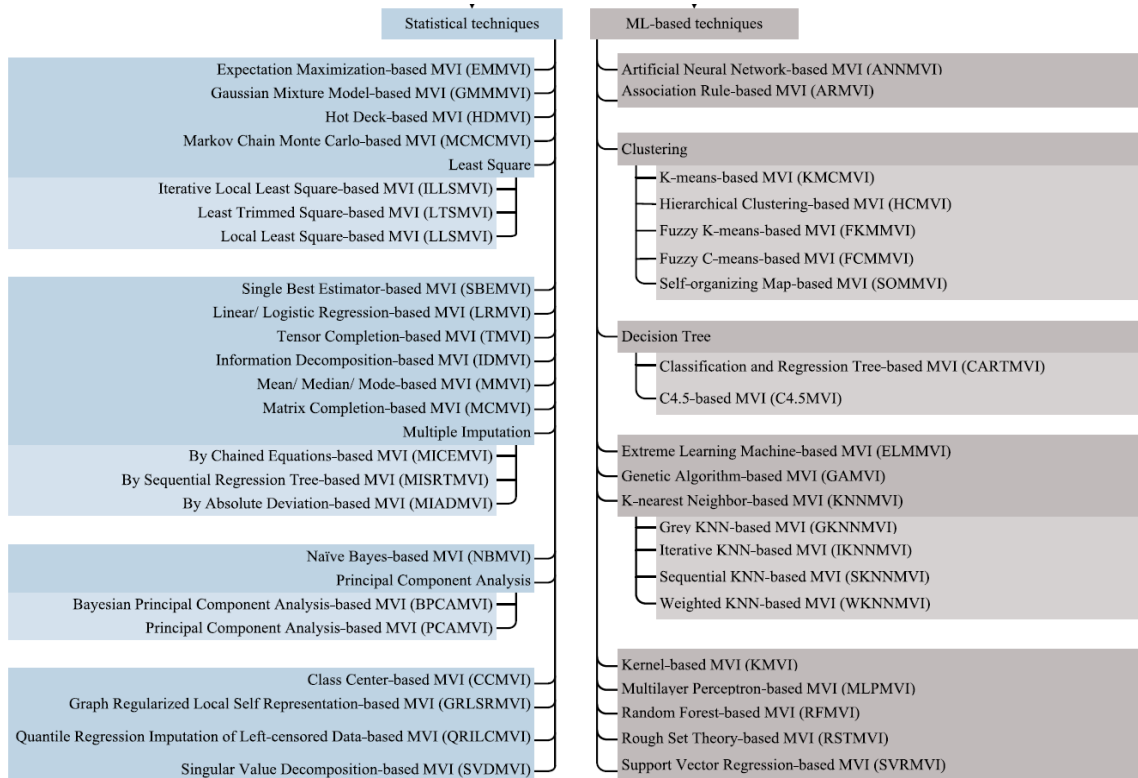
impact of missing value imputation techniques on machine learning (ML) performance. It emphasizes that missing data is a common issue in real-world datasets and must be handled properly to prevent biased or misleading results.

The authors provide an extensive and detailed analysis, covering literature from 2010 to 2021, reviewing 191 papers and articles that examine various imputation strategies and their effects on ML models. What so impressive about this study is that it not only categorizes MVI techniques but also evaluates their performance using various metrics and classification models. And the other one is **WC Tsai**

**missing value imputation: a review and analysis of the literature (2006-2017)** this paper reviews and analyzes 111 studies on missing value imputation (MVI) published between 2006 and 2017, focusing on how these studies were designed. It examines key challenges in the MVI process, such as selecting datasets, handling different levels of missing data, understanding missingness patterns, and choosing the right imputation techniques and evaluation metrics.

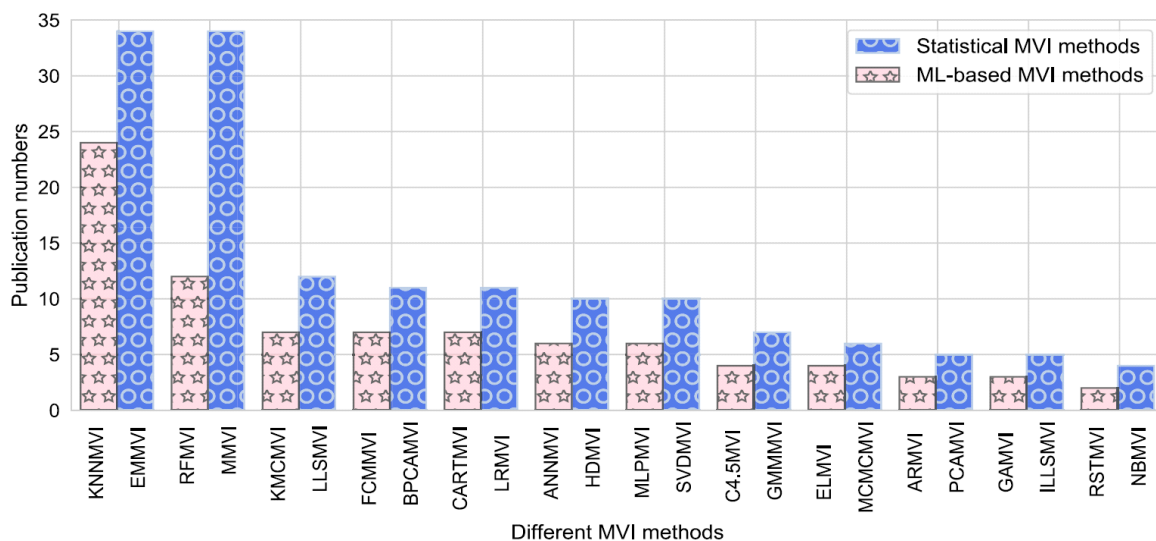
Also articles by Jason Brownlee PhD **on machinelearningmastery.com** have been used in particular **kNN Imputation for Missing Values in Machine Learning** and **statistical imputation for missing values**.

The figure below shows the categorized tree exhibition of the commonly employed MVI methods available in the literature.



**Figure 9**

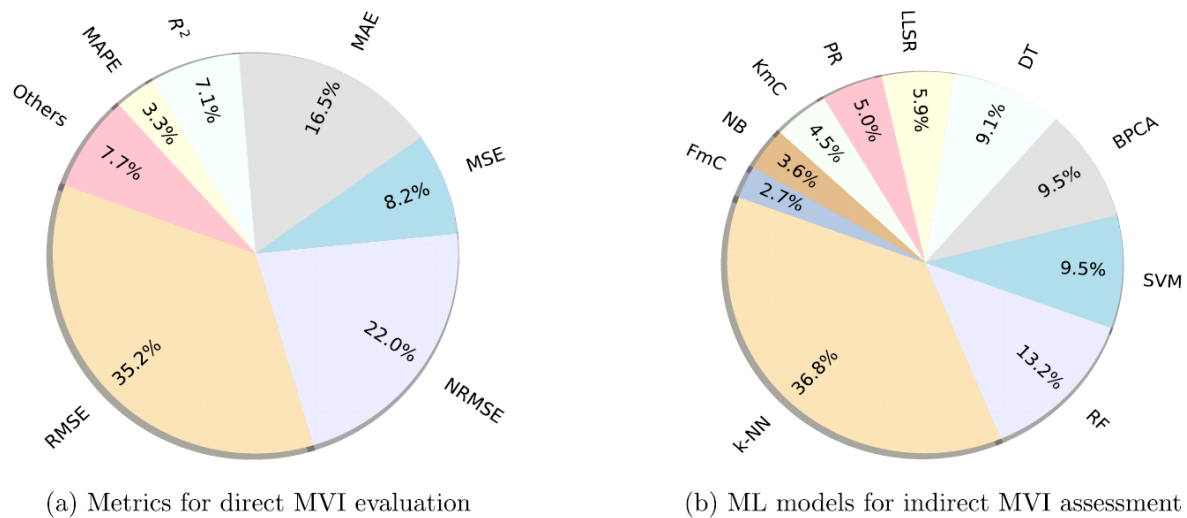
The below figure show top twelve most commonly employed MVI methods in the order of the highest to the lowest utilizations. The highest to lowest statistical and ML based methods are arranged side by side to compare those statistical and ML based methods appliances



**Figure 10**



the left figure (a) below indicates the most usually exercised direct metrics are RMSE, NRMSE, MSE, MAE, R2 AND MAPE only 7.7% articles practiced the rest of the metrics ,it's also points to be noted that RMSE and NRMSE are metrics applied in 57.2% articles therefore according to the last decades trends, RMSE metric is far better evaluation criterion for measuring the accuracy of the data missingness imputation on the other hand figure (b) bestows the percentage of top 10 most widley employed ML models such as KNN ,RF, BPCA ,DT ,LLSR ,PR, KMC, NB, AND FCMC for indirect evaluation .it is remarkable from figure (b) that KNN and RF models are the most widley employed ML models for indirect assessment in the last decade ,a total of 50% articles from 2010 to 2021 .



**Figure 11**

### **(a) Metrics for Direct MVI Evaluation**

This pie chart shows the distribution of different metrics used to directly evaluate missing value imputation methods. These metrics assess the accuracy and performance of the imputation methods by comparing imputed values to actual values.

#### **Breakdown of Metrics**

##### **Root Mean Squared Error (RMSE) – 35.2%**

RMSE is the most commonly used metric for evaluating imputation accuracy. It measures the square root of the average squared differences between actual and imputed values. Lower RMSE indicates a better imputation method.

**Normalized Root Mean Squared Error (NRMSE) – 22.0%**

A variation of RMSE that normalizes the error based on the range of the dataset. Helps in comparing performance across different datasets.

**Mean Absolute Error (MAE) – 16.5%**

Measures the average absolute difference between actual and imputed values. Unlike RMSE, it does not square the errors, making it less sensitive to large deviations.

**Mean Squared Error (MSE) – 8.2%**

Similar to RMSE but without taking the square root.

It penalizes larger errors more than MAE, making it useful for detecting major imputation failures.

**R<sup>2</sup> (Coefficient of Determination) – 7.1%**

Measures how well the imputed values explain the variance in the original dataset.

A higher R<sup>2</sup> value indicates a better imputation method.

**Mean Absolute Percentage Error (MAPE) – 3.3%**

Measures error as a percentage of actual values, making it useful for datasets with different scales.

**Others – 7.7%**

Includes less common metrics such as bias measures and correlation-based evaluations.

**(b) Machine Learning Models for Indirect MVI Assessment**

This pie chart presents the distribution of machine learning models used for indirect assessment of missing value imputation. Instead of evaluating imputation directly, these models assess how well imputed data performs in downstream ML tasks (e.g., classification, regression).

## **Breakdown of ML Models**

### **k-Nearest Neighbors (K-NN) – 36.8%**

The most frequently used ML model for indirect evaluation. It predicts missing values based on the similarity of nearby data points.

### **Random Forest (RF) – 13.2%**

A powerful ensemble method that can handle missing data during training.

Used for assessing how well imputation methods restore predictive accuracy.

### **Support Vector Machine (SVM) – 9.5%**

A classification and regression model used to test the effectiveness of imputation in structured datasets.

### **Bayesian Principal Component Analysis (BPCA) – 9.5%**

A probabilistic approach that estimates missing values using Bayesian inference.

Often used for imputing missing data in high-dimensional datasets.

### **Decision Trees (DT) – 9.1%**

A simple yet effective model for testing how missing values impact decision boundaries.

### **Least Squares Regression (LLSR) – 5.9%**

A linear regression model that assesses imputation by measuring prediction errors.

### **Polynomial Regression (PR) – 5.0%**

Similar to LLSR but captures non-linear relationships in data.

### **k-Means Clustering (KmC) – 4.5%**

A clustering method used to test how missing data affects data grouping.

### **Naïve Bayes (NB) – 3.6%**

A probabilistic classifier used to study how imputed data affects classification performance.

## Factorial Methods for Component Analysis (FmC) – 2.7%

Used for dimensionality reduction and evaluating the imputation impact on principal component distributions.

### Key Takeaways from the pie chart figure (a) and (b) above:

**1/**RMSE and NRMSE are the most commonly used metrics for evaluating missing value imputation, as they provide direct error measurements.

**2/**k-NN is the most frequently used ML model for assessing imputation quality, followed by RF and SVM.

**3/**Indirect evaluation methods (right chart(a) test how missing values affect overall machine learning model performance rather than just comparing imputed values to actual values.

**4/**A mix of classification, regression, and clustering models is used to assess the effectiveness of imputed data.

**5/**This visualization provides a comprehensive overview of how missing value imputation methods are evaluated in machine learning research.

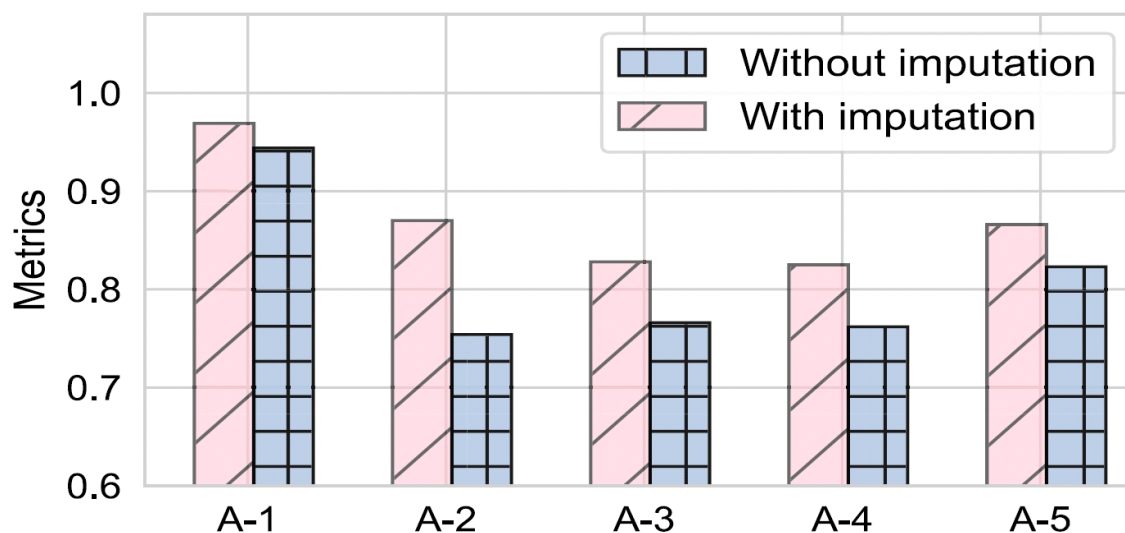
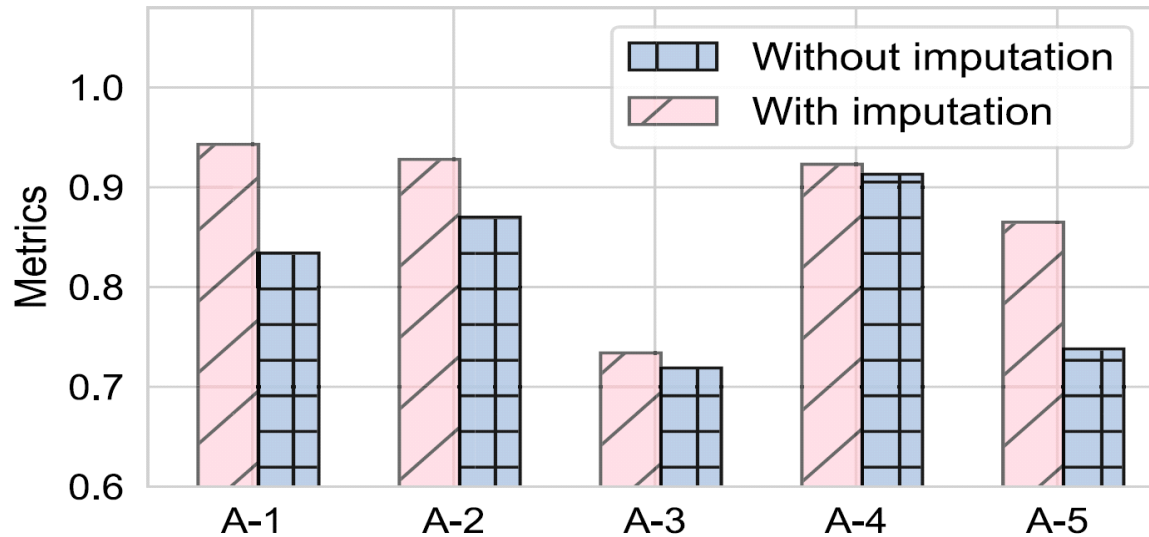


Figure 12

**The figure above is a demonstration of MLs performance improvements due to MVI method incorporation for the heart disease datasets (The Heart Disease (HD) dataset is available in UCI Machine Learning Repository)**

Where the articles A-1 to A-5, respectively, denote Nilashi et al.(1), Khennou et al. (2), Setiawan et al. (3), Saini et al. (4), and Rani et al. (5).



**Figure 13**

The figure above Demonstration of ML's performance improvements due to MVI method incorporation for the PID(***PIMA Indians Diabetes (PID) dataset***) , where the articles A1 to A5 **respectively, denote**

Hasan et al(6) Wang et al.(7) Christobel and SivaPrakasam(8) Maniruzzaman et al(9) and Kandhasamy and Balamurali(10)

## **Case study 2: Feature selection and transformation by machine learning reduce variable numbers and improve prediction for heart failure admission or death:**

Now we will review this interesting case study first we will give the researchers conclusion and our points of view will be in the end.

### **Objective**

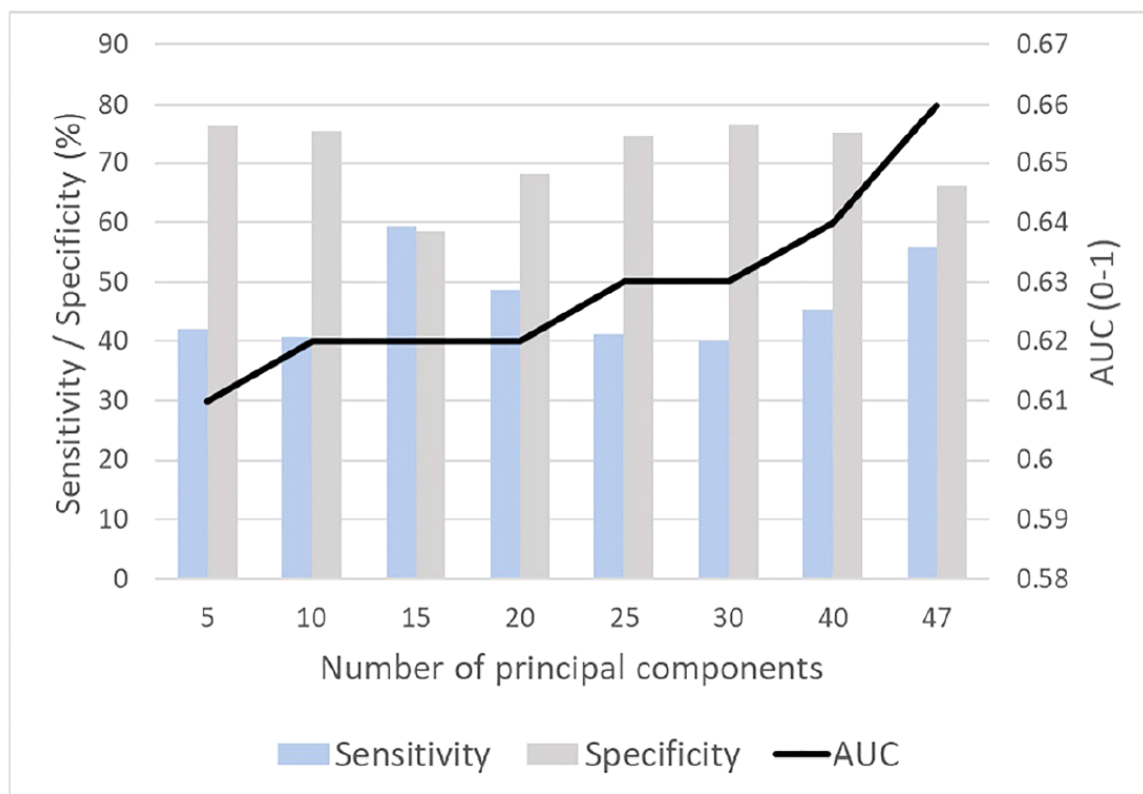
To use ML techniques to determine the most relevant and also transform variables for the prediction of 30-day readmission or death in HF patients.

### **Methods**

The study identified all Western Australian patients aged 65 years and above admitted for HF between 2003–2008 in linked administrative data. Then the researchers evaluated variables associated with HF readmission or death using standard statistical and ML based selection techniques. We also tested the new variables produced by transformation of the original variables. They developed multi-layer perceptron prediction models and compared their predictive performance using metrics such as Area Under the receiver operating characteristic Curve (AUC), sensitivity and specificity.

## **Results and conclusion reached by this study:**

Following hospital discharge, the proportion of 30-day readmissions or death was 23.7% in the researcher's cohort of 10,757 HF patients. The prediction model developed by them using a smaller set of variables ( $n = 8$ ) had comparable performance (AUC 0.62) to the traditional model ( $n = 47$ , AUC 0.62). Transformation of the original 47 variables further improved ( $p < 0.001$ ) the performance of the predictive model (AUC 0.66).



**Figure 14**

Figure (14): **Performance of the multi-layer perceptron (MLP) prediction model on variables generated by principal component analysis.**

This figure illustrates the performance of a multi-layer perceptron (MLP) prediction model when using variables transformed by Principal Component Analysis (PCA). The graph tracks three key performance metrics—Sensitivity (blue bars), Specificity (gray bars), and AUC (black line)—as the number of principal components increases from 5 to 47.

**Sensitivity (Blue Bars):** Represents the model's ability to correctly identify positive cases. Sensitivity starts relatively low (~30-40%) and fluctuates slightly as more principal components are added.

**Specificity (Gray Bars):** Measures the model's ability to correctly identify negative cases. Specificity remains consistently higher (~70%) compared to sensitivity.

**AUC (Black Line, right y-axis):** Represents overall model performance. The AUC gradually improves, starting at 0.61 with 5 principal components and reaching 0.66 with 47 principal components, indicating that increasing the number of principal components enhances model performance.

## Case Study 3: Implementation of Dimensionality Reduction in E-Commerce Recommendation:

### Introduction:

This case study presents two different experiments where one technology called *Singular Value Decomposition (SVD)* explore to reduce the dimensionality of recommender system databases. Each experiment compares the quality of a recommender system using SVD with the quality of a recommender system using collaborative filtering. The first experiment compares the effectiveness of the two recommender systems at predicting consumer preferences based on a database of explicit ratings of products. The second experiment compares the effectiveness of the two recommender systems at producing *Top-N* lists based on a real-life customer purchase database from an E-Commerce site.

### Case Study:

Application of Singular Value Decomposition (SVD)

#### Dataset Overview:

**Source:** Historical purchase records from a large e-commerce company.

Size: 6,502 users.

23,554 catalog items.

97,045 non-zero purchase entries, representing 99.996% sparsity.

**Data Transformation:** Converted all purchase amounts to binary values (1 for purchased, 0 otherwise) for consistency in modeling.

#### Preprocessing Techniques:

The Preprocessing Techniques that have been applied is 1/SVD for Dimensionality Reduction:

The original user-item matrix was factorized into three matrix  $u$ ,  $s$  and  $v$ : where  $S$  is the Reduced to retain only the top  $k$  singular values. And  $R_k$  is Low-rank approximation of the user-item matrix obtained by reconstructing  $R$  using reduced  $u$ ,  $s$  and  $v$ .

#### 2/Neighborhood Formation in Low-Dimensional Space:

Applied cosine similarity to form user neighborhoods in the reduced  $k$ -dimensional space.

Generated recommendations by analyzing purchase behavior within these neighborhoods.

Prediction experiment results



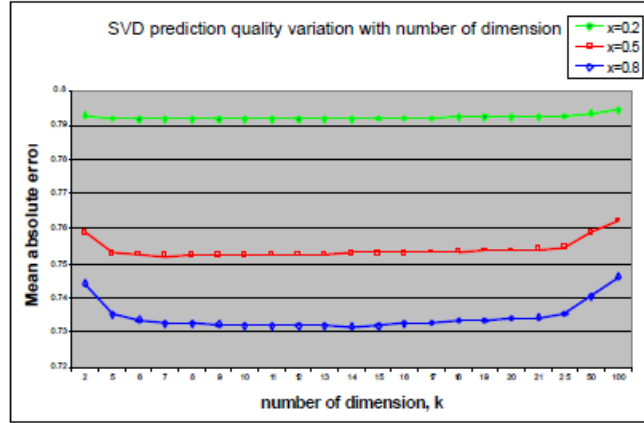


Figure 15

Top N recommendation experiment results

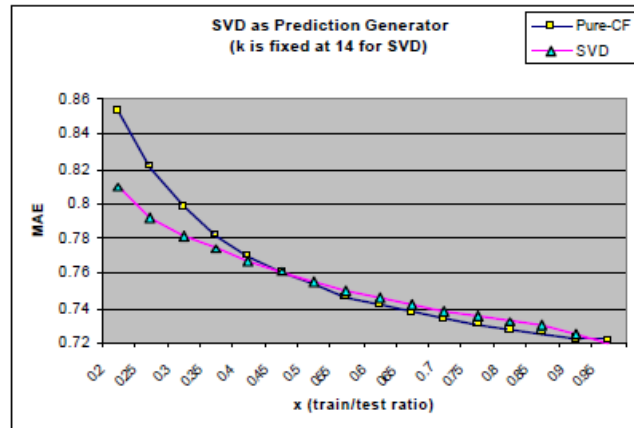
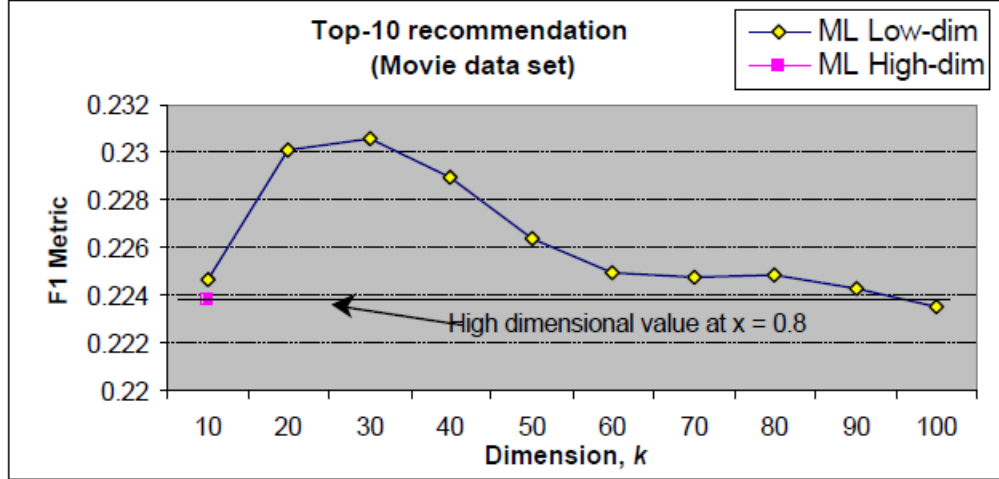


Figure 16

Determination of optimum value of k. (figure 16) SVD vs. CF-Predict prediction quality (figure 15)

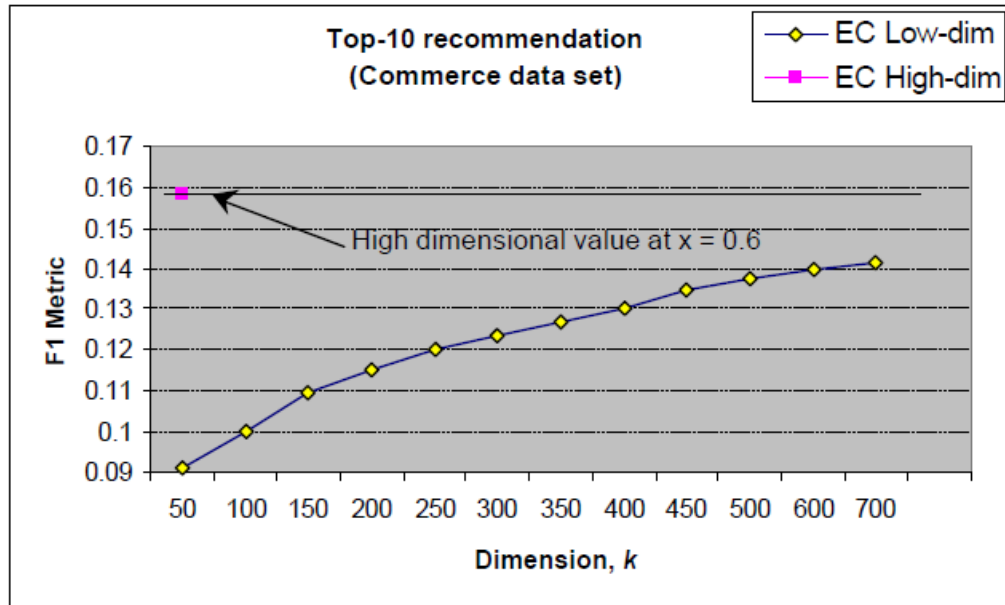
## Results

In case of the prediction experiment, we observe that in Figure (15) for  $x < 0.5$  SVD-based prediction is better than the CF-Predict predictions. For  $x > 0.5$ , however, the CF-Predict predictions are slightly better. This suggests that nearest-neighbor based collaborative filtering algorithms are susceptible to data sparsity as the neighborhood formation process is hindered by the lack of enough training data. On the other hand, SVD based prediction algorithms can overcome the sparsity problem by utilizing the latent relationships. However, as the training data is increased both SVD and CF-Predict prediction quality improve but the improvement in case of CF Predict surpasses the SVD improvement.



**Figure 17**

*Top-10 recommendation results for the Movie Lens data set.*



**Figure 18**

*From the plots of the recommender results (Figures 17 and 18), we observe that for the movie data the best result happens in the vicinity of  $k=20$  and in case of the e-commerce data the recommendation quality keeps on growing with increasing dimensions. The movie experiment reveals that the low dimensional results are better than the high dimensional counterpart at all values of  $k$ .*

In case of the ecommerce experiment the high dimensional result is always better, but as more and more dimensions are added low dimensional values improve. However, we increased the dimension values up to 700, but the low dimensional values were still lower than the high dimensional value. Since the commerce data is very high dimensional (6502x23554), probably such a small  $k$  value is not sufficient to provide a useful approximation of the original space. Also another factor to consider is the amount of sparsity in the data sets, the movie data is 95.4% sparse (100,000 nonzero entries in 943x1,682 matrix), while the ecommerce data is 99.996% sparse (97,045 nonzero entries in 6,502x23,554 matrix). To test this hypothesis we deliberately increased sparsity of our movie data (i.e., remove nonzero entries) and repeated the experiment and observed dramatic reduction in F1 values!

Overall, the results are encouraging for the use of SVD in collaborative filtering recommender systems.

The SVD algorithms fit well with the collaborative filtering data, and they result in good quality predictions. And SVD has potential to provide better online performance than correlation-based systems.

In case of the *top-10* recommendation experiment we have seen even with a small fraction of dimension, i.e., 20 out of 1682 in movie data, SVD-based recommendation quality was better than corresponding high dimensional scheme.

## **Case study 4: Implementation of Fraud Detection Using Imbalanced Dataset:**

### **Introduction:**

In the context of machine learning, an imbalanced classification problem states to a dataset in which the classes are not evenly distributed. This problem commonly occurs when attempting to classify data in which the distribution of labels or classes is not uniform. Using resampling methods to accumulate samples or entries from the minority class or to drop those from the majority class can be considered the best solution to this problem. The focus of this case study is to propose a framework pattern to handle any imbalance dataset for fraud detection. For this purpose, Undersampling (Random and NearMiss) and oversampling (Random, SMOTE, BorderLine SMOTE) were used as resampling techniques for the concentration of this experiments for balancing an evaluated dataset. a large-scale unbalanced dataset collected from the Kaggle website was used to test both methods for detecting fraud in the Tunisian company for electricity and gas consumption. It was also evaluated with four machine learning classifiers: Logistic Regression (LR), Naïve Bayes (NB), Random Forest, and XGBoost.

Standard evaluation metrics like precision, recall, F1-score, and accuracy have been used to assess the findings. The experimental results clearly revealed that the RF model provided the best performance and outperformed all other matched classifiers with attained a classification accuracy of 89% using NearMiss undersampling and 99% using Random oversampling.

### **Dataset Overview**

**Source:** Publicly available dataset from Kaggle, provided by the Tunisian Company of Electricity and Gas.

Characteristics:

Size: 4,476 samples with 21 features.

Class Distribution:

Fraudulent cases: 353 (minority class).

Non-fraudulent cases: 4,123 (majority class).

Features: Include Client ID, Invoice Date, Tariff Type, Counter Type (electricity or gas), and Consumption Levels.

## **Preprocessing Steps:**

### **1/Data Cleaning and Integration:**

A- Combined two CSV files (client and invoice datasets) into a single dataframe based on the Client ID.

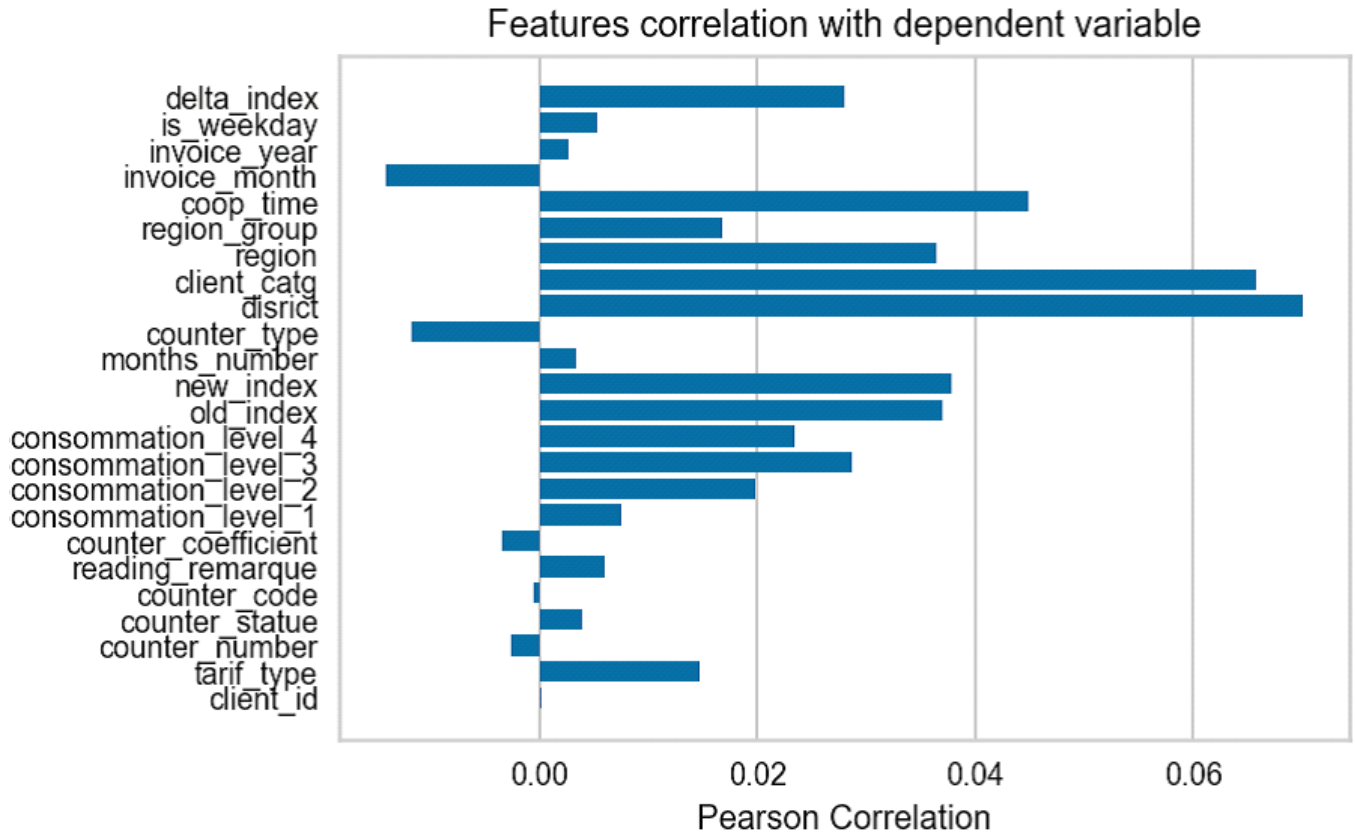
B- Addressed missing values and removed inconsistencies to enhance data quality.

### **2/Data transformation:**

Two different processes are applied that are data scalar and data clipping. The data transformation process is adopted to convert some features of the evaluated dataset from categorical to numerical format. Data scalar is also adopted to scale each input data feature in training and testing sets independently by subtracting the mean (called centering) and splitting by the standard deviation to transfer the scattering of the dataset to have a mean of zero and a standard deviation of one.

### **3/Feature Selection and Transformation:**

Conducted Pearson correlation analysis to identify relationships between features. Feature selection and data clipping Feature selection aimed to select significant features<sup>53</sup> from the used dataset to be fed to a machine learning classifier for the prediction process and drop the rest. In this phase, the features are ranked and decided to be adopted through this experiment. Accordingly, the major occurrence dissemination of the dataset features has been investigated, and the correlation matrix among the features has been computed. Figure below demonstrates the Pearson correlation for the features and dependent variables (target classes). From the result of the correlation, it was observed that there is an insignificant association between features, which indicates that the dataset features are generally independent of each other.



**Figure 19**

#### ***4/Data resampling techniques***

The dataset evaluated in this experiments has an imbalance problem in its classes; on the other hand, for balancing the distribution of its classes, the researcher applied various data sampling techniques on the used dataset for gaining robust results. The description of these techniques is presented as in Table 1 below:

Summarization of oversampling techniques.

Paper ID	Resampling Technique
Chawla <i>et al.</i> <sup>31</sup>	SMOTE
He <i>et al.</i> <sup>33</sup>	ADASYN
Chen <i>et al.</i> <sup>37</sup>	WRF, BR
Zhang <i>et al.</i> <sup>38</sup>	RWO
Han <i>et al.</i> <sup>29</sup>	BorderLine-SMOTE

**Figure 20**

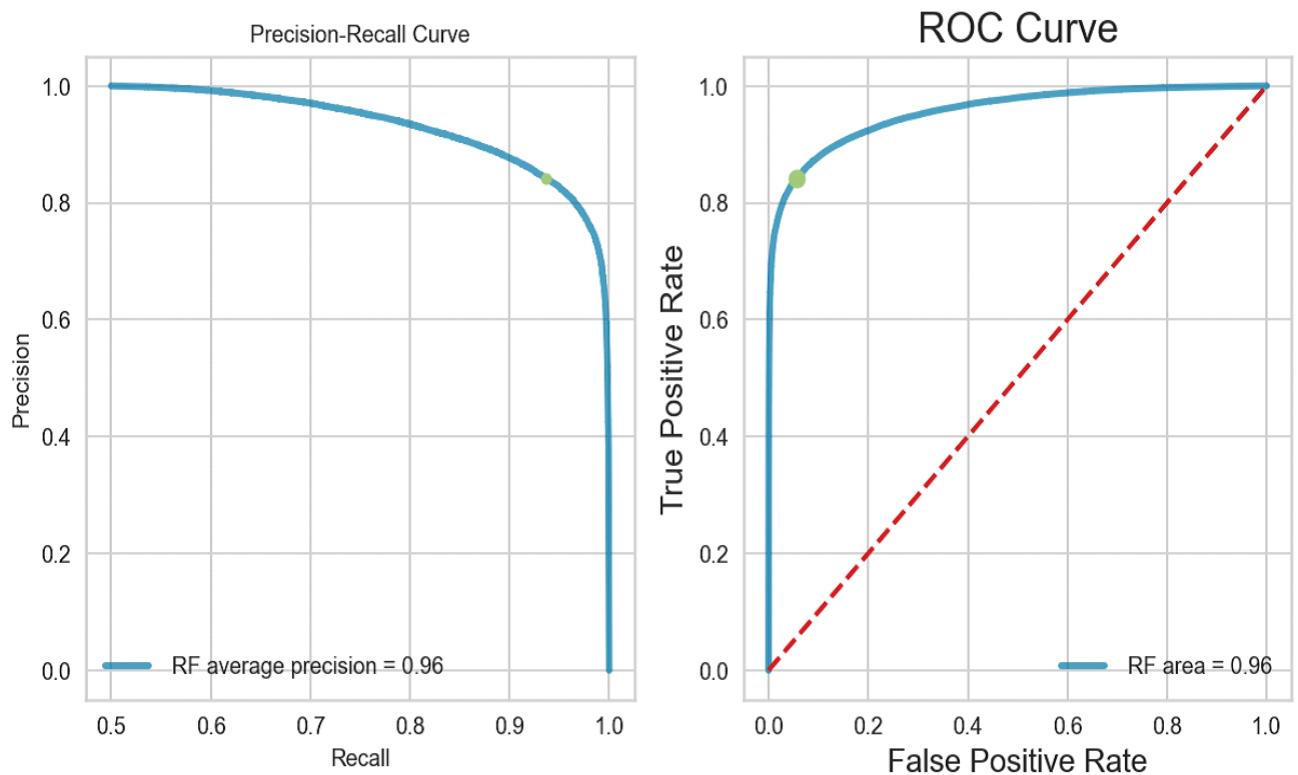
Applied Information Gain (IG) to rank feature importance for the classification task.

Standardized numerical features using z-score scaling.

#### 4/Dimensionality Reduction:

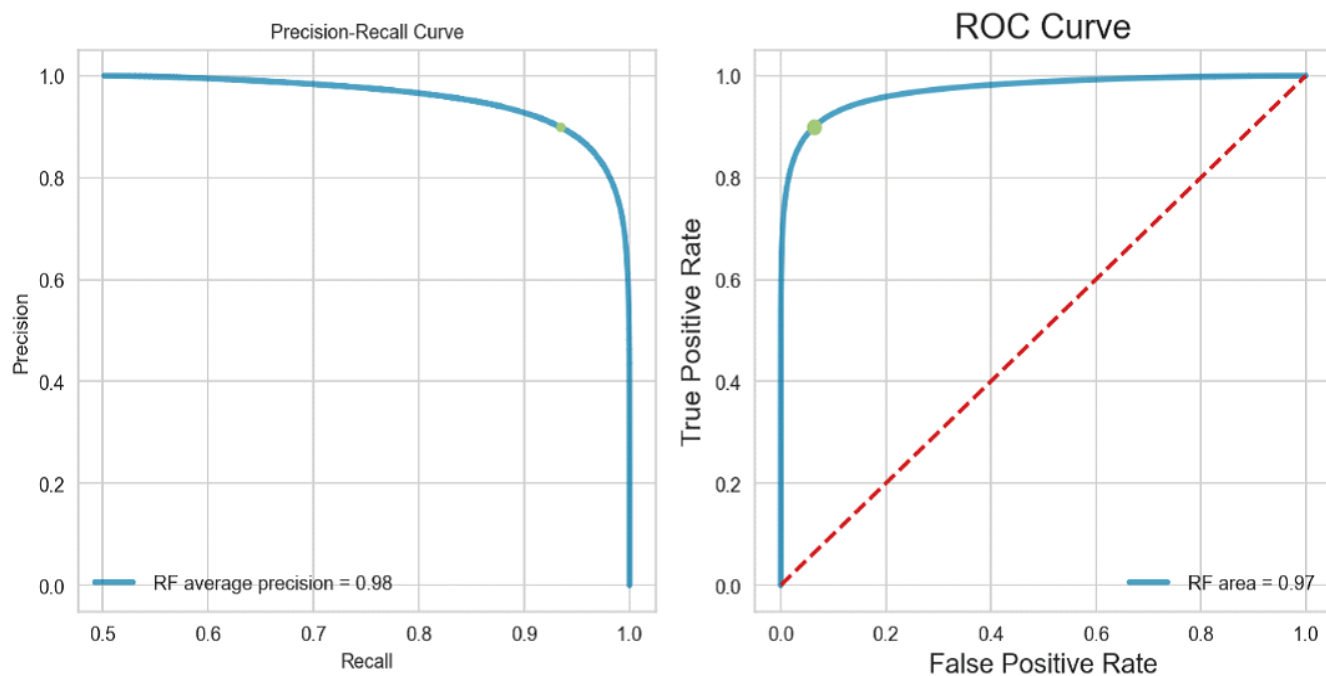
Used Principal Component Analysis (PCA) to reduce dataset dimensions, simplifying the feature space while preserving variance.

Equation: PCA compresses features into lower dimensions using eigenvalues.



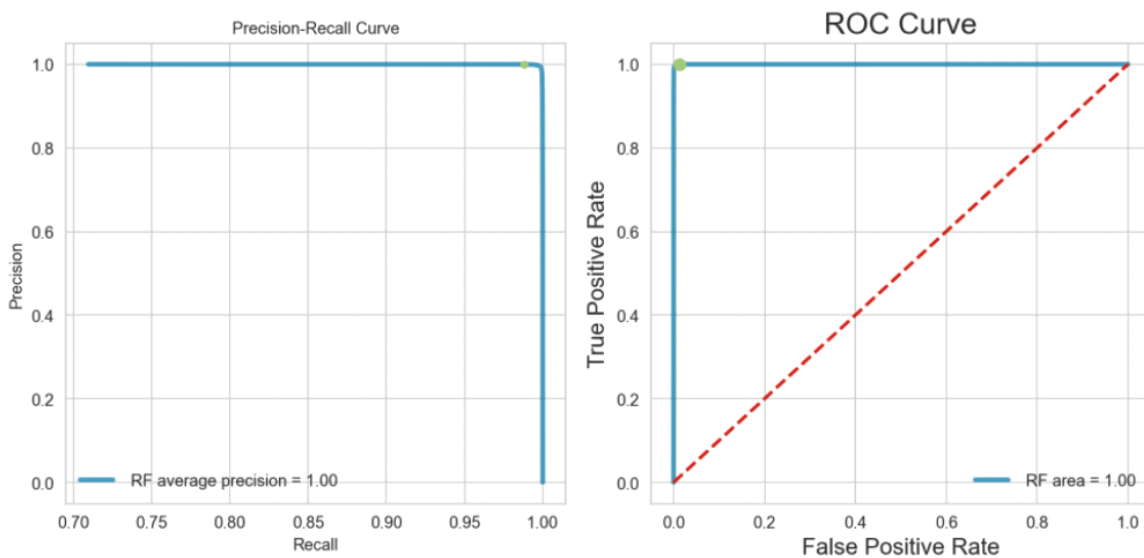
**Figure 21**

Performance plots of the RF classifier using the NearMiss undersampling.



**Figure 22**

The performance plots for the RF classifier using SMOTE oversampling.



**Figure 23**

The best performance plots for the RF classifier using Random oversampling technique.



Classification results using undersampling techniques.

Method Name	Classifier Name	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
NearMiss	LR	92.5	74.8	82.7	84.3
	NB	95.6	50.0	65.6	73.8
	RF	<b>93.8</b>	<b>83.8</b>	<b>88.6</b>	<b>89.2</b>
	XGBoost	94.1	82.1	87.7	88.5
Random undersampling	LR	59.0	58.9	59.0	59.0
	NB	55.3	68.8	59.9	55.1
	RF	<b>77.5</b>	<b>79.2</b>	<b>78.3</b>	<b>78.1</b>
	XGBoost	70.1	71.3	70.7	71.0

**Figure 24**

Classification results using oversampling techniques.

Resampling Technique	Classifier Name	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random oversampling	LR	59.0	59.0	59.0	59.0
	NB	51.8	91.9	59.0	59.0
	RF	<b>99.9</b>	<b>98.8</b>	<b>99.3</b>	<b>99.3</b>
	XGBoost	69.2	70.1	69.7	69.4
SMOTE	LR	69.5	47.6	56.5	63.3
	NB	97.6	22.6	36.7	61.8
	RF	<b>93.3</b>	<b>89.7</b>	<b>91.5</b>	<b>91.6</b>
	XGBoost	83.4	71.7	77.1	78.7
BorderLine SMOTE	LR	69.7	47.4	56.4	63.4
	NB	97.6	22.2	36.1	60.8
	RF	<b>93.2</b>	<b>90.8</b>	<b>92.0</b>	<b>92.1</b>
	XGBoost	83.1	74.9	78.8	79.8

**Figure 25**

## **Advantages of oversampling for machine learning:**

### **1/Does not decrease the size of the dataset:**

One advantage that oversampling has over other resampling schemes is that it does not decrease the size of the dataset. This can be important if you are using a complicated model with many parameters that need to be fit as it ensures that your model will have sufficient data to train on.

### **2/Does not lose any information:**

Another advantage of oversampling is that it ensures that you do not lose any information that is already present in the dataset. This is because all of the records that are contained in the initial dataset are also contained in the resampled dataset.

## **Disadvantages of oversampling for machine learning:**

Now we will talk more about the disadvantages of oversampling for machine learning. Here are some of the main disadvantages of oversampling for machine learning.

### **Can increase the chance of overfitting:**

One disadvantage of oversampling for machine learning is that it can increase the chance of your model overfitting to your training data. Whether you perform oversampling by introducing duplicated observations or synthetic observations that closely resemble existing observations, you are increasing the presence of characteristics in your dataset that were originally only applicable to a small number of observations. These characteristics might be specific details of a few specific observations rather than broadly applicable trends.

## **Advantages of undersampling for machine learning:**

### **No need to introduce redundant information into the dataset:**

The main advantage that undersampling has is that you do not have to add any artificial observations to your dataset that introduce repeated or redundant information into your dataset. Why is this beneficial? This is beneficial because when you duplicate existing observations (or create close analogs of existing observations), you are making it seem like the patterns that are seen in those observations are more widespread than they really are. This can lead to things like models overfitting to specific patterns that were only seen in a few observations in the original dataset.

## Disadvantages of undersampling for machine learning

### 1/Reduces the size of your dataset:

The first disadvantage of undersampling for machine learning is that it reduces the size of your dataset. Machine learning models generally perform better when they are trained on larger datasets with more observations, so this can have negative effects on the predictive performance of your model.

### 2/Loses information.

The next disadvantage of undersampling is that there is some loss of information. When you permanently remove observations from your dataset, you will naturally lose the information that was contained within those observations.

## Oversampling vs undersampling for machine learning

So when should you use oversampling for machine learning? And when would you be better off sticking with undersampling? In this section, we will provide guidelines you can follow to determine when to use oversampling or undersampling for machine learning.

**Oversample:** if you do not have a particular reason to undersample. The first guideline you can follow to determine whether to oversample or undersample your data is this – you should consider oversampling to be the default option and only reach for undersampling when you have a specific reason to do so. The decrease in data size and loss of data that you incur when you perform undersampling should be avoided when possible.

**Undersample:** when your data is very large. One example of a situation where you should spring for undersampling is when your data is very large. This is particularly true if you are going to need to reduce the size of your dataset anyways because training the data on the full dataset takes too much time or too many resources. In these situations, you are going to lose some data anyways and the decrease in the size of your dataset can actually be viewed as an advantage rather than a disadvantage.

**Undersample:** if your model is overfitting to oversampled data. Another situation where you might want to try out undersampling is if you have fit a model on oversampled data and you see that your model is overfitting to the training dataset. Oversampling is known to increase the chances of overfitting, so changing the resampling scheme that you are using may help to reduce the impact of overfitting.

## **Outlier Detection:**

### **Introduction:**

Outliers can significantly impact the performance of machine learning models, leading to inaccurate predictions and unreliable models. Detecting outliers is therefore a critical step in the data preprocessing pipeline. When dealing with outliers, we encounter a similar dilemma as with missing value imputation—outlier detection is highly dependent on both the context and the approach used. The definition of an outlier varies across different domains; what may be considered an anomaly in one dataset could be a meaningful data point in another. Additionally, the method chosen to handle outliers—whether statistical, machine learning-based, or hybrid—can significantly impact the results.

This variability makes relying on a single case study inadequate, as different perspectives exist on how outliers should be identified and treated. The effectiveness of an approach depends on factors like data distribution, domain-specific constraints, and the goals of the analysis. Therefore, multiple case studies are necessary to illustrate the diverse challenges and strategies for outlier detection, ensuring a more comprehensive understanding. Since this topic is big and the size of the project is limited, we will explore only few famous techniques in an article format.

**Now we will discuss some famous methods for dealing with outliers using the same fraud detection datasets used in the case study above:**

#### **1/Tukey's (1977):**

Technique is used to detect outliers in skewed or non-bell-shaped data since it makes no distributional assumptions. However, Tukey's method may not be appropriate for a small sample size. The general rule is that anything not in the range of  $(Q1 - 1.5 \text{ IQR})$  and  $(Q3 + 1.5 \text{ IQR})$  is an outlier, and can be removed.

**Inter Quartile Range (IQR) is one of the most extensively used procedure for outlier detection and removal.**

**Procedure:**

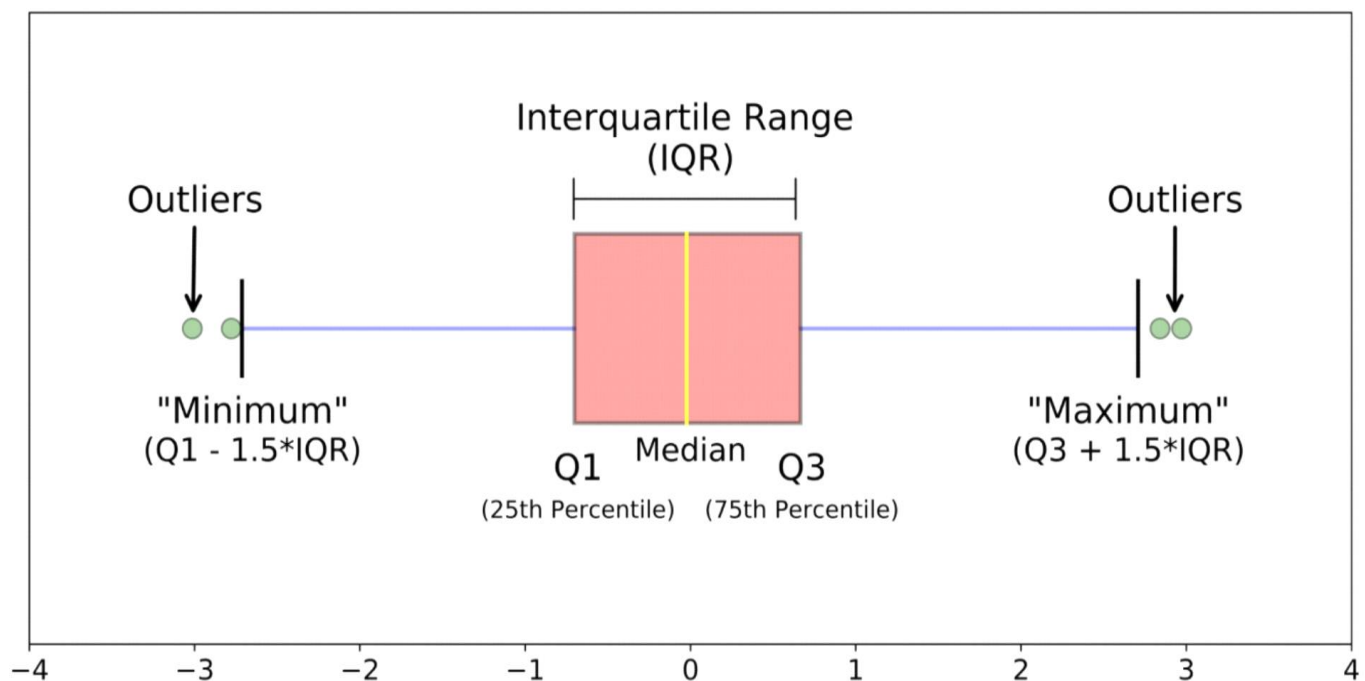
1/Find the first quartile, Q1.

2/Find the third quartile, Q3.

3/Calculate the IQR.  $IQR = Q3 - Q1$ .

4/Define the normal data range with lower limit as  $Q1 - 1.5 \text{ IQR}$  and upper limit as  $Q3 + 1.5 \text{ IQR}$ .

Any data point outside this range is considered as outlier and should be removed for further analysis.

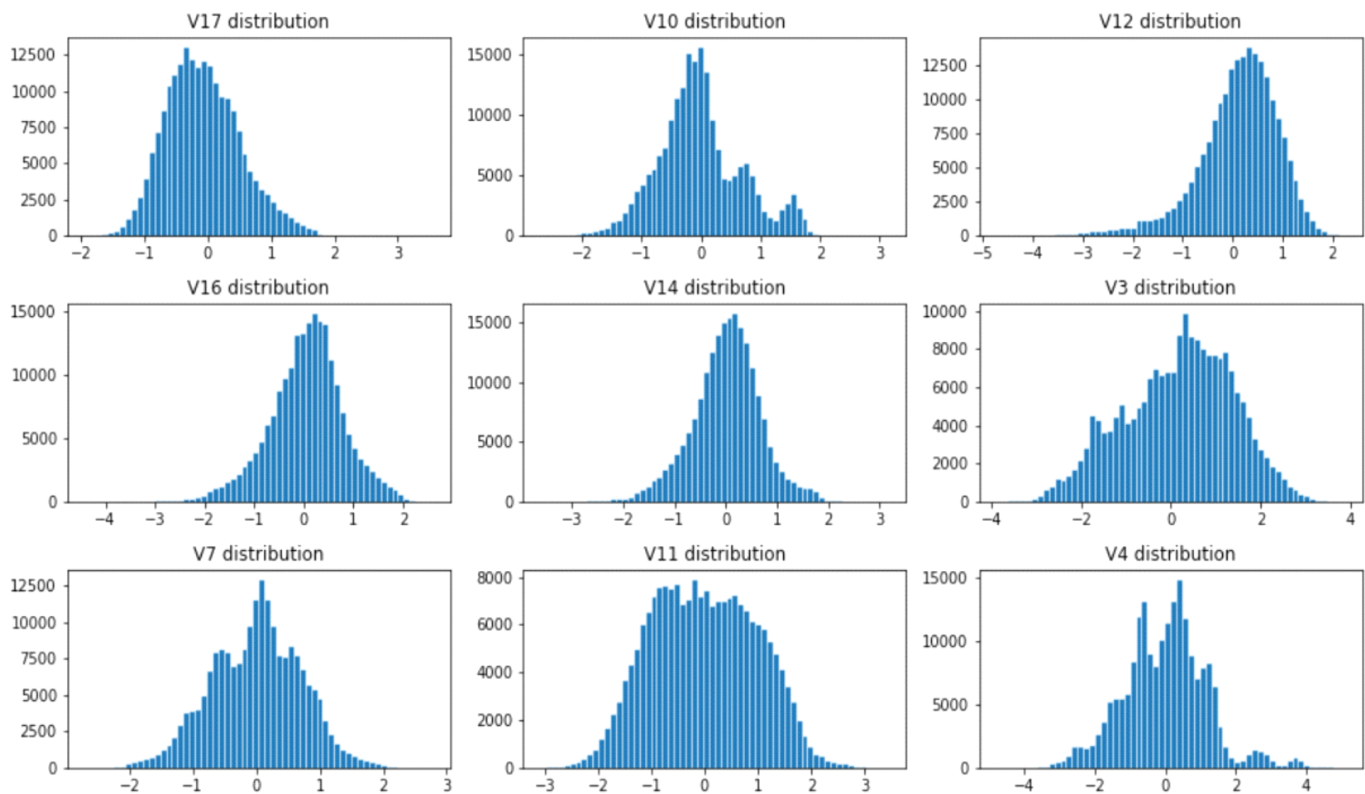


**Figure 26**

The figure show the tukey method

**After using the tukey methods on the Fraud Detection using the tukey methods on the datasets we find the number of outliers =Total number of outliers is: 31904**

### Distributions of most important features after dropping outliers using IQR Method



**Figure 27**

### 2/Standard deviation method:

If we know that the distribution of values in the sample is Gaussian or Gaussian-like, we can use the standard deviation of the sample as a cut-off for identifying outliers.

Standard deviation shows how much the individual data points are spread out from the mean. If a data distribution is normal then:

68% of the data values lie within one standard deviation of the mean

95% are within two standard deviations

99.7% lie within three standard deviations.

Depending on the set specification either at 2 times stdev or 3 times stdev, we

can detect and remove outliers from the dataset.

This method can fail to detect outliers because the outliers increase the standard deviation. The more extreme the outlier, the more the standard deviation is affected.

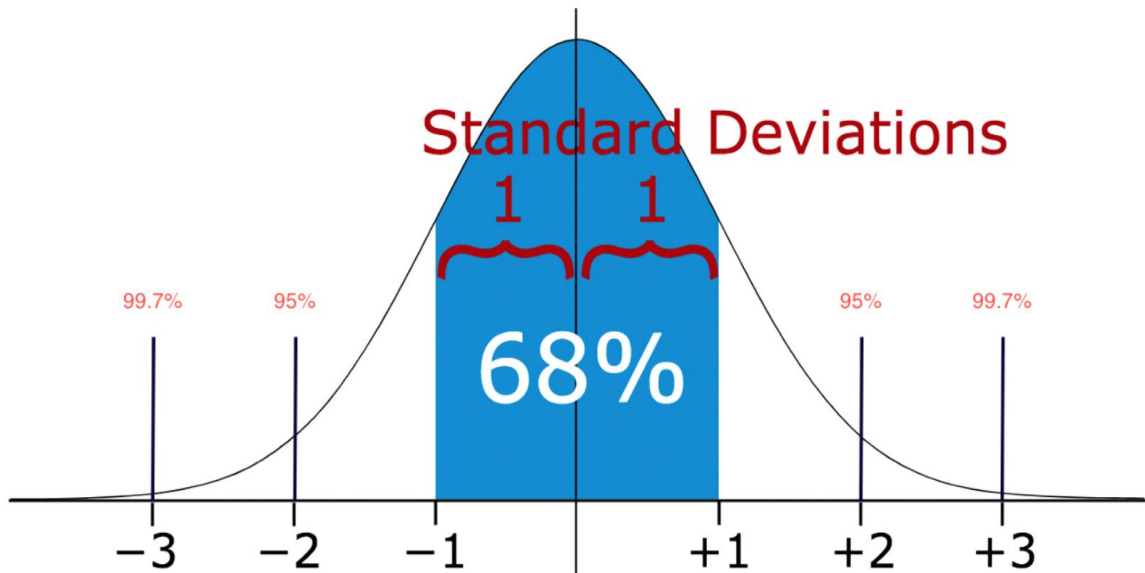


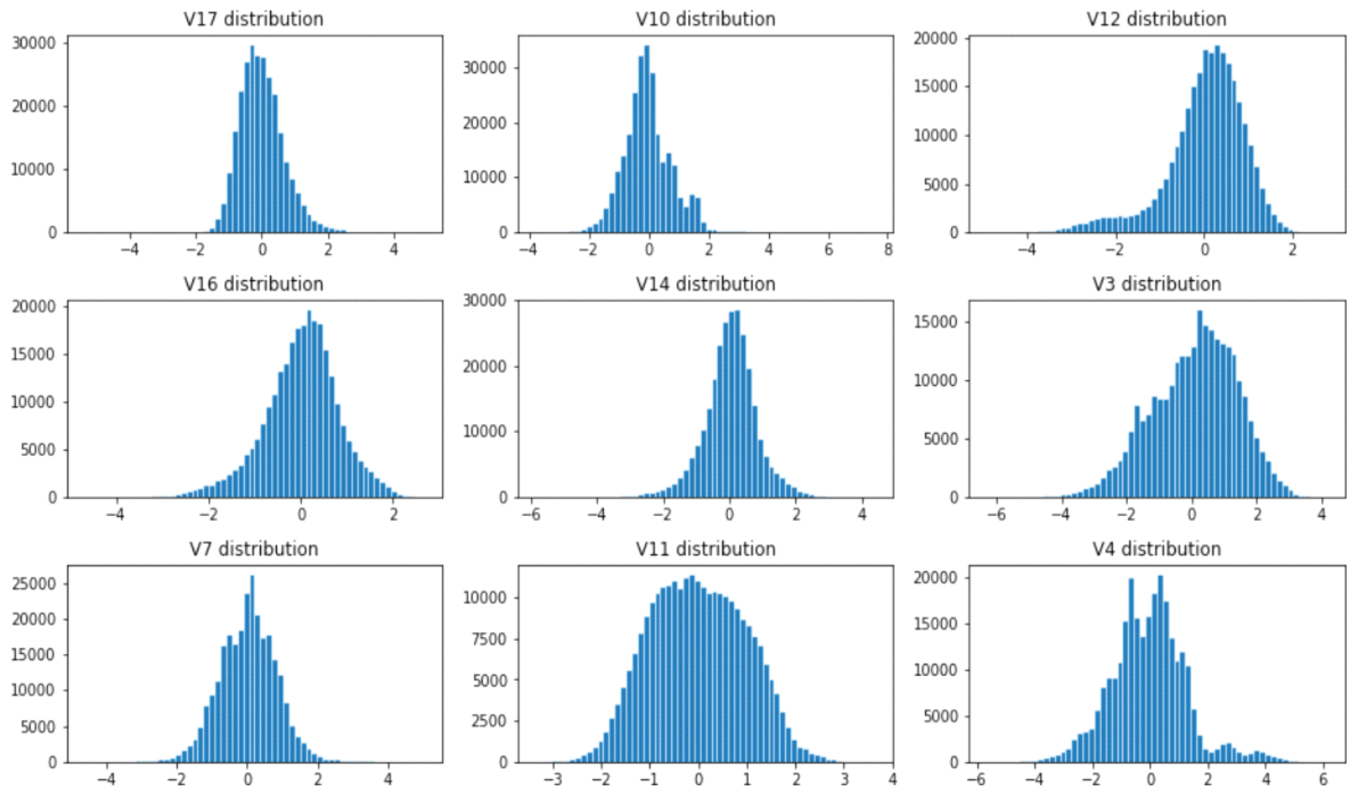
Figure 28

**Result:**

After using this method

**Total number of outliers is: 4076**

### Distributions of most important features after dropping outliers using Standard Deviation Method



**Figure 29**

### 3/Z-score method:

While calculating the Z-score we re-scale and center the data and look for data points which are too far from zero. Z-score is used to convert the data into another dataset with mean = 0. Z-score describes the position of a raw score in terms of its distance from the mean, when measured in standard deviation units. This technique assumes a Gaussian distribution of the data. The outliers are the data points that are in the tails of the distribution.



## Detecting Outliers with z-Scores

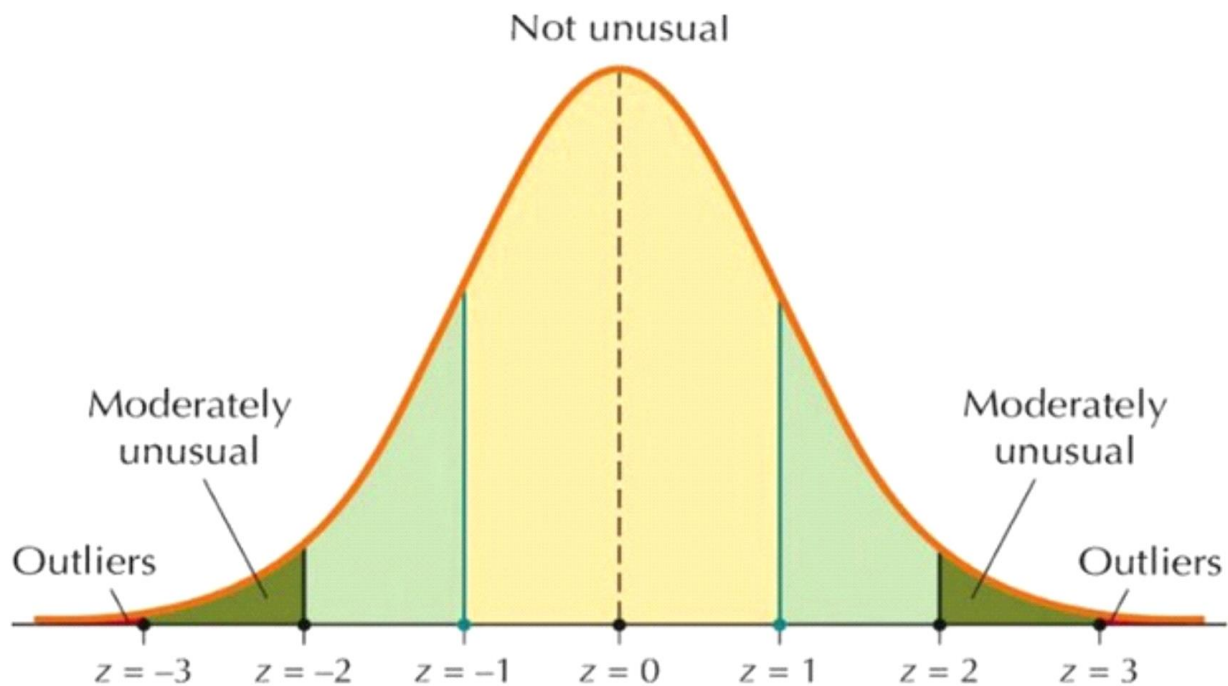


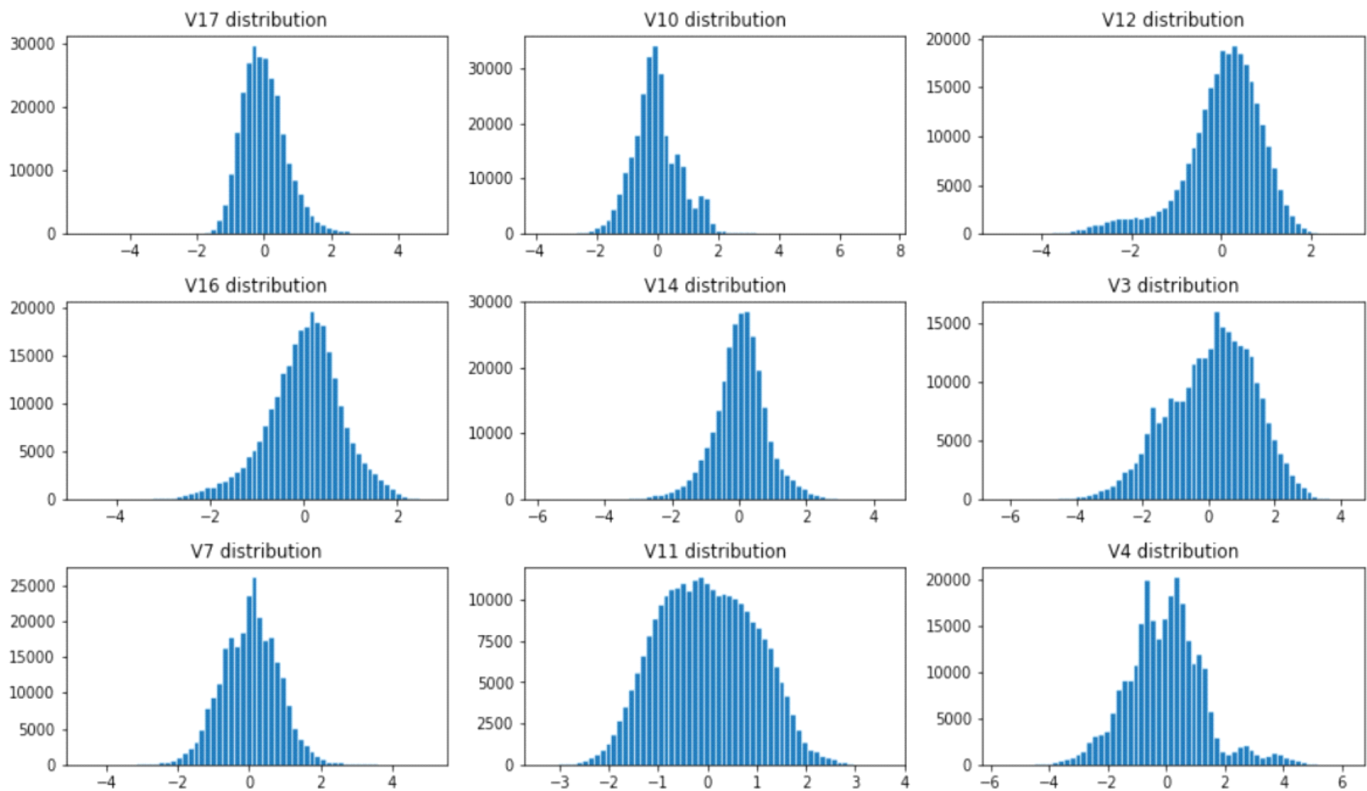
Figure 30

Using z score method on Fraud Detection

**Result:**

**Total number of outliers is: 4076**

### Distributions of most important features after dropping outliers using z-score



**Figure 31**

### Impact on Model Performance:

Outliers can have both positive and negative impacts on model performance, depending on the context and the method used to handle them.

#### 1. Positive Impacts:

##### A/ Improved Accuracy:

Outliers can distort the decision boundaries of machine learning models, leading to poor accuracy. By removing or appropriately handling outliers, models can better generalize to unseen data, thereby improving accuracy.

Example: In linear regression, eliminating outliers can result in a model with a lower mean squared error, leading to more accurate predictions.

## **B/Enhanced Precision and Recall:**

Outlier handling can refine the prediction boundaries, improving precision (the ratio of true positives to all positive predictions) and recall (the ratio of true positives to all actual positives).

Example: In a fraud detection model, properly managing outliers ensures that genuine cases of fraud are identified (high recall) without misclassifying non-fraud cases (high precision).

## **2. Negative Impacts:**

### **A/ Loss of Important Information:**

Outliers sometimes represent critical information. For instance, outliers in financial data could indicate rare but significant events, such as fraud or market crashes. Removing these outliers might lead to a model that misses these important occurrences.

Example: In financial trading, outliers may indicate a sudden market change, and removing them could lead to a model that fails to predict such events, decreasing its practical usefulness.

### **B/ Bias Introduction:**

Overzealous outlier removal can introduce bias, particularly if outliers are not randomly distributed but are instead concentrated within certain groups or conditions. This can lead to a model that is less fair or less applicable to diverse scenarios.

Example: In medical research, outliers might represent patients with rare diseases. Removing these data points could bias the model against recognizing these diseases.

### **C/Decreased Model Robustness:**

While removing outliers can improve model accuracy on a training dataset, it may also reduce the robustness of the model when applied to new, unseen data that may contain similar outliers. This can result in poor generalization.

Example: A credit scoring model that has outliers removed might perform well on historical data but poorly on new applications where such outliers are more prevalent.

## Future Trends and Research Directions:

### Emerging Techniques:

The field of outlier detection is rapidly evolving, with new techniques being developed to address the limitations of current methods. Advances in deep learning, particularly with architectures like GANs (Generative Adversarial Networks) and Variational Autoencoders (VAEs), are leading to more sophisticated approaches for identifying complex outliers in high-dimensional data. Additionally, hybrid methods that combine statistical and machine learning techniques are gaining traction, offering more robust and flexible solutions.

### Challenges and Open Problems:

Despite significant progress, challenges remain in outlier detection and handling. One major issue is scalability — many advanced techniques struggle with very large datasets. Another challenge is the interpretability of complex models, particularly in deep learning, where the decision-making process can be opaque. Ongoing research is focused on addressing these challenges, with the goal of developing methods that are both scalable and interpretable.

### Trade-Offs in Different Approaches:

**Choosing an outlier handling technique involves balancing simplicity, interpretability, and effectiveness. Here's a look at the trade-offs associated with different methods.**

#### 1. Simplicity vs. Effectiveness:

##### 1. Simple Techniques (e.g., Z-Score, IQR Method):

**Advantages:** Simple to implement and interpret, requiring minimal computational resources. These methods are widely understood and can be easily explained to stakeholders, making them suitable for quick diagnostics or when working with small to moderately sized datasets.

**Disadvantages:** These methods may not be effective in capturing complex outliers in high-dimensional data or datasets with non-normal distributions. They can also be less flexible, potentially missing context-specific outliers or incorrectly labeling non-outliers as outliers.

## **2. Complex Techniques (e.g., Isolation Forest, Autoencoders):**

**Advantages:** More effective at identifying outliers in complex, high-dimensional datasets. These methods can capture non-linear relationships and interactions between variables, which simpler methods might miss.

**Disadvantages:** Increased computational cost and complexity. These methods can be harder to interpret, making it challenging to explain to non-technical stakeholders why certain data points are considered outliers. Additionally, they often require tuning of hyper parameters and may not be as intuitive to apply.

## **Conclusion and Recommendations:**

In this final section, we present our findings and recommendations based on the case study examined throughout the previous chapter. Our analysis provides valuable insights and practical guidance for future endeavors in this field.

**In handling missing values, which is one of the hottest topics in data cleaning, we have not addressed the most important question: how to choose the appropriate imputation techniques?**

### **Conclusion and takeaway:**

The need for missing value imputation (MVI) depends on how crucial the missing data is. Some modern classification and regression models can handle incomplete datasets without requiring MVI, as they internally manage missing values. During the early stages of model development, researchers can compare results from these models using both incomplete data and data imputed with recommended baseline MVI methods. This helps determine whether missing values significantly impact the results and if imputation is necessary.

One key takeaway from this review is that there is no universally "best" imputation method. The optimal approach depends on various factors, such as dataset characteristics, feature selection, outlier detection, normalization techniques, the type of classifier used, the specific application domain, computational resources, time constraints, and whether the method prioritizes maintaining mean values or preserving relationships between attributes. Researchers must carefully select an MVI technique that aligns with these factors.

Additionally, selecting relevant features or samples before or after applying MVI can influence the quality of imputed data. Removing unrepresentative attributes or samples before imputation can result in a cleaner, more reliable dataset for training, potentially leading to better imputation accuracy. On the other hand, performing attribute or sample selection after imputation can enhance the effectiveness of the classifier by refining the dataset further.

## **Bibliography:**

- García, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Springer. <https://link.springer.com>
- Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Learning. International Journal of Computer Science, 1(2), 111- 117. <https://www.researchgate.net>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over Sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. <https://doi.org/10.48550/arXiv.1106.1813> DOI: <https://doi.org/10.1613/jair.953>
- A Review of Missing Values Handling Methods on Time-Series Data Irfan Pratama<sup>1</sup>, Adhistya Erna Permanasari<sup>2</sup>, Igi Ardiyanto<sup>3</sup>, Rini Indrayani<sup>4</sup> Department of Electrical Engineering and Information Technology Universitas Gadjah Mada 2016 International Conference on Information Technology Systems and Innovation (ICITSI) Bandung – Bali, October 24 – 27, 2016 ISBN: 978-1-5090-2449-0. <https://ieeexplore.ieee.org>
- D. Little, RJA and Rublin, “Statistical Analysis with Missing Data,” Wiley, New York. p. 381, 1987. <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119482260?msockid=10060b221ab467bd25d31e401ba06671>
- I. B. Aydilek and A. Arslan, “A hybrid method for imputation of missing values using optimized fuzzy c- means with support vector regression and a genetic algorithm,” Inf. Sci. (Ny)., vol. 233, pp. 25 35, 2013. <https://doi.org/10.1016/j.ins.2013.01.021> <https://www.sciencedirect.com>
- A. T. Sree Dhevi, “Imputing missing values using Inverse Distance Weighted Interpolation for time series data,” 6th Int. Conf. Adv. Comput. IcoAC 2014, pp. 255– 259, 2015. <https://doi.org/10.1109/ICoAC.2014.7229721> <https://ieeexplore.ieee.org>
- K. Strike, K. El Emam, and N. Madhavji, “Software Cost Estimation with Incomplete Data,” IEEE Trans. Softw. Eng., vol. 27, no. 10, pp. 890–908, 2001. <https://doi.org/10.1109/32.962560> <https://ieeexplore.ieee.org>
- Python for data analysis by Wes McKinney Orielly 3rd edition (2022). <https://www.oreilly.com>
- Hands on machine learning with sickit –learn keras and tensorflow orielly 3rd edition (2022) Aurélien Géron. <https://www.oreilly.com> . Outlier Analysis. Springer. <https://link.springer.com>
- Barnett, V., & Lewis, T. (1994). Outliers in Statistical Data. Wiley. <https://doi.org/10.1002/bimj.4710370219>
- Kim, H.-Y., & Kim, Y. (2016). Outliers in Data: A Z-Score Perspective. Korean Journal of Anaesthesiology.

- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *Proceedings of the 8th IEEE International Conference on Data Mining*. <https://doi.org/10.1109/ICDM.2008.17>
- Hodge, V. J., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*. <https://link.springer.com>
- Osborne, J. W., & Overbay, A. (2004). The Power of Outliers (and Why Researchers Should Always Check for Them). *Practical Assessment, Research & Evaluation*. <http://pareonline.net>
- Little, R. J. A., & Rubin, D. B. (2019). *Statistical Analysis with Missing Data*. Wiley.
- Hawkins, D. M. (1980). *Identification of Outliers*. Chapman and Hall. <https://link.springer.com>
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann. . <https://doi.org/10.1016/C2009-0-61819-5>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. . <https://link.springer.com>
- Sammut, C., & Webb, G. I. (2010). *Encyclopaedia of Machine Learning*. Springer. <https://link.springer.com>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer. <https://link.springer.com>
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer. <https://link.springer.com>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. <https://link.springer.com>
- Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. <https://doi.org/10.1109/5.726791>
- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the International Conference on Machine Learning (ICML)*. <https://doi.org/10.48550/arXiv.1502.03167>
- Zhang, Z., & Ma, L. (2012). Comparison of Data Normalization Methods for Principal Component Analysis. *Applied Mechanics and Material*.
- Nilashi M, Ahmadi H, Manaf AA, Rashid TA, Samad S, Shahmoradi L, Aljojo N, Akbari E. Coronary heart disease diagnosis through self-organizing map and fuzzy support vector machine with incremental updates. *Int J Fuzzy Syst* 2020;22:1376–88.
- Khennou F, Fahim C, Chaoui H, Chaoui NEH. A machine learning approach: Using predictive analytics to identify and analyze high risks patients with heart disease. *Int J Mach Learn Comput* 2019;9:762–7.
- Setiawan N, Venkatachalam P, Hani A. Missing data estimation on heart disease using artificial neural network and rough set theory. In: 2007 international



conference on intelligent and advanced systems. 2007, p. 129–33.

- Saini M, Baliyan N, Bassi V. Prediction of heart disease severity with hybrid data mining. In: 2017 2nd international conference on telecommunication and networks. IEEE; 2017, p. 1–6.
- Rani P, Kumar R, Ahmed NMS, Jain A. A decision support system for heart disease prediction based upon machine learning. J Reliable Intell Environ 2021;7:263—275
- Hasan MK, Alam MA, Das D, Hossain E, Hasan M. Diabetes prediction using ensembling of different machine learning classifiers. IEEE Access 2020;8:76516–31
- Wang Q, Cao W, Guo J, Ren J, Cheng Y, Davis DN. DMP\_MI: an effective diabetes mellitus classification algorithm on imbalanced data with missing values. IEEE Access 2019;7:102232–8.
- Christobel A, SivaPrakasam P. The negative impact of missing value imputation in classification of diabetes dataset and solution for improvement. IOSR J Comput Eng 2012;7.
- Maniruzzaman M, Rahman MJ, Al-Mehedi Hasan M, Suri HS, Abedin MM, El-Baz A, Suri JS. Accurate diabetes risk stratification using machine learning: role of missing value and outliers. J Med Syst 2018;42:1–17
- Kandhasamy JP, Balamurali S. Performance analysis of classifier models to predict diabetes mellitus. Procedia Comput Sci 2015;47:45–51.