Application of Machine Learning and Data Mining with Python for Business Analysis

Abstract

Organizations need to use business data to analsysis with different kinds of techniques and tools in order to analysis their business data. In this context, implementing machine learning (ML) and data mining (DM) analysis techniques and tools could be very helpful for dealing with business challenges.

Therefore, in this master's thesis, a comprehensive literature review is summarized to providing an overview of how machine learning and data mining techniques can be used to solve business issues with analytical actions. In this section briefly described data mining stanard processes, data mining architecture and applications that has been widely used in the business field, such as fraud detection, target marketing, and safe online transactions. There are various types of machine learning algorithms mentioned in the field, including supervised, unsupervised, semi-supervised, and reinforcement learning. And also, how machine learning algorithms are applied to the datasets and are evaluated using performance metrics.

Use the Python programming language and its standard libraries, such as Numpy, Scipy, Matplotlib, and Scikit-Learn, as an analytical tools for the practical section. These open-source Python programs are capable of performing data preparation, statistical analysis, data insight visualization, ML modeling and evaluation in order to discover hidden patterns and significant features in the dataset to compete prediction model.

Keywords: Data mining, Machine learning, Python, Data analysis, Statistics, Data warehosue, Python library, Data visualization, Algorithms.

Aplikace strojového učení a dolování dat s Pythonem pro obchodní analýzu

Abstrakt:

Organizace potřebují používat obchodní data k analýze pomocí různých druhů technik a nástrojů, aby mohly analyzovat svá obchodní data. V této souvislosti může být implementace analytických technik a nástrojů strojového učení (ML) a data miningu (DM) velmi užitečná pro řešení obchodních problémů.

Proto je v této diplomové práci shrnut komplexní přehled literatury, který poskytuje přehled o tom, jak lze techniky strojového učení a dolování dat použít k řešení obchodních problémů pomocí analytických akcí. V této části jsou stručně popsány standardní procesy dolování dat, architektura dolování dat a aplikace, které byly široce používány v obchodní oblasti, jako je detekce podvodů, cílený marketing a bezpečné online transakce. V této oblasti jsou zmíněny různé typy algoritmů strojového učení, včetně řízeného, nekontrolovaného, částečně řízeného a zesíleného učení. A také, jak jsou algoritmy strojového učení aplikovány na datové sady a jak jsou vyhodnocovány pomocí výkonnostních metrik.

V práci je použit programovací jazyk Python a jeho standardní knihovny, jako jsou Numpy, Scipy, Matplotlib a Scikit-Learn, jako analytické nástroje pro praktickou část. Tyto open-source programy Python jsou schopny provádět přípravu dat, statistickou analýzu, vizualizaci datového náhledu, modelování a vyhodnocování ML za účelem odhalení skrytých vzorců a významných funkcí v datové sadě, aby mohly konkurovat predikčnímu modelu.

Klíčová slova: Dolování dat, Strojové učení, Python, Analýza dat, Statistika, Datový sklad, Knihovna Pythonu, Vizualizace dat, Algoritmy.

Table of Content

1. Introduction

Businesses gather massive amounts of information on their customers, including their past purchases, responses to marketing campaigns, web search history, etc. Data may help us better understand our clients and make more well-informed business decisions in today's data-driven market.

Our generation became known as the "age of information" or "age of data"(Sarkar, Bali, Sharma 2018) as computing power and storage capacity improved. Furthermore, we must analyze huge data and build intelligent systems using concepts and techniques from artificial intelligence, data science, data mining, and machine learning. The most significant task that organizations and enterprises have undertaken in the last decade to utilize their data and comprehend and utilize this information is to make more informed decisions. Indeed, as technology has advanced, a successful ecosystem has emerged around subjects such as data mining, machine learning and so on.

Researchers, engineers, and data scientists have developed frameworks, tools, techniques, algorithms, and approaches to develop intelligent systems and models capable of automating tasks, detecting anomalies, performing sophisticated analyses, and predicting outcomes.

In generally data mining is concerned with the finding of knowledge from data. Machine learning focuses on prediction through training and learning. Many machine learning methods are used in data mining; machine learning also employs data mining methods as pre-processing for greater learning and accuracy.

In this thesis, illustrate the most widely used data mining standard tasks, and approaches in business applications. As part of my practical, tried to visualize the data, important hidden patterns in business data. Machine learning algorithms, techniques, evaluate the process of model implementation and accuracy, which may subsequently be used to generate predictions.

In the practical section, to generate the process of loan status using python tools, supplied a dataset to identify the customers segments that are suitable for loan grant, build a model by analyze the raw dataset and the changes it gradually in order to get it ready for machine learning model training. This model will be created using data mining task and machine

learning algorithms. This approach to help to finance company to determine whether applicants grant for loan using previous historical data.

1.1 Objectives

The primary concern of this master's thesis is to define the best way to apply machine learning models and data mining techniques in a business perspective. The following are some of the objectives of this thesis,

- To enhance decision making, business data should be analyzed and visualized using data mining techniques.
- To observe different technologies for machine learning and data mining applications.
- To evaluate the implementation of a machine learning algorithm for data analysis and business problem solving.

1.2 Methodology

To achieve the goals, a review the related literature, scientific papers, and internet resources on data mining and machine learning background in business prospects. This diploma thesis gathered all associated knowledge regarding the data mining standard process model CRISM-DM(refer to page 13), SEMMA(refer to page 10) and data warehousing system, which is very essential for understanding data mining use cases in any business context. Here also describe the most suitable approach to implementing data mining tasks in business applications.

On the other side review the machine learning algorithms, tasks and current business applications are related to the any business operation. Initially, studied business knowledge in a finance company that provides loans to applicants and then evaluates the customer's loan eligibility using historical data. In such case, data mining tasks will be implemented, such as understanding important insights of data through statistical analysis, visualize data, correlation between the target variable and categorical and numerical columns in the dataset.

Second, we'll require data to train our model for future predictions. The data for this thesis will be obtained straight from the Kaggle dataset. This information will only be used for the purposes of this thesis research. For our entire assignment, we will primarily employ Jupyter notebook as a tool to implement the Python programming language, as well as the Python library and the Windows OS.

Python packages such as NumPy and Pandas will be used for data cleansing. These libraries are quite useful for working with numeric numbers and data frames. We will plot graphs for data visualization using the Matplotlib and Plotly python packages. After cleaning our data, we will find outliers, create a heatmap to show the relationship between variables, and prepare it for machine learning modeling.

For machine learning modeling, all categorical columns must be encoded before splitting the target column from the dataset and re-sampling the unbalanced columns using a statistical technique. Scikit-learn is used to separate train and test data for the execution of machine learning algorithms. Cross validation and a confusion matrix will be used to choose the best scoring model and its parameters from two machine learning algorithms: linear regression and gradient boosting.

2. Literature Review

Machine learning and data mining are two distinct subfields of big data and artificial intelligence. They combine in business intelligence protocols and strengthen one another. Despite having many things in common, they each reach different conclusions. Data mining, commonly referred to as "knowledge discovery in databases," is the technique of finding interesting patterns in databases which can be used to make decisions and businesses forecast future trends, making decision-making easier and improving consumer experience. Data mining is a developing interest and important discipline, as well as an application field that may give a major competitive advantage to a business by utilizing the resources of big data warehouses.[1]

According to Doug Alexander of the University of Texas, data mining involves considerably more than just data analysis. Data mining is the computer-assisted process of sifting through and analyzing massive amounts of data in order to extract its significance. Data mining technologies reveal predicted information that experts would overlook because it goes against their expectations. By using these tools, firms are able to make proactive, knowledge-driven decisions about economics and business issues that would otherwise take a lot of time to resolve.[5]

Data mining consists of five major elements

- Extract, transform and load transaction data onto the data warehouse system
- Store and manage the data in a multi-dimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software
- Present the data in useful format such as a graph or a table.

2.1 Data Mining Standard Processes

2.2 Knowledge Discovery in Databases

When the concept "knowledge discovery from data" (KDD) was first used in the early 1990s (Piatetsky-Shapiro, 1991), there was a rush to create data mining algorithms that could address every issue associated with finding relevant information in massive amounts of data,[8]. Knowledge Discovery in Databases (KDD) is a method for automatically analyzing and modeling huge data sources. KDD is the systematic process of discovering valid, unique, valuable, and intelligible patterns in vast and big data sets. The basis of the KDD process is data mining (DM), which involves the inference of algorithms that examine the data, create the model, and identify hidden patterns in data. The model is used for data analysis, prediction, and understanding of facts.[3]

The discipline of KDD is focused with the creation of tools and processes for interpreting data. The fundamental issue that the KDD process attempts to solve is the modelling of low-level data—which are frequently too large to comprehend and digest easily—into other aspects that may be more condensed such as a brief report, more abstract (such as a descriptive estimation

or model of the process that generated the data), or more beneficial (for example, as in a prediction model for calculating the value of future events). The implementation of particular data-mining techniques for pattern extraction and discovery forms the basis of the procedure.[4]

KDD Process Steps: The number of distinct steps in the KDD process is typically interpreted differently.

Selection [6]

- The targeted data is selected in the first stage, together with the variables that will be used to evaluate the success of the knowledge discovery, from a database of compiled data.

Pre-processing [6]

- It's all about improving the data that's being worked with.
- Improving data includes the concept of data cleansing.
- The preparation procedure improves the quality of the data set by inserting missing attributes, eliminating duplicate instances, and solving data discrepancies.

Transformation [6]

- This stage focuses on transforming preprocessed data to fully used will type.
- By reducing the scope in terms of diversity, and data properties are securely established for future review.
- The data is structured and sorted.

Data Mining [3]

- Used to extract potentially useful patterns.
- These patterns are graphed, trended, and visualized in a format that is uniquely useful to the process for which KDD is being implemented.

  This step involves –

  1. Task-relevant data is transformed into a pattern.

2. Determines the objective of the model through classification and characterization.

This step's method incorporates grouping, clustering, and regression, with the chosen one depending on the expected and intended outcome of the procedure.

Interpretation / Evaluation [6]

- In the last phases, the data is passed on for interpretation and documentation.
- The data has been cleaned, transformed, separated based on key features, then framed into visual representations

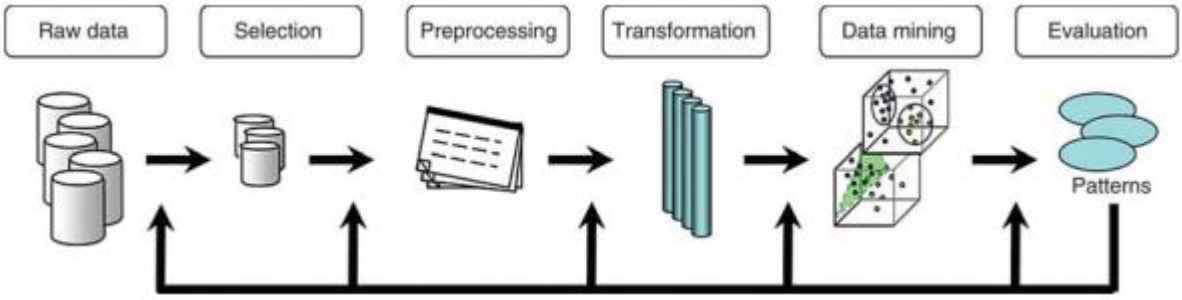


Figure 1 KDD Process Steps Pipeline.[6]

Use Cases and Advantage of KDD Process: KDD is a powerful tool for assisting businesses and companies in staying current with consumer wants, behaviors, and actions. There are some advantages of using the KDD process and it also has some challenges involved in it.

Market Forecasting

Database marketing systems, which examine client databases to identify various consumer categories and predict their behavior, are the main use in marketing. Over half of all merchants, according to Business Week (Berry 1994), are either utilizing or planning to employ database marketing, and those who do so claim positive outcomes. For instance, American Express estimates a 10 to 15% rise in credit card use. Market-basket analysis (Agrawal et al. 1996) systems are another significant marketing tool that uncovers trends such "If client buys X, he/she is also likely to buy Y and Z." Retailers benefit from such patterns.[4]

Iterative Process

The KDD method is iterative, which means that new knowledge is continually incorporated into it to improve its effectiveness. In this approach, formally obtained and previously undiscovered information is used to improve the data's refinement at each level (knowledge). As a result, a loop is created that feeds back into the setting of objectives after the ultimate outcome has been implemented.[7]

Anomaly Identification

The more we know about process flaws or security vulnerabilities, the more we can protect ourselves against them, leveraging their expertise to improve process efficiency and security and aiding future utilization development.[7]

2.3 SEMMA

SEMMA is defined by SAS Institute as a logical arrangement of the functional tool set of SAS enterprise miner for carrying out the basic data mining operations (SAS Institute, 2005). The word SEMMA is short for "*sample, explore, modify, model, and assess"*. Using a simple representative sample of data, SEMMA aims to simplify things. pick and use appropriate exploratory statistics and visualization methods. Model the variables to change the most important predictive factors, forecast results, then verify the correctness of a model. Business miner can be utilized as part of the client's iterative data mining technique. SEMMA is primarily concerned with the model development elements of data mining. [8, 9]

The key difference between the original KDD process and SEMMA is that SEMMA is embedded within SAS products such as Enterprise Miner and it is unlikely to utilize SEMMA technique outside of them, whereas KDD is an open process that can be implemented in a variety of situations.[8]

SEMMA Process Steps,

Step 1 Sample

This stage comprises selecting an acceptable volume dataset subset from a sizable dataset that has been provided for the model's creation. Finding variables or factors that are impacting the

process is the aim of the first stage of the process. After that, categories for preparation and validation are created from the obtained data.[10]

Step2 Explore [9]

In order to obtain a better understanding of the dataset, the user searches for unexpected trends and anomalies. Following the sample of the desired data, the next stage is to visually or quantitatively analyze it for underlying trends or groups. Exploratory assists in the improvement and modification of the discovery process.

If visual analysis does not reveal any apparent trends, statistical techniques such as component analysis, correspondence analysis, and clustering can be used to investigate the data. For example, in data mining for a direct mail campaign, clustering may uncover groups of clients with unique ordering habits. Limiting the discovery process to each of these unique groups independently may enhance the possibility of finding deeper patterns that may not be strong enough to be noticed if the entire dataset is analyzed together.

Step 3 Modify [9]

The user generates, chooses, and alters the variables that will be used to concentrate the model-building process. Based on the discoveries made during the research phase, it may be necessary to change data in order to add information, such as customer grouping and important subgroups, or to include new variables.

In order to focus on the most important factors, it could also be required to exclude non-significant variables and search for outliers. When "mined" data change, one could also need to edit data. Since data mining is a dynamic, iterative process, it is possible to alter the data mining techniques or models when fresh data becomes available.

Step 4 Model [9]

The user looks for a set of variables that may accurately anticipate the desired result. After data preparation, models that explain data patterns are ready to be built. Artificial neural networks, decision trees, rough set analysis, support vector machines, logistic models, and

other statistical models, such as time-series analysis, memory-based reasoning, and principal component analysis, are some of the modeling approaches used in data mining.

Depending on the data, each type of model has specific advantages and is suitable for the particular data mining situations. For example, artificial neural networks are very good at fitting extremely complex nonlinear relationships, whereas rough sets analysis is known to produce reliable outcomes in ambiguous and inadequate situations and problems.

Step 5 Assess [9]

The user assesses the value and accuracy of the data mining process' findings. The user evaluates the models to determine how well they perform in this last phase of the data mining process. a typical technique for evaluating a model while sampling. Both this reserved sample and the sample used to build the model should function if the model is correct. The model can also be evaluated using known data.

For example, if consumers in a file have high retention rates and your model predicts retention, evaluate if the model correctly identifies these customers. Furthermore, actual uses of the concept, such as partial mailings in a direct mail campaign, aid in demonstrating its validity.

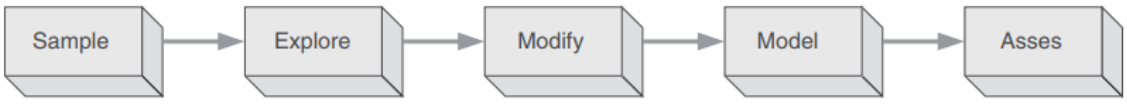


Figure 2 SEMMA Process Steps in Data Mining. [8]

2.4 CRISP-DM Process Model

The CRISP-DM data mining process model gives an overview of a data mining project's life cycle. It comprises the phases of a project, their tasks, and the links between these activities. According to CRISP-DM, the life cycle of a data mining model is divided into six stages from business understanding to deployment. The stages are not in a rigid, by numbers procedure.

Moving back and forth between stages is always necessary. It is determined by the outcome of each phase whether phase or specific job of a phase must be completed next. [8]

The CRISP-DM model has been described the industry-specific, standardized phases of data mining. The term refers to the cross-industry standard method for data mining ("Cross-industry standard process for data mining," 2018). It was established in 1996. It lists six steps in all, as seen in the subsections below (Dubitzky, 2008).[11]

2.4.1 Six Steps of CRISM-DM

1) Business Understanding:

This first stage focuses on analyzing the project's goals and objectives from a business perspective, then applying this understanding to define the data mining challenge and create an initial approach to accomplish the goals.[8]

The sub-steps as follows, [11]

- Find out what the business goals are.
- Determine the purpose of data mining.
- Make a project plan.

The CRISP-DM model consists of 6 stages, with arrows showing how the phases are dependent on one another,

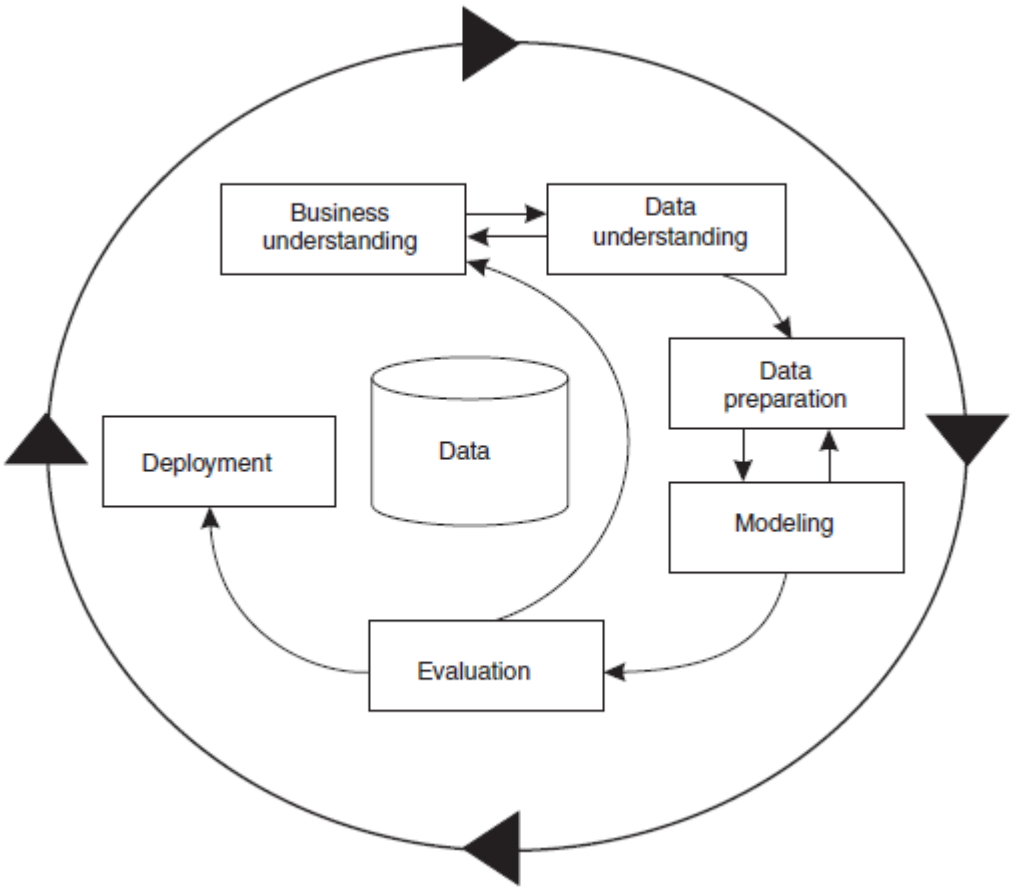


Figure 3: CRISP-DM Model for data mining ('Cross-Industry Standard Process for Data Mining,' 2018) [11]

2) Data Understanding [11]

This process begins with data gathering and familiarization with the data. Data loading and integration are also performed as needed. If any issues arise during this process, they are well documented. This stage focuses on discovering insights in the data, identifying issues with data quality, and finding interesting subsets within the data set that aid in the formation of hypotheses regarding hidden facts within the data.

This process may be basically divided into,

- Data collection.
- Data description.
- Validation of data quality.
- Exploration of data.

3) Data Preparation

The data preparation phase includes all operations that are performed to create the final data set (data that will be input into the modeling tool(s)) from the raw data. Data preparation procedures are likely to be repeated several times and in no particular order. Table, record, and attribute selection, as well as data processing and cleansing for modeling tools, are all tasks.[8]

This stage can be broken down into the following steps, [11]

- Selection of data.
- Data cleaning
- Structure of data
- Data integration
- Data formatting.
-

4) Modeling [11]

The modeling stage is the most essential in the CRISP-DM lifecycle. In this stage, one or more modeling approaches are selected and applied to the data. For the same data-mining activity, multiple parameters can be defined, and alternative models can be developed. This is done because some models have special requirements for the data format. The chosen model is then tested and verified to determine its quality and validity for the given data-mining challenge. Empirical tests are done during model testing to assess the model's strength.

In general, the following steps must be taken,

- Modeling techniques selection.
- Model development.
- Evaluation of the model.

5) Evaluation [11]

This stage evaluates and reviews the model's architecture to ensure that it accomplishes the business goal and solves the data mining challenge correctly. The outcomes are examined in order to validate the business needs. The models are evaluated based on the business success criteria, and the models that satisfy these criteria are selected as the final models to be used. At this step, one should analyze the model to see whether specific business contexts have been addressed by the model.

In summary, the stages are as follows,

- Assessment of the outcome
- Technique of review.
- Making a decision on new steps.

6) Deployment

In general, the model's construction is not the conclusion of the project. Even if the model's goal is to expand data knowledge, it will be required to arrange the extracted information and provide it to the client in a proper way. The deployment step might be as easy as creating a report or as sophisticated as establishing a repeatable data mining process, depending on the needs.[8]

The following sub-steps of deployment phase, [11]

- Plan of deployment
- Implementation of final project
- Evaluation of the project.

2.5 Data Mining Architecture [12]

The approach employed to link together the analytical processes in the data mining cycle (Figure 4) to establish a repeatable business process is referred to as data mining architecture. A business will struggle to design and deploy analytical models without a workable architecture, specifically among big data warehouses.

The architecture is less important when data volumes are quite small (usually around 100,000 records) and there are just a few models that need to be controlled. Data mining may be implemented by loading the results into the proper application after applying the model algorithm to the necessary data on a powerful PC or server.

However, the choice of data mining architecture becomes more crucial when the data warehouse includes millions of entries and/or when there are several models to be installed. The best-practice method in these situations includes,

- Performing data preparation tasks directly in the data warehouse, such as profiling, converting, and creating analytic datasets.
- Creating SQL code from data mining models so that it may be performed directly in the data warehouse, eliminating the need to send significant amounts of data into another system for model scoring.

Data mining architecture included six components as follows, [13]

1. **Data Sources:** Data or historical data from a variety of sources, including the internet, databases, data warehouses, big data, text files, and/or documents. The success of data mining is more suited to massive amounts of data. Before being integrated into another system or being sent to a database or data warehouse, this data must first be cleaned and normalized.
2. **Data Storage:** The data storage server might be either a database or a data warehouse server, or both. This is the actual location where the data is saved for retrieval or processing by a user during data mining. The data that has been recorded should have already been cleansed and integrated.
3. **Data Mining Engines:** These are the fundamental components of every data mining system. It is made up of many modules or applications that process and retrieve measurable and qualitative data. Data association, classification, characterization, clustering, prediction, reporting, analysis, unification, and other tasks are performed here.

4. **Pattern Evaluation Modules:** Pattern analysis is investigating a regular sequence of data algorithms or a repeated occurrence of data to build a quantifiable or common threshold.

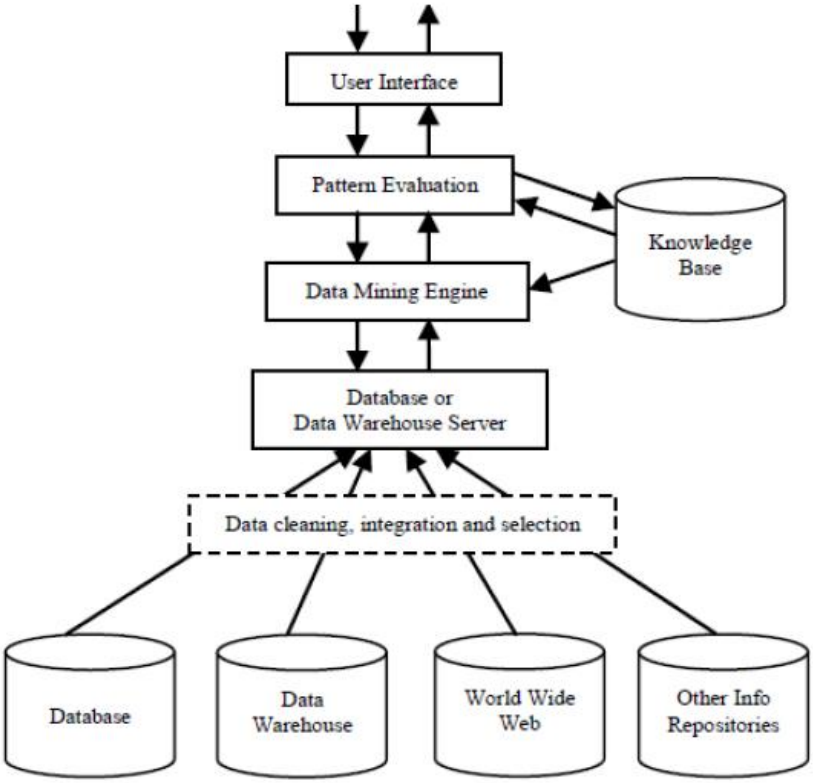


Figure 4: Typical Data Mining Architecture. [14]

5. **Graphical User Interface:** Tasks and queries are entered to generate a readable result from the stored data where the user interacts with the data mining systems.
6. **Knowledge Base:** In order to get a more accurate and reliable result from the data mining process without having to input the same pattern or code again to produce the same output, it can be helpful to look into a knowledge base for searches, previous data, and/or patterns where the resulting processes had been done before.

2.6 Data Mining Techniques

Data that is important to businesses can take numerous forms. Data mining may be used in a variety of business contexts. When faced with millions of accident claims, a vehicle insurance firm recognizes that not all of them are authentic. They will spend more money on inquiry than they would pay in claims if they are particularly hard and thoroughly analyze each claim. Insurance companies have developed methods for profiling claims that take into account a variety of factors in order to offer an early indication of instances that are likely to require the expenditure of costs for inquiry. With a significant number of application manufacturers, fraud detection has become a competitive data mining sector. This is typical of many applications for data mining. [9]

Data mining describes a variety of methodologies and strategies for extracting meaningful information from huge databases. Numerous implementation techniques and algorithms have emerged as a result of the significance of data mining in many diverse businesses. Data mining is divided into two categories: descriptive data mining and predictive data mining. [15]

- **Predictive Data Mining** [15]
    - Analyzes historical data to discover what is going on in a business's past and present.
    - Prediction focuses on identifying relationships between independent variables and relationships between dependent and independent variables.
    - Forecast explicit values based on data patterns, with the purpose of identifying a statistical or neural network model that may be utilized to forecast some type of interesting outcomes.
- **Descriptive Data Mining** [15]
    - Forecasts and creates models based on past and present data, helping businesses to make predictions about the future.
    - Descriptive data mining presents a data collection in a quick but thorough manner and provides interesting data features without any specified target.
    - The fundamental structure, relationships, interconnections, etc. of the data are given greater attention by descriptive model than the goal value itself.

- These techniques use the data provided to them to demonstrate how various things are connected. Data is "described" by them.

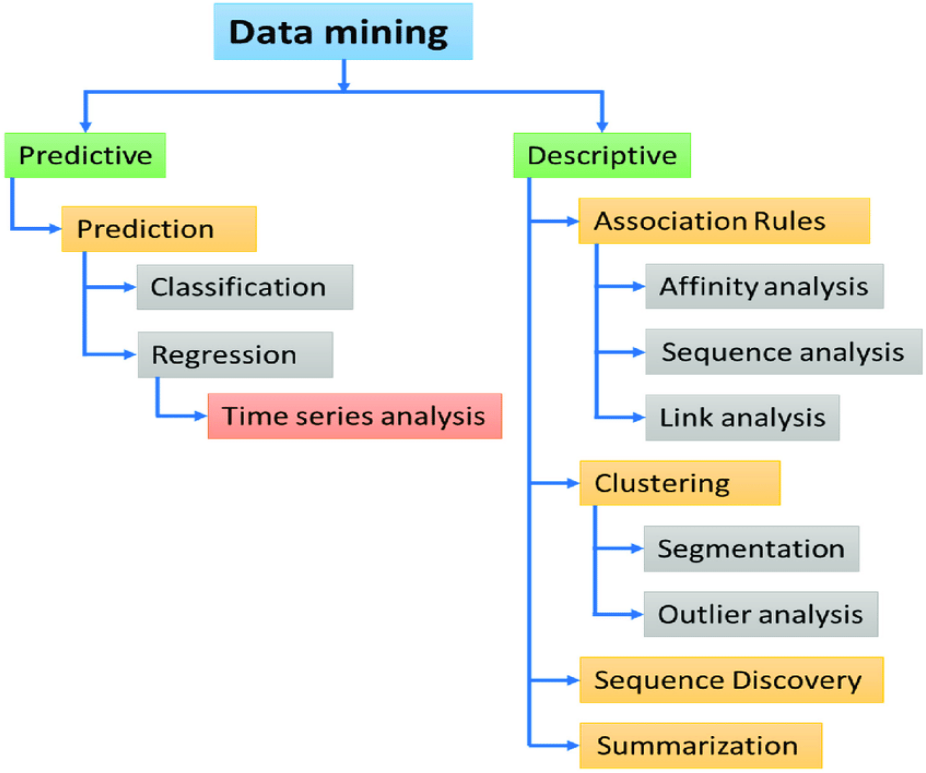


Figure 5: Overview of Predictive and Descriptive data mining. [20]

Data mining techniques are divided into categories based on the type of pattern we are looking for,

Association: Association is the establishing of a relationship or link between two data events. Association rule refers to this type of togetherness or connection. Association rule mining is a data mining approach created in 1993 by agrawal, imielinski, and swami. This is a well-organized data mining technique for searching for hidden or desired patterns in data. The main objective of this technique is to explore relationships between various items in a relational database, transactional databases, and other forms of data repositories [11, 16]

It is also known as "Market Basket Analysis" because this was first use of association rules mining. The primary purpose is to find for associations between things that occur together

more frequently than predicted from an essentially random sample test of all possible outcomes.

Association rules are primarily strategies for identifying interesting, intriguing associations between different parameters variables in a database or data set. This data mining technique can be useful in detecting unique hidden sequences inside data. For examples, a customer typically purchases bread and sandwich meat together. With this sort of relationship between the two items, a company may create a marketing strategy that incorporates both of them, like putting them together in a coupon for a discount or for TV advertising.[11]

Association Rule used algorithms are,

- Apriori, SETM, AIS, AprioriTid, AprioriHybrid [11]

Classification: A machine learning-based data mining technique is classification. Classifying attributes into target categories. In general, classification is used to place each piece of information in a batch of data into one of the established categories or groups. Mathematical methods including decision trees, linear regression, neural networks, and statistics are used in the classification procedure. [15]

An algorithm is used to process a collection of attributes and the corresponding outcome, which is referred to as a goal or a prediction attribute. The idea is to find correlations and relationships between the variables that might improve in prediction. The predicted class is referred to as the target class.

Using a classifier, the classification predicts the value of a category target class from other categorical or numerical classes. A categorical class is one that has a set of fixed discrete values (for example, yes or no, adult or child, weekly, monthly, or daily, etc.). [11]

Different kinds of classification models, [17]

- Decision tree induction for classification
- Classification Based on Associations
- Naïve Bayesian Classification
- Neural Networks

- Support Vector Machines (SVM)

Decision trees were a data mining approach used to evaluate credit risk in 2003 (Mues et al., 2003), and visualization was achievable with the assistance of decision trees.[11]

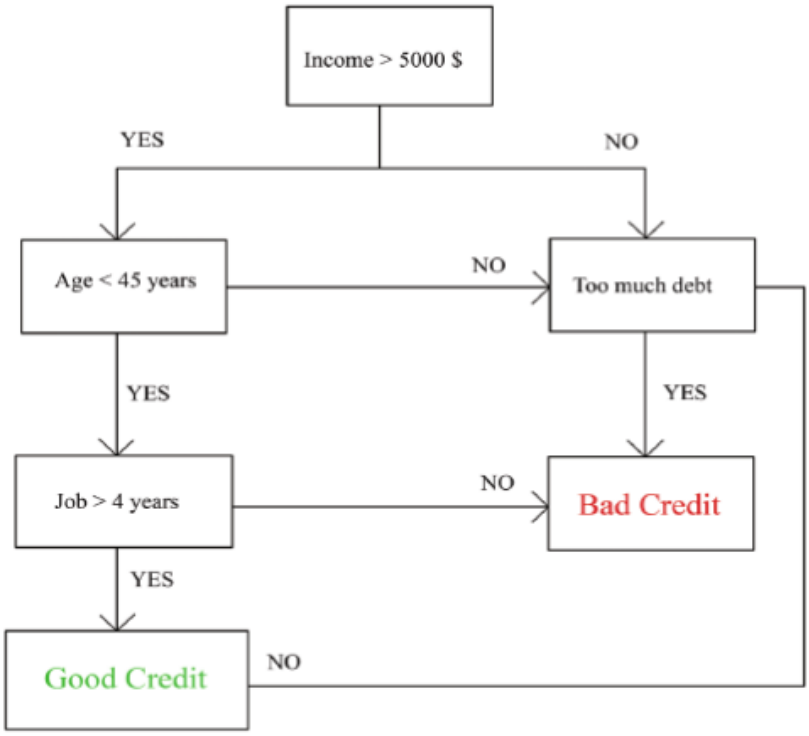


Figure 6: Decision tree (rule induction) along with a classification implementation that calculate credit rating. [11]

Clustering: Involves grouping data into clusters so they can be treated as groups. Clustering is defined as the recognition of comparable classes of objects. We may detect overall distribution patterns and relationships among data characteristics by utilizing clustering algorithms to locate dense and sparse areas in object space. Classification may also be used to differentiate groups or classes of objects, but it is expensive, therefore clustering can be used as a preprocessing strategy for attribute subset selection and classification. For example, to categorize genes with similar functioning and generate groups of customers based on purchase behaviors. [17]

As a data mining tool, the clustering algorithm has various applications in sectors such as security, biology, astronomy, business intelligence, and so on. It has been widely used in

various fields, including pattern identification, market analysis, picture mining, image processing, and so on.

Clustering can assist businesses in detecting groups of similar clients and characterizing their purchasing patterns. They can be classified into comparable categories based on the products they purchase. [11]

Major types of clustering methods, [18]

- Partitioning method
- Hierarchical method
- Density-based method
- Grid-based method
- Model-based method
- Outlier analysis

Prediction: Regression, a statistical methodology developed by Sir Frances Galton (1822-1911), a mathematician and Charles Darwin's cousin, is by far the most extensively used tool for numeric prediction (hereinafter referred to as prediction). Many texts, in fact, use the words "regression" and "numeric prediction" interchangeably. [18]

The regression approach can be used to predict outcomes. To model the relationship between one or more independent variables and dependent variables, regression analysis can be utilized. In data mining, independent variables are known attributes, while dependent variables are what we want to forecast. Unfortunately, many real-world situations cannot be predicted.

For example, sales volumes, stock prices, and product failure rates are all exceedingly challenging to forecast because they may be influenced by complicated interactions among several predictor factors. Therefore, it may be essential to estimate future values using more advanced techniques (such as logistic regression, decision trees, or neural networks). Frequently, the same model types may be applied to classification and regression. For instance, classification trees (used to categorize categorical answer variables) and regression trees may both be created using the CART (Classification and Regression Trees) decision tree

technique (to predict continuous dependent variable). Additionally, neural networks may produce both regression and classification modes. [17]

Types of regression method, [17]

- Linear Regression
- Multivariate Linear Regression
- Nonlinear Regression
- Multivariate nonlinear Regression

Anomaly or Outlier Detection: Finding patterns in data that are not normal and unexpected. Outlier detection (also known as anomaly detection) is the process of identifying data objects having behaviors that significantly differ from expectation. These objects are referred to as anomalies or outliers. [18]

The deviation detection model is the process of identifying items, events, or observations in data mining which do not follow an expected pattern or other items in a dataset. (Han et al., 2011). Deviation detection can give important information for further analysis. [11]

A credit card business, for example, protects its consumers from credit card fraud and pays extra attention to card usages that are out of the regular. In this case, if a card owner's buy amount is significantly more than normal, and the transaction happens distant from the owner's resident city or region, the purchase is suspicious. Such transactions should be detected as quickly as possible, and the card owner should be alerted for verification. Many credit card businesses use this technique. However, when a credit card is stolen, the regular transaction pattern is significantly changed; the places and goods purchased are frequently considerably different from those of the authorized card owner and other customers. The goal of credit card fraud detection is to detect transactions that are out of the regular. [18]

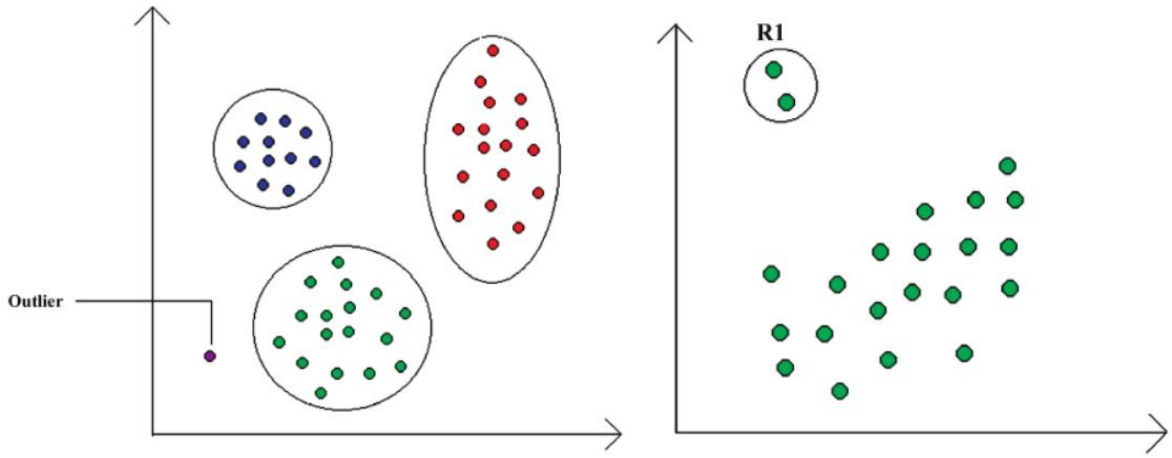


Figure 7: Example of outlier in above figures, here dot and R1 objects are outliers. [18]

Sequential Patterns: Tracing a series of events that take place in a sequence. Sequential patterns are similar to association rule mining. Sequence detection is an effective tool for identifying patterns, trends, or often occurring similar events in data. In simpler terms, it is the detection of common subsequences in a group of sequences, where each sequence indicates events that occur at particular periods. [11]

A common example is the analyzing of customer data over time. When the customer wants to buy more products, this mined information may be utilized to automatically offer certain specific items to the customers. These recommendations are based on the customer's frequency and previous purchasing history. [11]

2.7 Data Mining application in Business Context

The goal of early data mining applications was to provide firms a competitive edge. As e-commerce and e-marketing have become commonplace in the retail sector, research into data mining for businesses keeps growing. Data mining is being utilized more and more to investigate potential applications in different fields, including online and text analysis, financial analysis, business, government, and research. More application-specific data mining systems and tools, as well as covert data mining features integrated into many services, may be in the horizon. The following problem types are covered by data mining applications: classification, prediction, association, and detection.[19]

2.7.1 List of Data Mining Applications [9]

Table 1: List of Data Mining Applications in Business Context.

Area	Technique	Application	Problem type
Finance	Neural network	Forecast stock price	Prediction
	Neural network Rule induction	Forecast bankruptcy Forecast price index futures Fraud detection	Prediction Prediction Detection
	Neural network Case-based reasoning	Forecast interest rates	Prediction
	Neural network Visualization	Delinquent bank loan detection	Detection
	Rule induction	Forecast defaulting loans Credit assessment Portfolio management Risk classification Financial customer classification	Prediction Prediction Prediction Classification Classification
	Rule induction Case-based reasoning	Corporate bond rating	Prediction
	Rule induction Visualization	Loan approval	Prediction
Web	Rule induction Visualization	User browsing similarity analysis	Classification, Association
	Rule-based heuristics	Web page content similarity	Association

Telecom	Neural network Rule induction	Forecast network behavior	Prediction
	Rule induction	Churn management Fraud detection	Classification Detection
	Case-based reasoning	Call tracking	Classification

Marketing	Rule induction	Market segmentation Cross-selling improvement	Classification Association
	Rule induction Visualization	Lifestyle behavior analysis Product performance analysis	Classification Association
	Rule induction Genetic algorithm Visualization	Customer reaction to promotion	Prediction
	Case-based reasoning	Online sales support	Classification

2.7.2 Some Overview of Data Mining Application

Target Marketing: The foundation of marketing management is targeting. It is concerned with providing the correct products to the customer at the right time and through the effective channels. Data mining may be used to create marketing campaigns to promote sales and launch new services. Nowadays, banks serve a wider range of customers, each of which is seeking for something special. Recent advances in computer and database technology are assisting in these aims by utilizing database marketing, data mining, and, more recently, CRM technologies to better understand the client and approaches. [11,19]

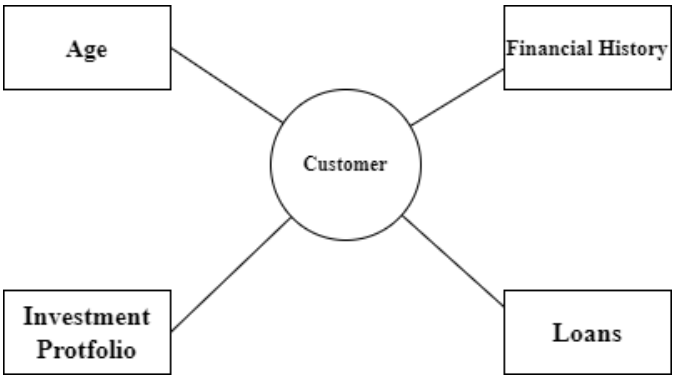


Figure 8: Example of each customers data associated with financial history, loans, investment, age. [11]

Traditionally, data mining techniques were primarily focused on extracting quantitative and statistical data from data warehouses. These traditional approaches were helpful in obtaining interesting interpretations of the data as well as insights into the processes behind the data. Although these procedures finally lead to knowledge discovery, they are still vulnerable to human mistake. Human analysts may overlook vital information that might assist enhance business profitability and operations.

Compared to existing techniques, data mining can help more efficiently. The amount of human work is significantly decreased by the data mining approaches.[11]

In terms of marketing, data mining has two key applications:

- customer retention and
- customer attraction.

The decision-making process for a targeted marketing application. In the case of a new product, the procedure is frequently launched with a test mailing to a sample of customers in order to analyze customer feedback. People in the crowd who "look like" the test purchasers are then chosen for the campaign. For a previously promoted program, the prediction model is based on the outcomes of the prior campaign for the same product. The left side of Figure 9 relates to the testing phase, whereas the right side refers to the rollout phase. The target audience, also known as the universe, is often, but not always, a subset of the customer list that includes only consumers who, based on some earlier evaluation, are possible customers for the company.[19]

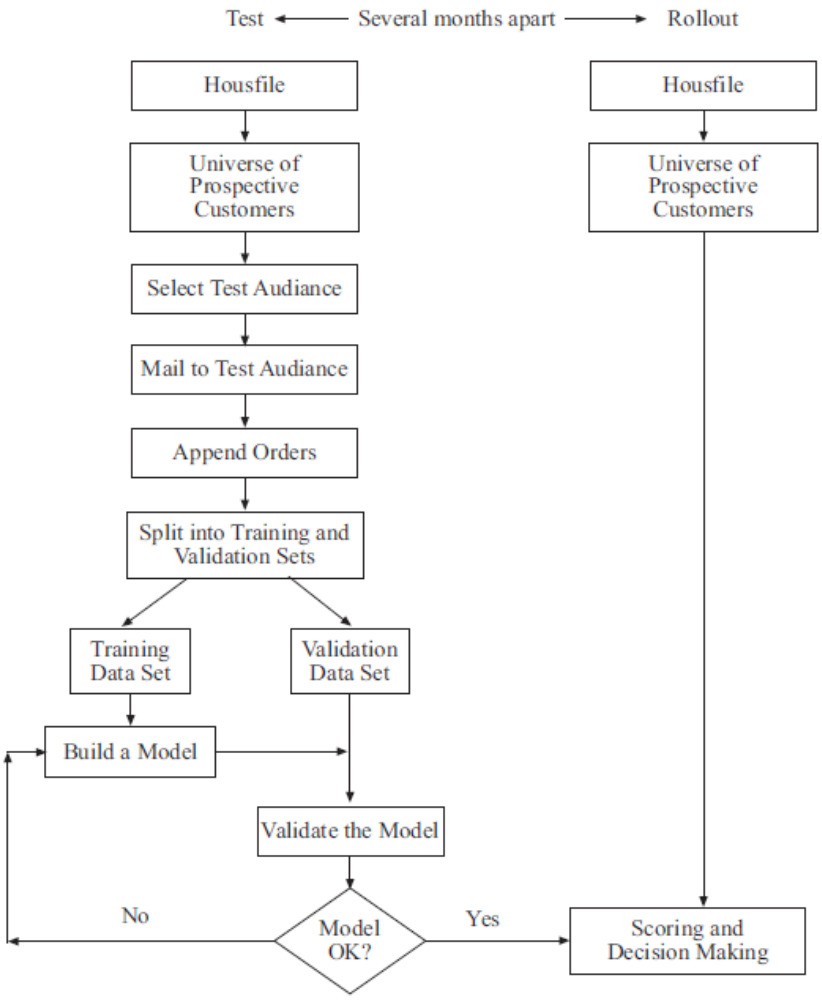


Figure 9: Decision making process on targeted marketing application for new customer promotion.[19]

Online Transaction: The banking business is rapidly expanding, and banks are introducing online banking and e-marketing. As the number of customers in the online market rises, so does the number of malicious persons with harmful intentions. The number of incidents of fraud continues to rise. Even though, there are unpleasant persons on the internet, users are continually targeted by various sorts of cyber-attacks. In the event of fraud and identity theft, malicious people can get illegal access to sensitive information about the customer, such as their account number and credit card data. This may be accomplished by using numerous hacking techniques such as phishing, packet sniffing, DoS attacks, and so on.

Banks have taken steps to secure their customers by implementing strong encryption and authentication techniques. However, this is not always sufficient to prevent customer data hacking. In this situation, the technique is similarly to that of typical fraud detection, except that the data is now encrypted, and data mining is used to discover anomalies in the customer's purchasing history. By integrating data mining, steganography, and cryptography, Devadiga et al. have built security methods (2017) [11]

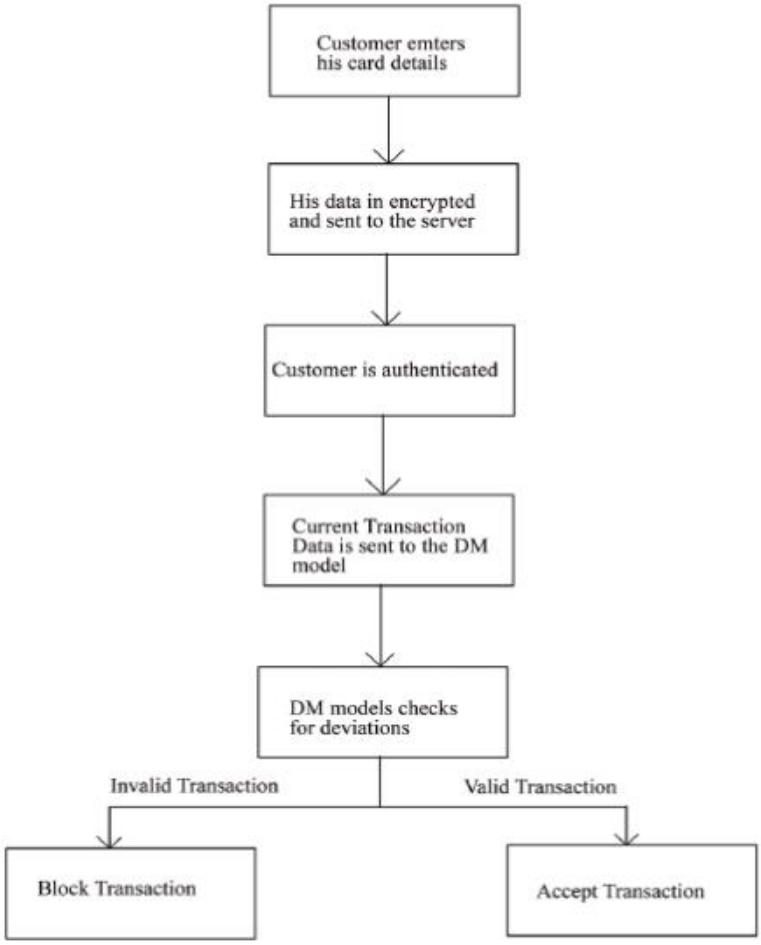


Figure 10: Typical use case of online transaction activity protection.[11]

Fraud Detection: Classification, anomaly detection, and association techniques in data mining are useful for detecting fraudulent activities.

The use of data mining techniques to customers transaction data helps in the recovery of usable knowledge from historical data. Data mining techniques enable in the discovery and exploration of relationships and correlations between financial variables in order to identify suspicious behaviors that may present a fraud risk to the bank. These techniques assist in focusing attention on transactional behavior and classifying data into two types: fraud and non-fraudulent. [11]

Financial organizations, particularly banking sectors, use primarily two ways to determining fraud patterns, online transaction check and offline transaction check (Ramageri and Desai, 2013; Moin and Ahmed, 2012). For this reason, institutions acquire and manage data warehouses of sanctions and politically exposed persons data files from Compliance and Anti Money Laundering solution and data providers such as the US Office of Foreign Assets Control (OFAC). Financial Action Task Force (FATAF), Financial Market Supervisory Authority (FINMA), Financial Services Authority (FSA), Hong Kong Monitory Authority (KKMA), and Reserve Bank of India (RBI) develop standards and require financial institutions to provide various reports on a regular basis. The system can access various data sources and prepare reports based on combinations of data mining techniques such as classifying, clustering, segmentation, association rules, sequencing, regression, pattern analysis, and decision trees (Handl and Knowles, 2012; Poovammal and Ponnavaiko, 2009; Wang and Dong, 2009; Issam, 2012; Tremblay et al., 2010; Akbar et al., 2010; Herawan and Deris, 2011; Petry and Zhao. [35]

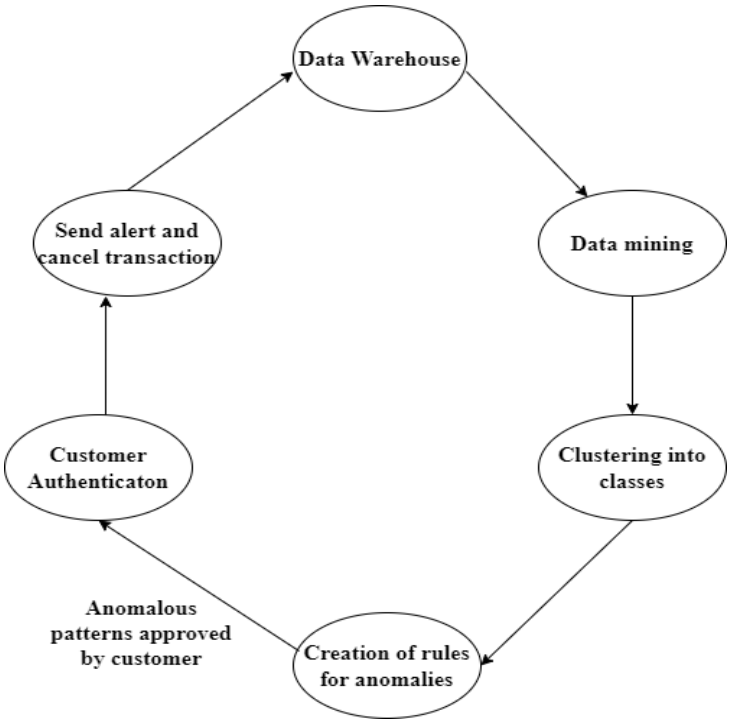


Figure 11: The process of monitoring credit card fraud using the data mining approach. [11]

Data mining technologies are designed to help human decision-making. As a result, data mining is designed to reveal patterns that humans can understand. Machine learning, on the other hand, automates the process of discovering patterns that may then be utilized to create predictions.[22]

3. Machine Learning

The first section of my work discusses the background of data mining in business. Machine learning is one type of artificial intelligence. Arthur Samuel, an AI pioneer, characterized it in the 1950s as "the branch of research that provides computers the ability to learn without specifically being programmed."

With the massive expansion of data in recent years, several businesses are attempting to benefit on this new resource for their business solutions. Machine learning (ML) is becoming increasingly significant in practically all areas of business, from marketing to governmental activities to scientific, health, and security applications (Chen et al., 2012). Furthermore, many businesses rely on ML models implemented in their information systems to improve the

efficiency of their operations or to offer new services and products (Schüritz et al., 2016). (2015) (Dinges et al.). According to Davenport (2006), organizations who can utilize their data sources through analytical tools gain a significant competitive advantage.[21]

Google stated in November 2016 that it has integrated its multilingual neural machine translation system into Google Translate, marking one of the first success stories of deep neural artificial neural networks in scale production. According to Google, this upgrade increased Google Translates translation quality more in a single step than the previous 10 years combined.[22]

3.1 Machine Learning Techniques

Machine learning has emerged as one of the most significant issues among development companies searching for advanced techniques to integrate data assets to help the company achieve a new degree of knowledge. Why include machine learning in the equation? Organizations may use proper machine learning models to continuously forecast changes in the company so that they can better predict what's next. Because data is continually being supplied, machine learning models ensure that the solution is always up to date. The benefit is straightforward: Using the most relevant and continuously changing data sources in the framework of machine learning, we can forecast the future.[2]

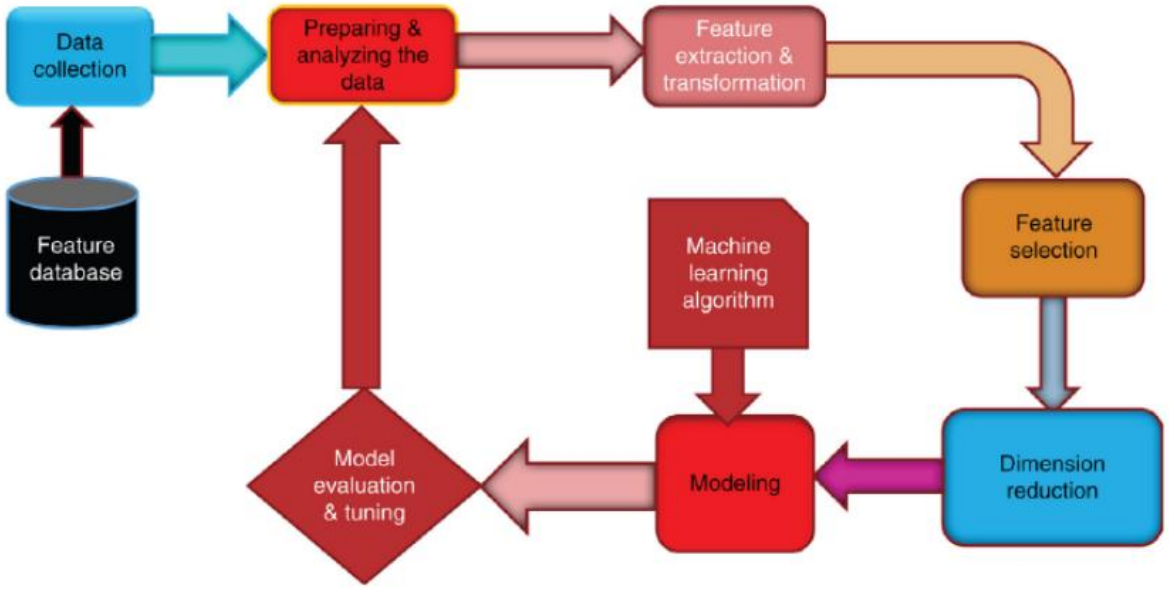


Figure 12: A typical machine learning process. [24]

The four primary categories of machine learning methods based on human supervision are as follows:

1. Supervised Learning
2. Unsupervised Learning
3. Semi-supervised Learning and
4. Reinforcement Learning.

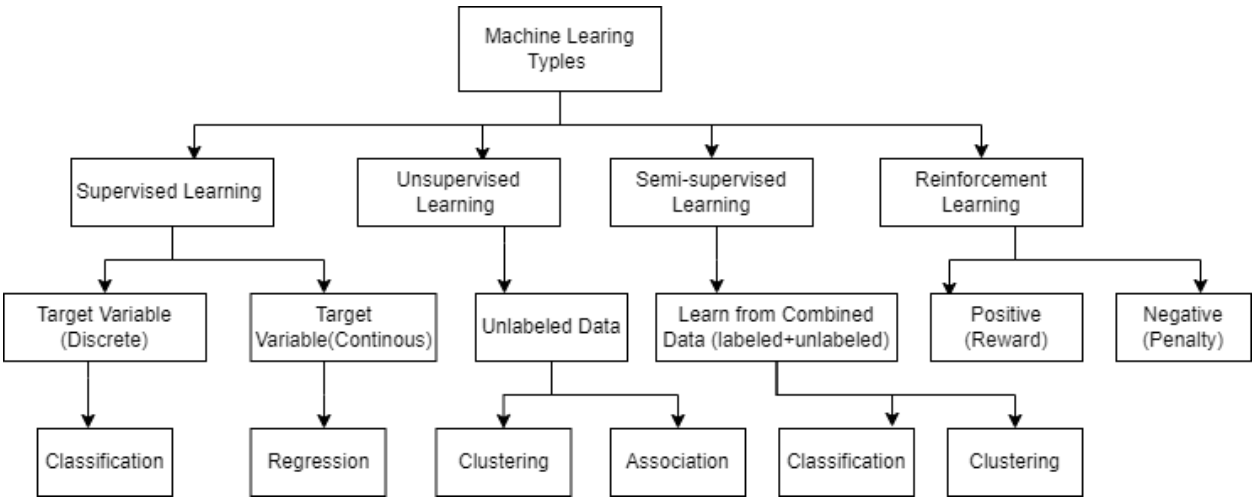


Figure 13: Several types of machine learning techniques.[23]

3.1.1 Supervise Learning

The goal of supervised learning is to learn the mapping between sample input and output pairs. Simply described, a supervised learning method may have multiple input variables and a single output variable. Logically, a supervised learner's prediction skill is proportional to the number of instances available for learning. In general, supervised learning is applied to two different tasks: classification and regression. In summary, supervised algorithms or models learn from labeled data (task-driven approach)

For instance, an algorithm might be trained to detect images of dogs and other objects on its own using photos of them that have been labeled by humans. The most popular kind of machine learning nowadays is supervised learning.[23]

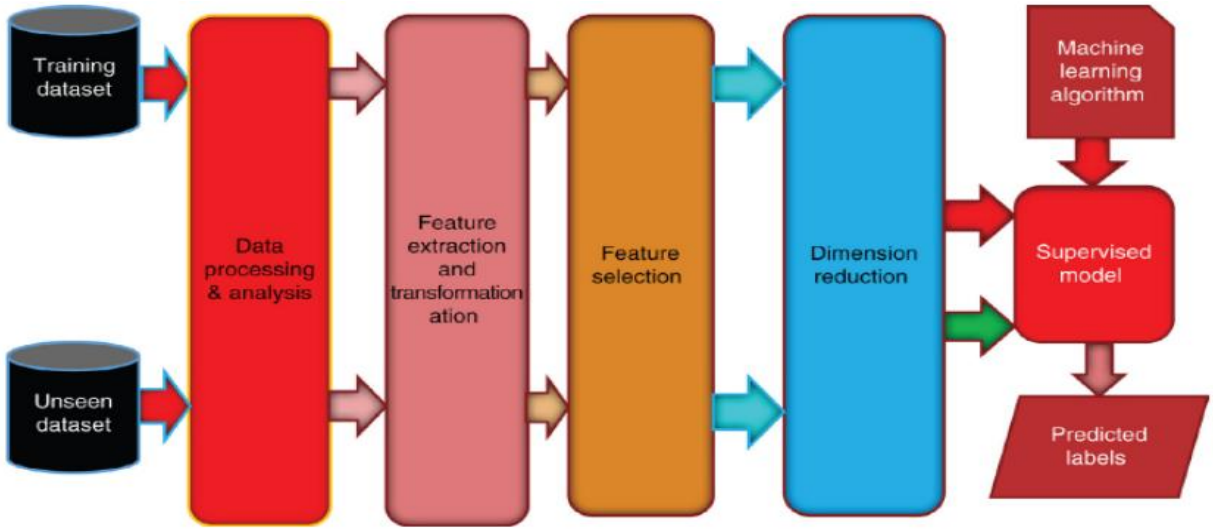


Figure 12: Supervised machine learning pipeline. [24]

3.1.2 Unsupervised Learning

Unsupervised learning is a data-driven technique that analyzes unlabeled datasets without the involvement of humans. This is frequently used for generating feature extraction, relevant trend and structure identification, result groupings, and experimental reasons. Clustering, density estimation, feature learning, dimensionality reduction, association rule discovery, anomaly detection, etc. are some of the most popular unsupervised learning tasks. In summary, algorithms or models learn from unlabeled data (Data-Driven Approach)

For instance, by analyzing online sales data, an unsupervised machine learning algorithm may determine the various customer groups making purchases.[23]

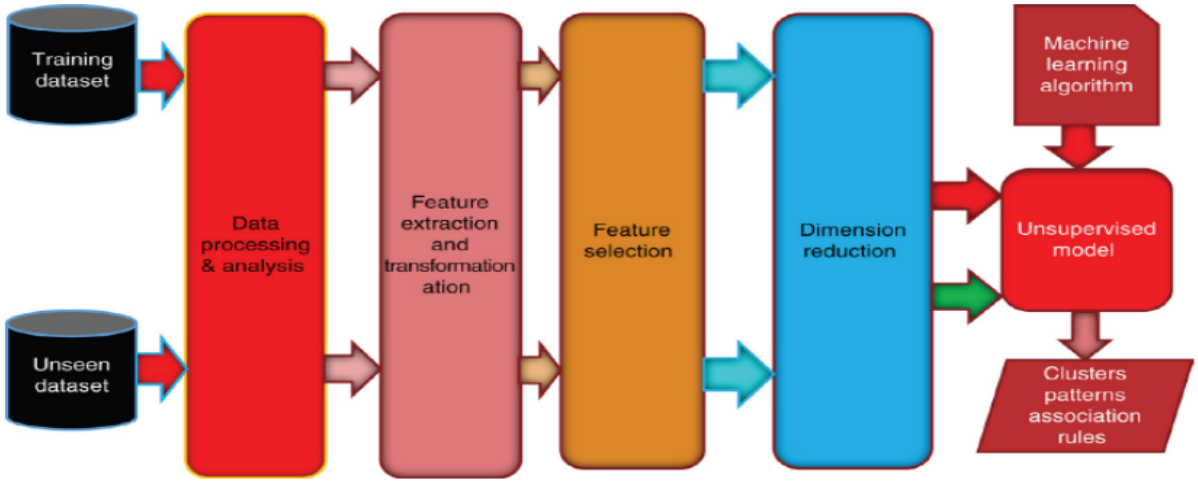


Figure 13: Unsupervised machine learning pipeline. [24]

3.1.3 Semi-supervised Learning

Both labeled and unlabeled data can be used for semi-supervised learning. Consequently, it lies in the middle between learning "with supervision" and "without supervision". In the real world, semi-supervised learning is helpful since unlabeled data are common while labeled data may be uncommon in certain contexts. A semi-supervised learning model's ultimate objective is to create predictions that are more accurate than those made only utilizing the model's labeled data.

There are some applications for semi-supervised learning. includes labeling, fraud detection, and machine translation classifying text and data.[23]

3.1.4 Reinforcement Learning

Reinforcement learning is a form of machine learning technique that allows software agents and machines to automatically analyze the best behavior in a given context or environment in order to increase their efficiency. This sort of learning is focused on reward or penalty, and its ultimate purpose is to use environmental activists insights to take action to raise the reward or reduce the risk. It is a powerful tool for training AI models that can assist in the automation or optimization of the operational efficiency of sophisticated systems such as robotics, autonomous driving tasks, manufacturing, and supply chain logistics. However, it is not

recommended for solving simple or straightforward problems. In conclusion, models are based on reward or penalty (environment-driven approach).[23]

3.2 Machine Learning Tasks and Algorithms

All ML algorithms have one thing in common: they model patterns in prior data and generate predictions, and the quality and relevance of this experience are critical variables in their performance. Where they vary is that each form of algorithm has unique properties and brings unique problems in machine learning. Figure 16 shows the overall framework of a machine learning-based predictive model, in which the model is trained using historical data in phase 1 and the outcome is created for new test data in phase 2.

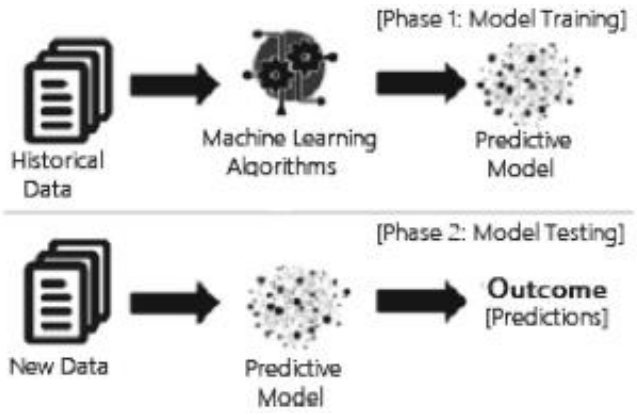


Figure 14: Example of machine learning based predictive model with training and testing phase.[24]

Linear Regression

Linear regression, or *ordinary least squares* (OLS), is the simplest and most classic linear method for regression. This is a well-known regression algorithm as well as one of the most common machine learning algorithms. The dependent variable is continuous in this approach, the independent variable(s) might be continuous or discrete, and the regression line is linear. Linear regression uses the best fit straight line to build a link between the dependent variable (Y) and one or more independent variables (X). The following equations define it:

$y = a + bx + e$ ----------(i)

$y = a + b_1x_1 + b_2x_2 + \cdots + b_nx_n + e$----------(ii)

where a represents the line's intercept, b its slope, and e its error term. Based on the provided predictor variable, this equation may be used to forecast the value of the target variable (s). In contrast to basic linear regression, which only contains one independent variable described in Eq. (i), multiple linear regression allows two or more predictor variables to model the response variable, y, as a linear function defined in Eq. (ii) [23]

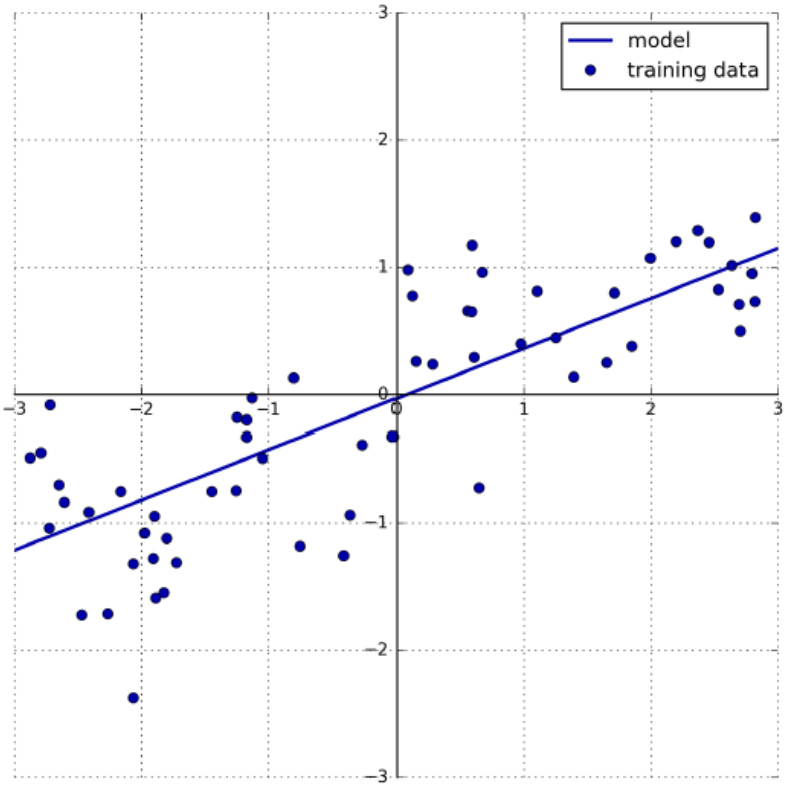


Figure 15: Example of prediction on linear model in dataset.[26]

Businesses can use linear regressions to analyze trends and provide estimates or projections. For instance, if a company's sales have been rising consistently each month for the previous few years, the company might predict sales in the upcoming months by performing a linear analysis of the sales data with monthly sales.

Logistics Regression

Logistic Regression is another popular statistical model with a probabilistic base that is used to solve classification problems in machine learning. A logistic function, also known as the sigmoid function, is commonly used in logistic regression to estimate the probabilities. When

the dataset can be divided linearly, it performs well but can overfit high-dimensional datasets. One of the main issues with logistic regression is the assumption that the dependent and independent variables are linearly related. Although it may be applied to classification and regression problems, classification problems are the one it is most applied on. [23]

Figure 16: Example of decision boundary of a logistic regression on the dataset.[26]

Decision Tree

The non-parametric supervised learning technique known as the decision tree is widely used. Both the classification and the regression tasks are handled using decision learning techniques. Popular decision tree algorithms include ID3, C4.5, and CART. Additionally, the recently suggested BehavDT and IntrudTree by Sarker et al. are successful in the essential application areas, such as user behavior analytics and cybersecurity analytics, accordingly.

Marketing strategies occasionally make use of decision trees. You might wish to forecast the results of distributing a 20 percent discount to clients and potential clients.[23]

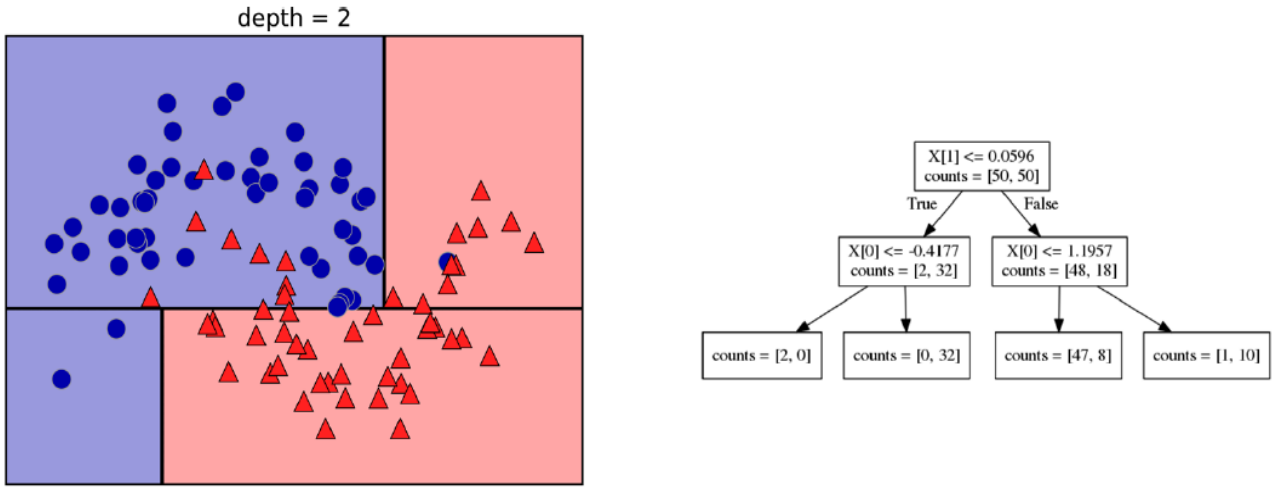


Figure 17: Example of decision boundary with corresponding decision tree.[26]

k-Nearest Neighbor

The k-NN method is likely the most basic machine learning algorithm. The only thing required to build the model is to save the training dataset. To create a forecast for a new data point, the algorithm seeks the "nearest neighbors" in the training dataset.

KNN analyzes data and classifies new data points using closeness measurements (e.g., Euclidean distance function). It is relatively resistant to noisy training data, and its accuracy is dependent on data quality. The most difficult aspect of KNN is determining the ideal number of neighbors to consider. KNN may be used for both classification and regression. [23, 26]

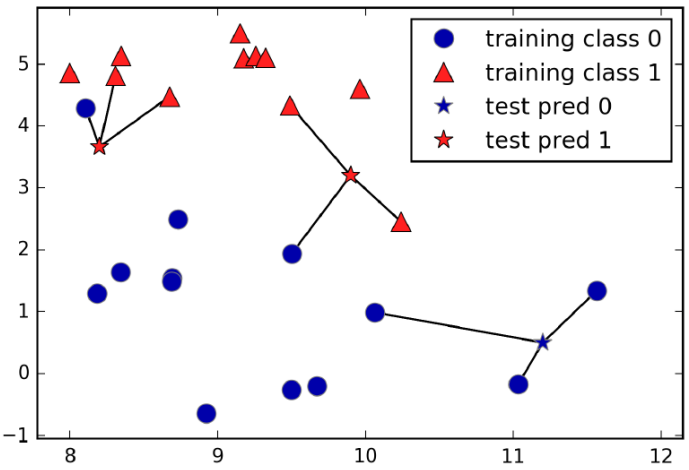


Figure 18: Example of prediction made by three nearest neighbors on datasets.[26]

Random Forest

A random forest classifier is a well-known machine learning ensemble classification approach. This approach employs "parallel ensemble" which involves fitting multiple decision tree classifiers in parallel on multiple data set sub-samples and using majority voting or averages for the conclusion or result. It reduces the over-fitting problem while increasing forecast accuracy and control.

As a result, the random forest learning model with several decision trees is often more accurate than a model based on a single decision tree. It combines bootstrap aggregation (bagging) with random feature selection to generate a succession of decision trees with controlled variance. It is applicable to both classification and regression tasks and works well with both categorical and continuous datasets.[23]

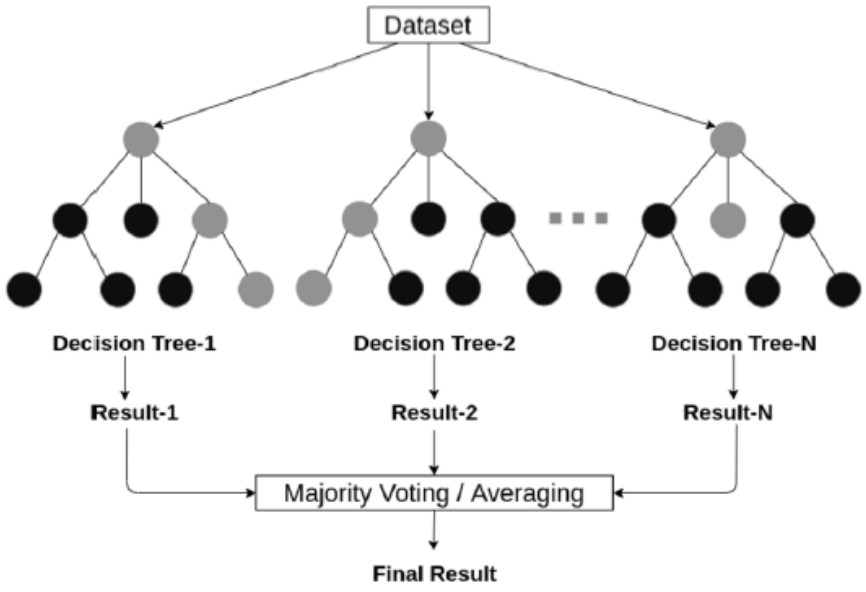


Figure 19: An example of random forest structure with several decision tree. [23]

The random forest algorithm can help doctors with difficulties including gene expression classifications, biomarker identification, and sequence annotation. As an outcome, doctors may make educated predictions about how patients will respond to various medications. [27]

Extreme Gradient Boosting

Gradient Boosting, like Random Forests, is an ensemble learning technique that builds a final

model based on a succession of individual models, often decision trees. The gradient is used to minimize the loss function, similar to how neural networks utilize gradient descent to optimize weights. Extreme Gradient Boosting (XGBoost) is a kind of gradient boosting that considers more detailed approximations while finding the optimal model. XGBoost is easy to use and can handle big datasets.[23]

k-Means Clustering

One of the most basic and widely used clustering algorithms is k-means clustering. It identifies cluster centers that are representative of certain sections of the data. The method alternates between two steps: allocating each data point to the nearest cluster center and then setting each cluster center as the mean of the data points given to it. When the assigned of instances to clusters no longer changes, the process is done.[26]

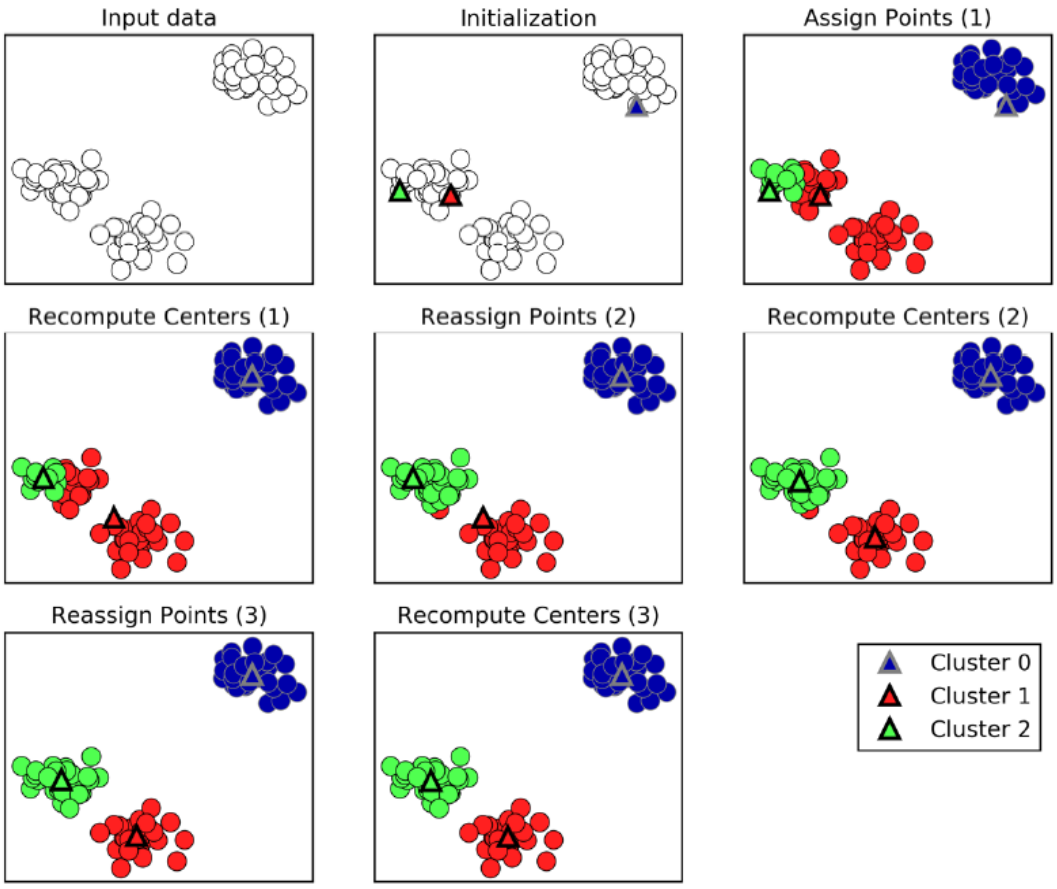


Figure 20: An example of input data and three steps of the k-means algorithm.[26]

Deep Learning and Artificial Neural Network

Deep learning is one of a larger family of machine learning algorithms that use artificial neural networks (ANN). Deep learning provides a computational framework for learning from data by merging numerous processing levels, such as input, hidden, and output layers. The fundamental benefit of deep learning over typical machine learning approaches is that it performs better in a variety of situations, particularly when learning from enormous datasets. Given the rising amount of data, deep learning is preferred over machine learning. It may, however, differ based on the data qualities and experimental setup. Deep learning methods that are commonly used include: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN, or ConvNet), and Long Short-Term Memory Recurrent Neural Network (LSTM-RNN). [23]

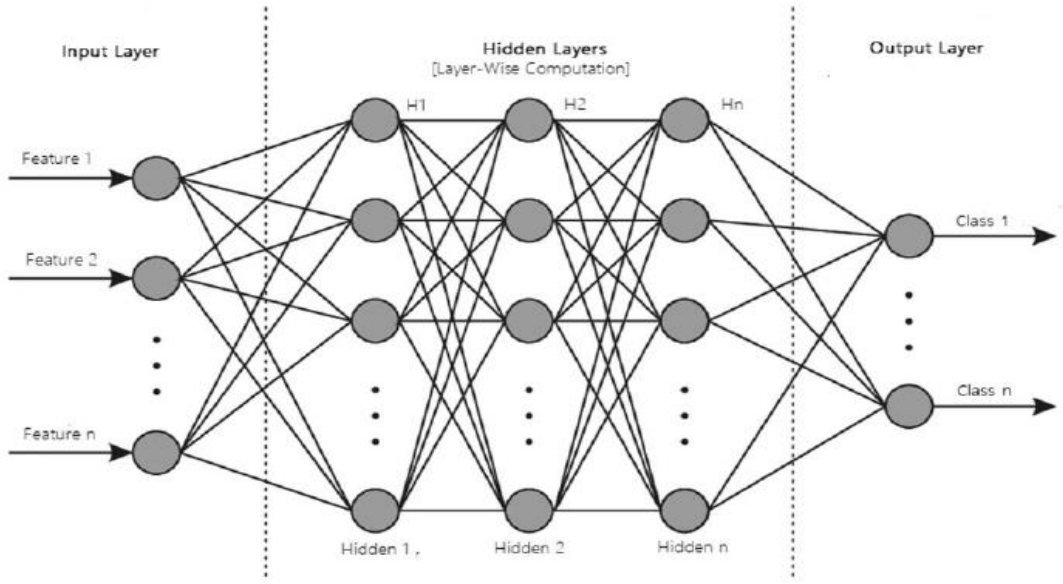


Figure 21: An example of Artificial neural network with multiple processing layers.[23]

Support Vector Machines (SVMs)

A support vector machine (SVM) is a supervised learning method that may be used to solve a variety of classification and regression problems. The SVM algorithm's goal is to create a hyperplane that optimally separates data points of one class from those of another class.

In high- or infinite-dimensional space, a support vector machine constructs a hyper-plane or set of hyper-planes. Intuitively, the hyper-plane, which has the highest distance from the

nearest training data points in any class, achieves a strong separation since, in general, the larger the margin, the lower the classifier's generalization error.

It works well in high-dimensional environments and might act differently depending on the mathematical functions known as the kernel. The most common kernel functions employed in SVM classifiers include linear, polynomial, radial basis function (RBF), sigmoid, and so on. SVM does not perform well when the data set contains additional noise, such as overlapping target classes.[23]

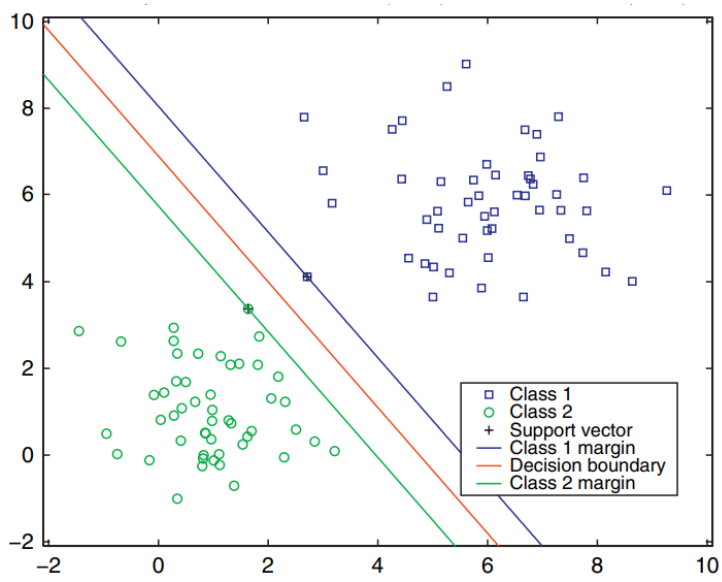


Figure 22: An example of support vector machine (SVM) with margin and support vectors for a perfectly separated on dataset.[31]

Most used applications signal processing medical applications, natural language processing, and speech and image recognition.[31]

3.3 Model Evaluation

The logical question is: How can we determine whether a model is good or bad? It doesn't necessarily follow that whatever we created using a well-known algorithm would perform well. The solution to these issues is model assessment, which is a key part of the entire machine learning workflow.

Metrics for evaluating models will differ depending on the sort of model we have, therefore

metrics for classification or clustering models will be different from metrics for regression models.[35]

- **Confusion Matrix:** One of the most common ways to evaluate a classification model is the confusion matrix. By comparing a data point's actual class label with its predicted class label, a confusion matrix is produced. [33]

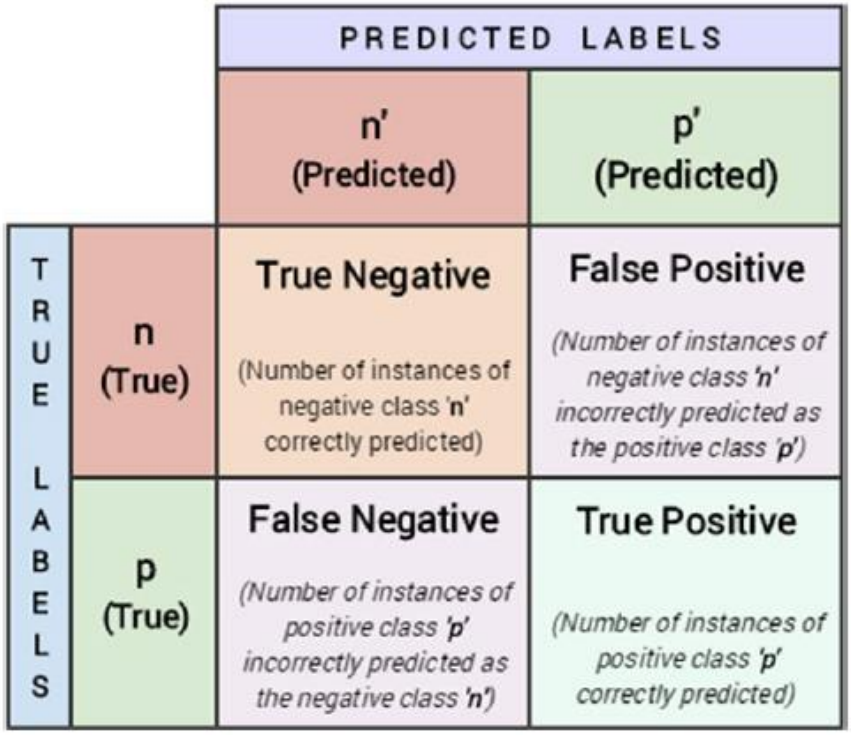


Figure 23: A typical Structure of confusion matrix. [35]

- **Performance Metrics:** The confusion matrix is not a performance metric for classification models in and of itself. However, it may be used to compute a number of metrics that are relevant in a variety of contexts.
    i. **Accuracy:** This is one of the most widely used metrics for classifier performance. It is defined as the model's total accuracy or the percentage of accurate predictions. The confusion matrix's accuracy can be calculated using the following formula:

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

When our classes are almost evenly distributed and the accuracy of those classes' predictions is important, accuracy measure is commonly utilized.[35]

ii. **Precision:** The model's precision is defined as the ratio of cases accurately identified as positive to all cases positively classified.

Precision = $\frac{TP}{TP+FP}$

In comparison to a model with lower precision, a high precision model will identify a greater percentage of the positive class. When maximizing the number of positive classes is our primary objective, precision becomes essential. [33, 35]

iii. **Recall:** The percentage of relevant data points is measured by a model's recall. It is described as the quantity of accurately predicted cases of the positive class.

Recall = $\frac{TP}{TP+FN}$ [33]

iv. **F1 Score:** Both accuracy and recall should be optimized in a balanced way. A statistic called the F1 score, which is the harmonic mean of precision and recall, helps in classifier optimization for balanced precision and recall performance.

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

The lowest recall and accuracy indicate the poorest F 1-score, which is represented by the number 0. [35, 33]

3.4 When Use of Machine Learning [28]

As its acceptance in the business rises, machine learning has proven to be a strong tool for a wide range of challenges. Despite the huge amount of interest and buzz produced by individuals both inside and outside the industry, machine learning (ML) is not a magical instrument that can fix all issues. Even for situations that ML can handle, ML solutions may not be the best options. Before starting on an ML project, consider if it is required and cost-effective.

1. Learn (the system has capacity to learn): A relational database is not an ML system since it cannot learn. A relational database can clearly declare the relationship between two columns, but it is unlikely to be capable of determining the relationship between these two columns on its own.

There must be something for an ML system to learn from for it to learn. In most circumstances, ML systems learn from data. For example, if you want to train an ML system to predict the rental price of an Airbnb listing, you must give a dataset with each input being a listing with all of its features (square footage, number of rooms, area, amenities, rating of that listing, and so on) and the corresponding output being the rental price.

2. Complex (the patterns are complex): Consider a service like Airbnb, which has a large number of housing listings, each with its own zip code. You wouldn't require an ML system to arrange listings by state. Because the pattern is straightforward, each zip code matches to a recognized state, we can simply utilize a lookup table.

Machine learning has shown great potential in tasks requiring complicated patterns, such as object identification and speech recognition. What is complicated for robots is not the same as what is difficult for people. Many jobs that are difficult for humans are simple for machines.

3. Patterns (there are pattern to learn): Machine learning solutions are only useful when there exist patterns to learn. Because there is no pattern in how these results are formed, rational individuals do not spend money in developing an ML system to forecast the future outcome of a fair die.

However, there are patterns in how stocks are valued, and corporations have spent billions of dollars developing ML systems to understand those patterns.

It is possible that a pattern does not exist, or that if patterns do exist, your dataset is insufficient to capture them. For example, there might be a trend in how Elon Musk's tweets impact Bitcoin prices. We won't know for sure until we've carefully trained and validated ML models on his tweets. Even if all models fail to generate good predictions of Bitcoin values, this doesn't exclude the possibility of a pattern.

3.5 Machine Learning Use Cases in Business

According to a recent survey, 67% of businesses use machine learning. Machine learning is becoming more popular in business and consumer applications. Since the mid-2010s, there has been a flow of applications that use ML to provide users with improved or previously unachievable services.[29] According to Algorithmia's 2020 status of business machine learning report, Machine learning applications in companies are diverse, providing both internal use cases enhancing customer experience, retaining customers, and communicating with customers as well as external use cases improving customer experience, producing customer insights, and internal processes automation.

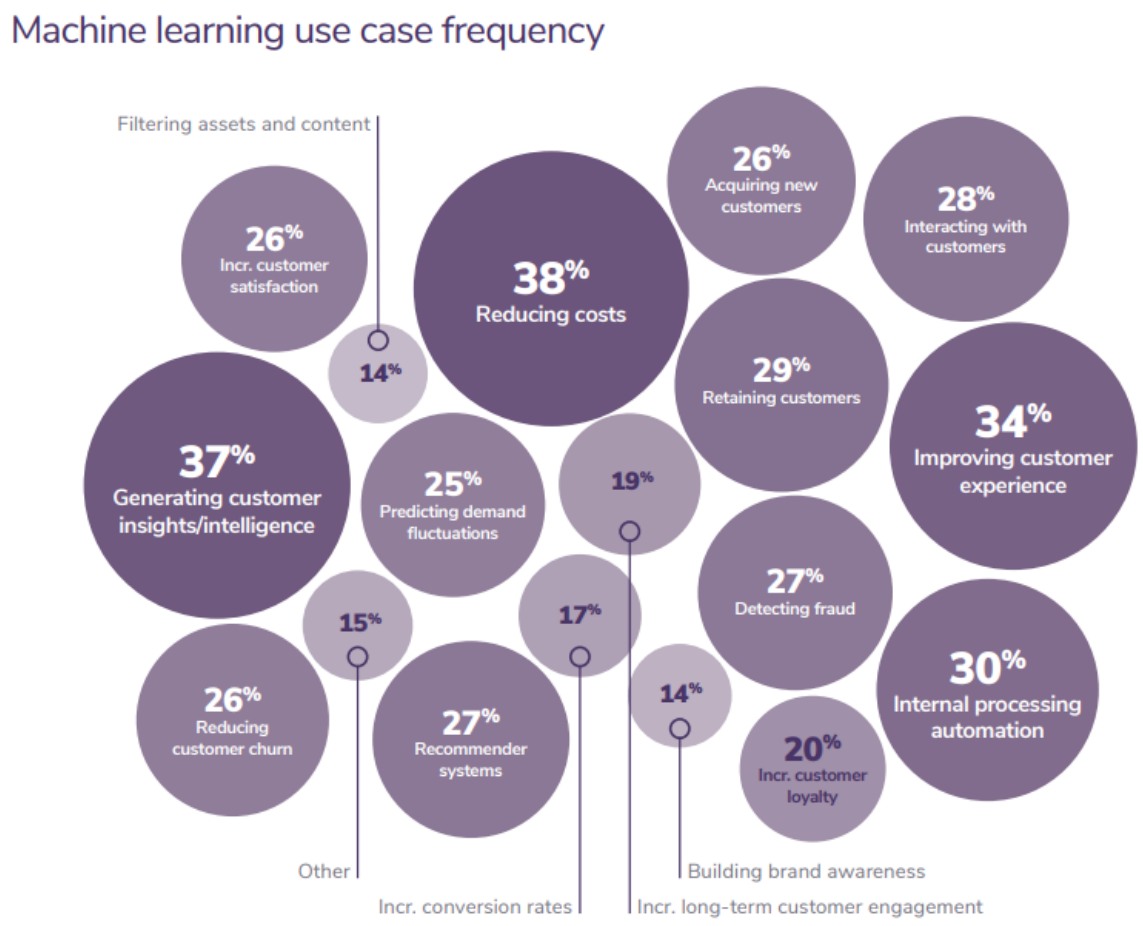


Figure 24: According to Survey by Algorithmia in 2020, state of enterprise machine learning.

Search engine and recommendation system: Machine learning is at the core of some business models, such as Netflix's recommendations algorithm or Google's search engine. Many businesses are strongly involved in machine learning, even if it is not their main business model.

Customers are given recommendations for products when they visit websites like Amazon or Netflix that are thought to best fit their interests. Customers search results are likely to be powered by machine learning if we decide to search for particular things if we don't like any of the system recommendations. [28, 29]

Smart Application: Predictive typing makes mobile phone typing more convenient. A machine learning algorithm makes suggestions for potential following sentences. Automatic translation from one language to another using machine translation. It may eliminate the language barrier between individuals of various cultures and enable communication.

With the introduction of intelligent personal assistants like Alexa and Google Assistant, machine learning is becoming more popular in our homes. Smart security cameras can alert us to visitors or when pets leave the house. [28, 29]

Forecast Customer Demand and Price Optimization: To manage a business, it's critical to be able to forecast customer demand so that the company can set a budget, store inventory, allocate resources, and update pricing strategy. For example, if you own a grocery shop, you want to stock enough items so that customers can find what they're searching for, but you don't want to overstock since your groceries will go bad and you will lose revenue.

Pricing estimation is the process of targeting to maximize a specified target function, such as a company's profit, revenue, or growth rate. Machine learning-based pricing optimization is most suited for situations with a high volume of transactions, such as online advertisements, reservations for flights, hotels, and rides, as well as situations where consumers are willing to pay a fluctuating price. [28, 29]

Churn Prediction: Churn prediction is the forecast of when a particular customer is about to stop using your products or services so that you may take suitable efforts to get them back. Churn prediction may be applied not just for customers but also for staff.

It is important to keep customers satisfied and stop them from leaving by resolving issues as soon as they arise. Previously, when a consumer opened a support ticket or sent an email, it had to be processed first, then routed via many departments until it reached the inbox of someone who could handle it. A machine learning algorithm can assess ticket content and forecast where it should go, reducing response time and increasing customer satisfaction. It's also useful for categorizing internal IT tickets. [28, 29]

Chatbots: Many businesses are using online chatbots, in which consumers or clients engage with a machine rather than with humans. These algorithms make use of machine learning and natural language processing, with bots learning from previous discussions to provide suitable replies. [28, 29]

Brand Monitoring: Brand monitoring is another popular application of machine learning in the industry. A company's brand is a valuable asset. It is essential to monitor how the general public and consumers view the brand. You could be interested in knowing when/where/how it's referenced, both openly (for example, when someone says "Google") and implicitly (for example, when someone says "the search giant"), as well as the attitude connected with it. If there is a rapid increase in unfavorable sentiment in brand references. A common machine learning problem is sentiment analysis. [28, 29]

Construction Industry: Concrete is widely utilized in the building sector, causing environmental issues like as energy use, natural resource reduction, and greenhouse gas emissions (Naseri, Jahanbakhsh, Hosseini, & Nejad, 2020). Researchers have developed machine approaches for predicting critical mechanical properties of concrete, such as compressive strength, in order to evaluate its quality (Naseri et al,, 2020)

The higher the compressive strength of the concrete machine, the less energy and material consumed, the less greenhouse gas (embodied CO2) emitted, and, ultimately, the lower the production budget required. Six ML algorithms were developed to anticipate the compressive strength of eco-friendly concrete (Naseri et al., 2020). These include the soccer league competition algorithm, the water cycle algorithm (WCA), the artificial neural network (ANN), the genetic algorithm (GA), regression, and the support vector machine (SVM) [30]

4. Python for Machine learning and Data mining

In the past, machine learning research and industrial development have been made possible by a variety of other programming languages and environments. However, the scientific computing community has greatly increased its use of the general-purpose Python language over the past ten years, and as a result, the majority of current machine learning and deep learning packages are now Python-based. According to a recent KDnuggets poll of over 1800 participants for preferences in analytics, data science, and machine learning, Python remained the most widely used language in 2019.

Python libraries are available for data loading, visualization, analytics, natural language processing, image processing, and other tasks. This extensive toolset offers data scientists a wide range of general- and special-purpose features. One of the primary benefits of Python is the ability to interact directly with the code via a terminal or other tools such as the Jupyter Notebook. Data mining and machine learning are essentially iterative processes in which the data drives the analysis. It is important for these processes to have tools that allow for rapid iteration and simple involvement. [33, 34, 32]

4.1 Python Ecosystem for Machine Learning and Analytics

Machine learning and scientific computing systems frequently use linear algebra operations on multidimensional arrays, which are computational data structures used to represent higher order vectors, matrices, and tensors.

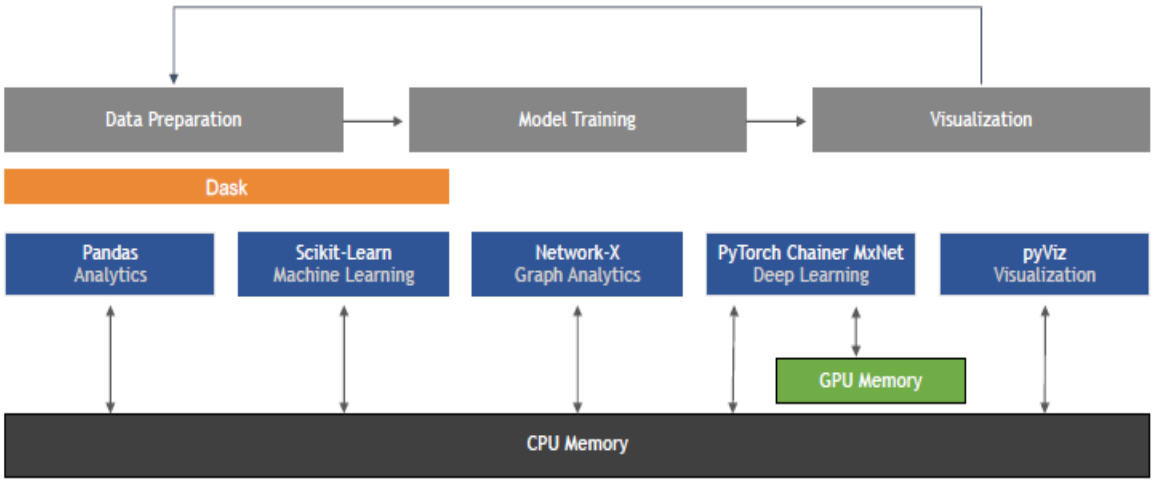


Figure 25: A typical python ecosystem for scientific computing, data science, and machine learning.[34]

Because these processes may frequently be parallelized over several processor cores, libraries like as NumPy and SciPy use C/C++ implementations to overcome threading and other python concerns.[34]

4.1.1 Jupyter Notebook

Jupyter Notebook, a web-based interactive programming environment, will be introduced with popular libraries NumPy, pandas, and matplotlib. Users may create and arrange workflows in data science, scientific computing, computational reporting, and machine learning using its versatile interface. The Jupyter Notebook makes it simple to combine code, text, and graphics.

In a single document, Jupyter notebooks contain software code, computational output, explanatory text, and rich content. In-browser code editing, code execution, and computation output are all possible with notebooks. With [.ipynb] extension, a notebook is stored. The Jupyter Notebook project supports a wide range of computer languages, including Julia (Ju), Python (Py), and R. [37]

4.1.2 NumPy Library

NumPy is one of the most important open-source python packages for scientific computing. It has support for multidimensional arrays, as well as high-level mathematical functions including linear algebra operations and the fourier transform, as well as pseudorandom number generators.

NumPy's basic capability is the ndarray class, which is a multidimensional (n-dimensional) array. The array's items must all be of the same type. The following is an example of a NumPy array:

In [2]:

```
import numpy as np
x = np.array ([[1,  2,  3],  [4,  5,  6]])
print ("x: \n {} " .format(x))
```

Out [2]:

```
x:
[[1 2  3]
[4  5  6]]
```

For example, according to NumPy case study, NumPy has important role in cricket analytics as in the context of various player and game strategies, numerical capabilities assist in identifying the statistical importance of observational data or match occurrences, estimating the game outcome by comparison with a dynamic or static model. [32, 34, 33]

4.1.3 SciPy Library

SciPy is a Python library of utilities for scientific computing. It has advanced linear algebra procedures, mathematical function optimization, signal processing, specific mathematical functions, and statistical distributions, among other features.

Scikit-learn implements its algorithms using SciPy's library of functions. For us, the most important aspect of SciPy is scipy.sparse: This produces sparse matrices, which are another data structure used in scikit learn.[32]

4.1.4 Matplotlib Library

Matplotlib, Seaborn, Bokeh, and Altair are popular data visualization libraries in the python machine learning community. Matplotlib is Python's main scientific plotting package. It includes tools for creating high-quality visualizations such as line charts, histograms, and scatter plots. Visualizing data and other elements of analysis may provide valuable insights, and we will use matplotlib for all of our visualizations. When working within the Jupyter Notebook, the **%matplotlib** notebook and **%matplotlib inline** commands can show figures directly in the browser. [32, 34]

For example, this code generates the graph shown in figure 25,

In[6]:

```
%matplotlib inline
import matplotlib.pyplot as plt

# Generate a sequence of numbers from -10 to 10 with 100 steps in between
x = np.linspace(-10, 10, 100)

# Create a second array using sine
```

```
y = np.sin(x)

# The plot function makes a line chart of one array against another
plt.plot(x, y, marker="x")
```

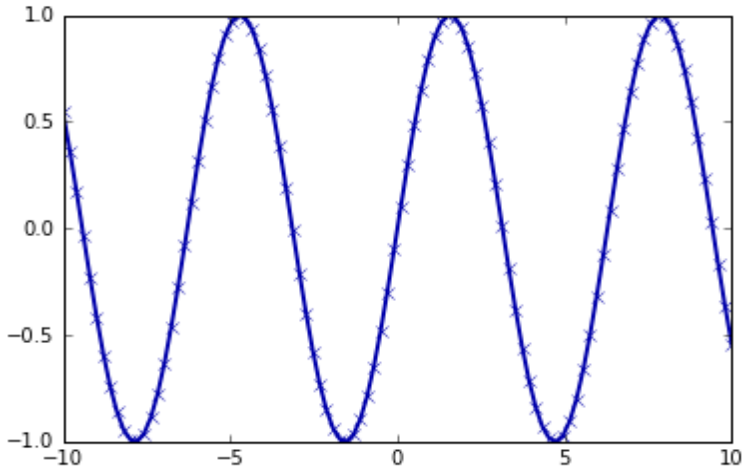


Figure 26: Matplotlib is used to create a basic line plot of the Sine(sin) function.[32]

4.1.5 Pandas Library

Pandas is a Python data manipulation and analysis toolkit. It is based on a data structure known as the DataFrame. A pandas DataFrame is essentially a table, comparable to an excel spreadsheet. Pandas provides a wide number of ways for modifying and operating on this data, including SQL-like queries and table joins. Unlike NumPy, which requires all array elements to be of the same type, pandas permit each column to have its own type.

Unlike NumPy, which requires that all array elements be of the same type, pandas permit each column to be of a different type (for example, integers, dates, floating-point numbers, and texts). Pandas also has the capacity to ingest data from a wide range of file types and databases, such as SQL, Excel files, and comma-separated values (CSV) files. Pandas has been the de-facto standard for encoding tabular data in Python for extract, transform, load" (ETL) scenarios and data analysis in recent years. [32, 34]

Here's a simple example of utilizing a dictionary to create a DataFrame:

In[7]:

```
import pandas as pd
```

```
# Create a simple dataset of people
data =
{'Name': ["John", "Anna", "Peter", "Linda"],
'Location' : ["New York", "Paris", "Berlin", "London"],
'Age' : [24, 13, 53, 33]
}
data_pandas = pd.DataFrame(data)

# IPython.display allows "pretty printing" of datagrams in the Jupyter notebook

display(data_pandas)
```

This code produces the following output:

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

[32, 34]

4.1.6 Scikit-Learn Library

Scikit-learn is a popular and essential Python machine learning and data science program (Pedregosa et al., 2011). This is true for a wide range of machine learning techniques, including core machine learning topics like classification, regression, and clustering. All of the fundamental machine learning methods, including as support vector machines, logistic regression, random forests, K-means clustering, and hierarchical clustering, were included in the library, which is likely the cornerstone for practical machine learning. Scikit-learn is mostly written in Python, however part of the core code is written in Cython for improved efficiency.

Scikit-learn is dependent on two additional Python libraries, NumPy and SciPy. We should additionally install matplotlib for visualization and interactive development.

Scikit-learn contains a first-class API for unifying the creation and execution of machine learning pipelines, in addition to its various classes for data processing and modeling known as estimators. It enables the execution of a set of estimators that includes data processing,

feature engineering, and modeling estimators. In addition, Scikit-learn has an API for testing trained models using popular approaches such as cross validation. [33, 34 32]

Scikit-learn ecosystem, allowing users to inherit advanced Scikit-learn capabilities such as pipelining and cross-validation as below,

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn import datasets
from sklearn.model_selection import train_test_split

iris = datasets.load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test =\
    train_test_split(X, y, test_size=0.3,
                     random_state=42, stratify=y)

pipe = make_pipeline(StandardScaler(),
                     PCA(n_components=2),
                     SVC(kernel='linear'))

pipe.fit(X_train, y_train)
y_pred = pipe.predict(X_test)
print('Test Accuracy: %.3f' % pipe.score(X_test, y_test))
```

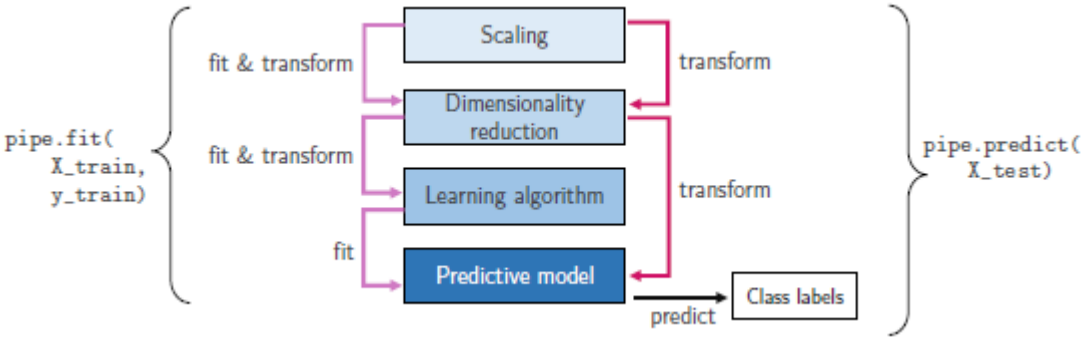


Figure 27: An example of Scikit-learn pipeline. [34]

Above code is example of linear support vector machine features from the iris dataset where new features axis through principal component analysis using of pipeline objects and illustrates how the pipeline works when the fit technique is used to the training data and the predict method is applied to the test data.[34]

5. Practical Part

5.1 Project Environment

Set up an environment for my project's implementation by installing Python 3.6.8 and a Jupyter notebook. We need to install and import some Python libraries into the Jupyter notebook. My thesis goal is to use to solve business problems using data mining tasks and machine learning algorithms. The main reason for using Python is that its open-sourced libraries make it easy to manipulate data and are user-friendly. Nowadays, the majority of data scientists use Python and related environments to solve ML and DM projects.

Example of installation python library as below represents NumPy,

```
pip install numpy
```

Source: https://numpy.org/install/

Importing Libraries: Input python code

```
#For the numerical operatios
import numpy as np

#For data frame operations
import pandas as pd

#Data vizualisation
import matplotlib.pyplot as plt
import seaborn as sns

#For machine learning algorthim
import sklearn

#For handling imbalance dataset
import imblearn
```

5.1.1 Business Understanding

A finance company provides in all types of home loans. They have a presence in all urban, suburban, and rural areas. The customer first applies for a home loan, and the company then verifies the customer's loan eligibility.

The company wants to automate the loan eligibility process (in real time) based on the information provided by the customer when filling out the online application form. Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and other details are included.

To Automate process of home loan,

- Provided a dataset to identify the customer segments that are eligible for loan.
- Create a robust model by finding hidden trends and patterns.
- Trends and patterns can be utilized to predict a candidate's loan status.

To evaluate performance of model,

- Accuracy.
- Precision, Recall and F1 Score.

We can assess the most accurate model to forecast applicant's loan status.

5.1.2 Data Understanding

Load dataset: First, collect this data set from the Kaggle and stored my project environment. The dataset is first imported into Jupyter notebook using the **read.csv()** function from the pandas library.

```
In [4]: #Lets import the dataset using the read_csv function
        data = pd.read_csv('LoanData.csv')
```

Let check the shape of the dataset, where have found 614 rows and 13 columns and get columns name using the **.columns** functions,

```
In [5]: #Check the shape of dataset
        data.shape
Out[5]: (614, 13)
```

```
In [6]: #Check the column names present in the dataset.

        data.columns
```

```
Out[6]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
               'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

Now let's check the top five row of the data using with **.head** function, which will help us to check the values present in each of the columns.

```
In [8]: #Lets check head of dataset.
        data.head()
```

Out[8]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loa
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	

As above output, we know that loan status column is the target column, and the remaining columns, aside from the loan status column is known as independent column.

Execute describe function: Descriptive statistics summarize or describe the characteristics of a dataset. Its measurements of central tendency explain the data set's center (mean, median, mode) and measures of variability describe the data set's dispersion (variance, standard deviation) as well as frequency distribution within the data set (count).[38]

In our dataset two types of variables contains one is numerical or quantitative data and another is categorial or qualitative data. First step, we must check the descriptive statistics using **describe()** function for the numerical data or continuous data, whose values either integer or float.

Descriptive Statistics--

```
In [9]: #For numerical variables/data in the dataset
        data.describe()
```

Out[9]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

There has some interesting insight from this output, It is quite clear that in the columns, applicant income, co-applicant income and loan amount having outliers. Mean value and maximum value of applicant income has huge differences similarly co-applicant income column and loan amount value column. Outliers consider very bad for predictive models as outliers creates lower performance level of models and destroy learning pattern of the data.

Now, execute descriptive statistics on categorical columns or columns with object data. We will use the same **describe()** function again, but this time we will add include is equal to object,

```
In [10]: #For categorial variables
         data.describe(include = 'object')
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
count	614	601	611	599	614	582	614	614
unique	614	2	2	4	2	2	3	2
top	LP001002	Male	Yes	0	Graduate	No	Semiurban	Y
freq	1	489	398	345	480	500	233	422

Now we have the total number of records in the column, the unique number of records in the column, the top category, which is the category with the highest occurrence among the other categories in the columns, and the frequency of the top category.

Similarly, we can now understand the dynamics and statistics of the detail set's categorical columns, such as the loan status column. This indicates that there are 614 entries, and two distinct values present in this column, which we already know will be 'Y', which indicates that the loan is given, and N, which indicates that the loan is not granted.

```
In [15]: data['Loan_Status'].value_counts()

Out[15]: Y    422
         N    192
         Name: Loan_Status, dtype: int64
```

The frequency 'Y' has 422 records out of the total 614 recordings in this test set. Using the **value_counts()** function, we can observe that the number of records is more than any record. We might also claim that the data is imbalanced. When we utilize machine learning models within balanced classes, we usually get very bad results that are fully biased towards a class with a higher distribution.

5.1.3 Data Preparation

Data Cleaning: We will clean up the dataset before proceeding with the development of our predictive models. With dirty or unclean data, we cannot build a predictive model. Missing values and outlier values are regarded as dirty or unclean data. If there are any missing values in the data set, we can use the **ISNULL** function in combination with the SUM function to determine the number of missing values in each of the columns in our data set.

```
In [16]: #checking the missing values on dataset,
         data.isnull().sum()
```

```
Out[16]: Loan_ID              0
         Gender              13
         Married              3
         Dependents          15
         Education            0
         Self_Employed       32
         ApplicantIncome      0
         CoapplicantIncome    0
         LoanAmount          22
         Loan_Amount_Term    14
         Credit_History      50
         Property_Area        0
         Loan_Status          0
         dtype: int64
```

As we can see output, many of the columns in the data set have missing values. In that case, we can impute or replace the missing values with statistical values such as mean, median, or mode.

Imputing Missing Values: The missing values in the categorical columns are usually imputed or replaced using the **mode** values.

```
#lets impute/replace mode values to categorical columns,
data['Gender'] = data['Gender'].fillna(data['Gender'].mode()[0])
data['Married'] = data['Married'].fillna(data['Married'].mode()[0])
data['Dependents'] = data['Dependents'].fillna(data['Dependents'].mode()[0])
data['Self_Employed'] = data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])
```

The loan amount, loan amount term, and credit history are also identified to be numerical columns. Therefore, all the missing values in these columns will be replaced with the **median** value,

```
#lets impute/replace median values to numerical columns,
data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].median())
data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].median())
data['Credit_History'] = data['Credit_History'].fillna(data['Credit_History'].median())
```

Let's see how many missing values there are in the dataset.

```
data.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

As we can see, the output is zero, indicating that we have successfully replaced all of the missing values in the dataset.

Visualization Outliers: Descriptive statistics shown that, the columns applicant income, co-applicant income, and loan amount contain outliers. Let's use the box plot to visualize the outliers in these columns. We can see that there are a lot of outliers in these columns, so let's remove them from the dataset to make our data clean for modeling.

```
#Visualize the outlier using box plot,


plt.style.use('ggplot')
plt.rcParams['figure.figsize'] = (7, 6)

plt.subplot(2, 2, 1)
sns.boxplot(data['ApplicantIncome'])
plt.xlabel('Applicant Income', fontsize=10)

plt.subplot(2, 2, 2)
sns.boxplot(data['CoapplicantIncome'])
plt.xlabel('Coapplicant Income', fontsize=10)

plt.subplot(2, 2, 3)
sns.boxplot(data['LoanAmount'])
plt.xlabel('Loan Amount', fontsize=10)

plt.suptitle('Findings Outliers in Data')
plt.show()
```

Output:

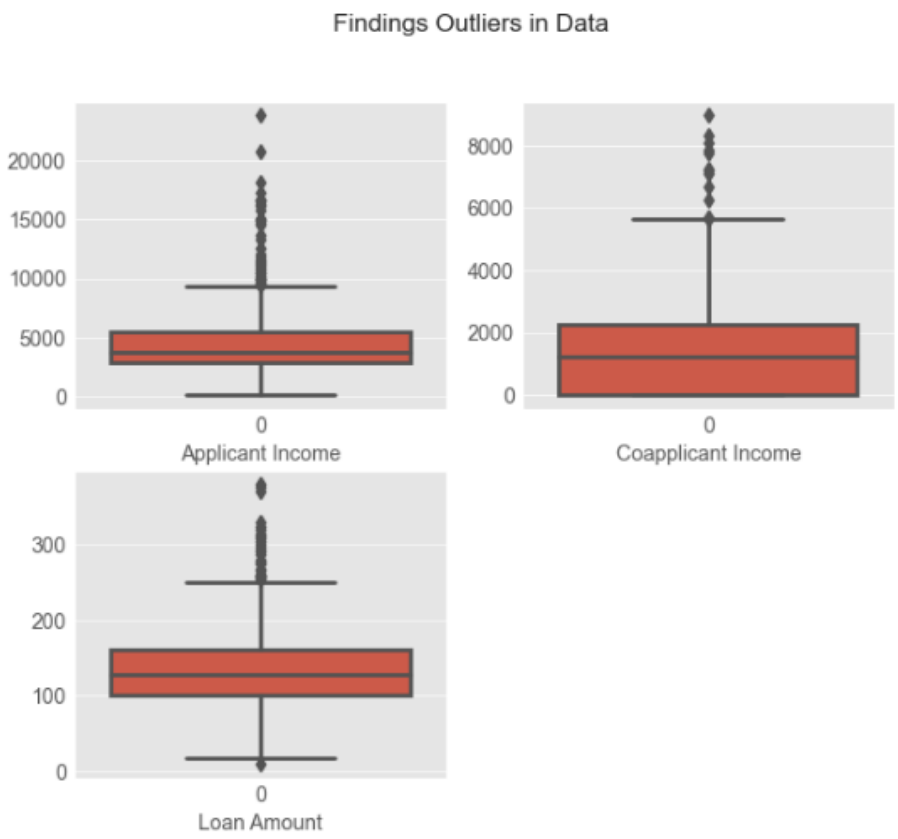


Figure 28: Box plot help us to visualization the outliers.

We can see that there are a lot of outliers in these columns, so let's remove them from the data set to make our data clean for modeling.

Remove Outliers: First, let's remove all the records from the applicant income column that have a value more than 25,000 from the data,

```
#Lets remove the outliers from the data.

#check the data shape before removing the outliers
print("Before removing outliers", data.shape)

#lets eliminate/filter the data having more that 25000 income,
data = data[data['ApplicantIncome'] < 25000]

#After removing of the outliers,
print("After removing outliers", data.shape)
```

```
Before removing outliers (614, 13)
After removing outliers (607, 13)
```

We can observe that the total number of records before eliminating these records was 614. After eliminating it became 607, indicating that there are seven outliers in this column.

After that, we will eliminate any records with values more than 10,000 from the co-applicant income,

```
#Lets remove outliers from co-appilcant income column,

#lets check the shape of data defore removing outliers,
print("Before removing outliers", data.shape)

#lets eliminating the customers having more than 10,000 coapplicant income,
data=data[data['CoapplicantIncome']<10000]

#Check the records shape of data after removing the outliers,
print("After removing outliers", data.shape)
```

```
Before removing outliers (607, 13)
After removing outliers (601, 13)
```

Finally, we will remove any records with loan amount values more than 400,

```
#Lets remove outliers from loan amount income column,

#lets check the shape of data defore removing outliers,
print("Before removing outliers", data.shape)

#lets eliminating the customers having more than 400 loan amount,
data=data[data['LoanAmount'] < 400 ]

#Check the records shape of data after removing the outliers,
print("After removing outliers", data.shape)
```

```
Before removing outliers (601, 13)
After removing outliers (590, 13)
```

After eliminating all of the outliers from the dataset, the total number of records in the dataset is 590.

Distribution of data using univariate analysis:[37] In our dataset, we may perform a univariate analysis and discover some interesting facts about the features in the details.

- Univariate analysis is the most basic type of statistical analysis, and it can be influencing or descriptive.
- The important point is that just one variable is involved.
- When multivariate analysis is more suited, if univariate analysis can produce misleading results.

This is a necessary step in understanding the variables in the data set,

- Analyze the numerical column distribution in the dataset.
- Use pie charts and count plots to analyze the distribution of categorical columns.
- Use pie charts when there are few categories in the categorical column and count plots when there are many.

Let's check the distribution of three numerical columns first: applicant income, co-applicant income, and loan amount. To plot all three of these columns together, we are using the **subplot()** function from the matplotlib library, which is part of the seaborn library for data visualization,

```
# Univariate Analysis on Numerical Columns

plt.style.use('ggplot')
plt.rcParams['figure.figsize'] = (8, 7)

plt.subplot(2, 2, 1)
sns.distplot(data['ApplicantIncome'], color = 'green')

plt.subplot(2, 2, 2)
sns.distplot(data['CoapplicantIncome'], color = 'green')

plt.subplot(2, 2, 3)
sns.distplot(data['LoanAmount'], color = 'green')

plt.suptitle('Univariate Analysis on Numerical Columns')

plt.show()
```

Output:

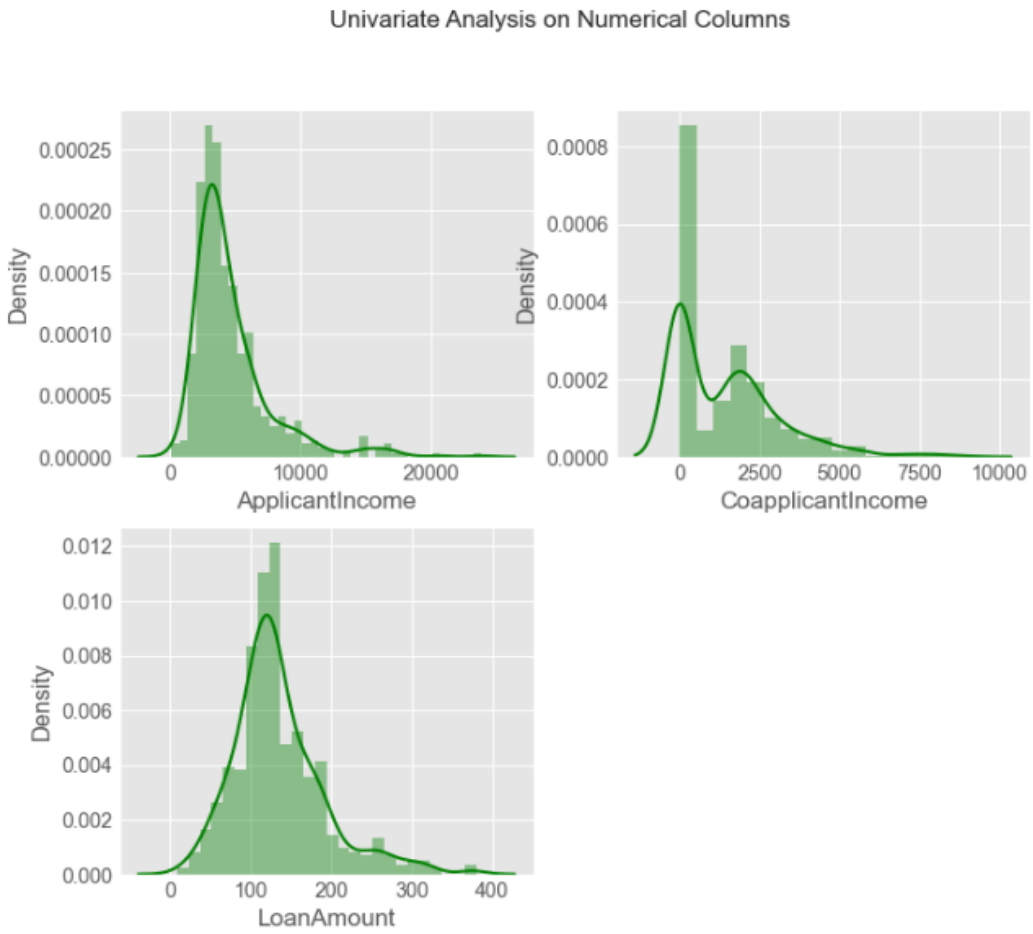


Figure 29: Distribution of Numerical Columns (applicant income, co-applicant income and loan amount).

We can see the findings, and one key finding is that all these charts are skewed in nature and can lead to misleading results when doing predictive modeling. As a result, we must use various transformation techniques to eliminate skewness from the data.

Remove Skewness using Log Transform: In this scenario, we will utilize the log transform to eliminate the skewness from these columns.

```
#Remove the skewness from the ApplicantIncome and CoapplicantIncome columns,

plt.style.use('ggplot')
plt.rcParams['figure.figsize'] = (7, 3)

#Apply log transformation to remove skewness..
data['ApplicantIncome'] = np.log1p(data['ApplicantIncome'])
data['CoapplicantIncome'] = np.log1p(data['CoapplicantIncome'])

#check the plot after implement log transformation on thses columns,n
plt.subplot(1, 2, 1)
sns.distplot(data['ApplicantIncome'], color = 'green')

plt.subplot(1, 2, 2)
sns.distplot(data['CoapplicantIncome'], color = 'green')

plt.suptitle('Implementing log Transformation')

plt.show()
```

Output:

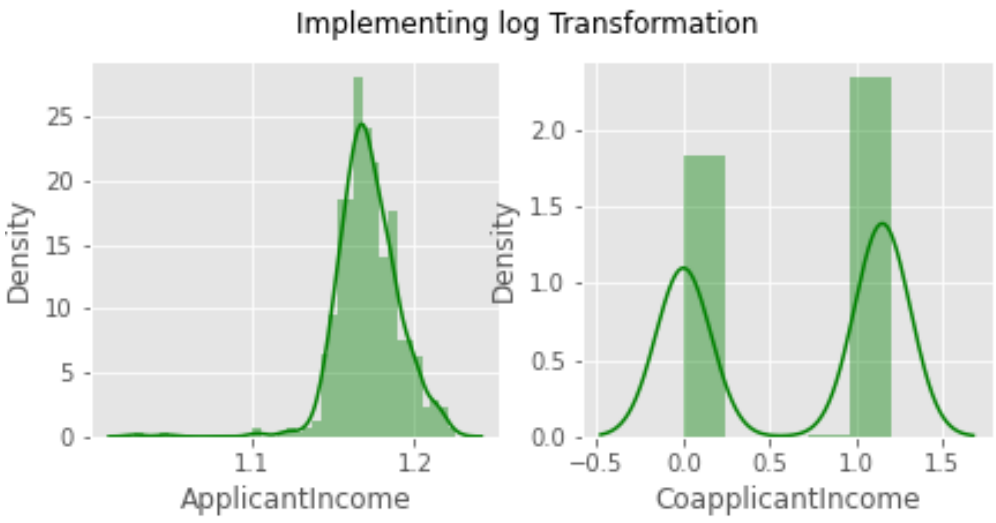


Figure 30: Applicant income and co-applicant income depicting normal distribution after log transform.

These distributions now closely look like the normal distribution. As a result, we have successfully overcome the skewness problem from numerical columns.

Next, we are going to check the distribution of categorical columns present in the detail set using the **countplot** function with **subplot**,

```
## Univariate Analysis on Categorical Columns

plt.rcParams['figure.figsize'] = (18,8)

plt.subplot(2, 4, 1)
sns.countplot(data['Gender'], palette = 'deep')

plt.subplot(2, 4, 2)
sns.countplot(data['Married'], palette = 'deep')

plt.subplot(2, 4, 3)
sns.countplot(data['Dependents'], palette = 'deep')

plt.subplot(2, 4, 4)
sns.countplot(data['Self_Employed'], palette = 'deep')

plt.subplot(2, 4, 5)
sns.countplot(data['Credit_History'], palette = 'deep')

plt.subplot(2, 4, 6)
sns.countplot(data['Property_Area'], palette = 'deep')

plt.subplot(2, 4, 7)
sns.countplot(data['Education'], palette = 'deep')

plt.subplot(2, 4, 8)
sns.countplot(data['Loan_Status'], palette = 'deep')

plt.suptitle('Univariate Analysis on Categorical Columns')
plt.show()
```

Output:

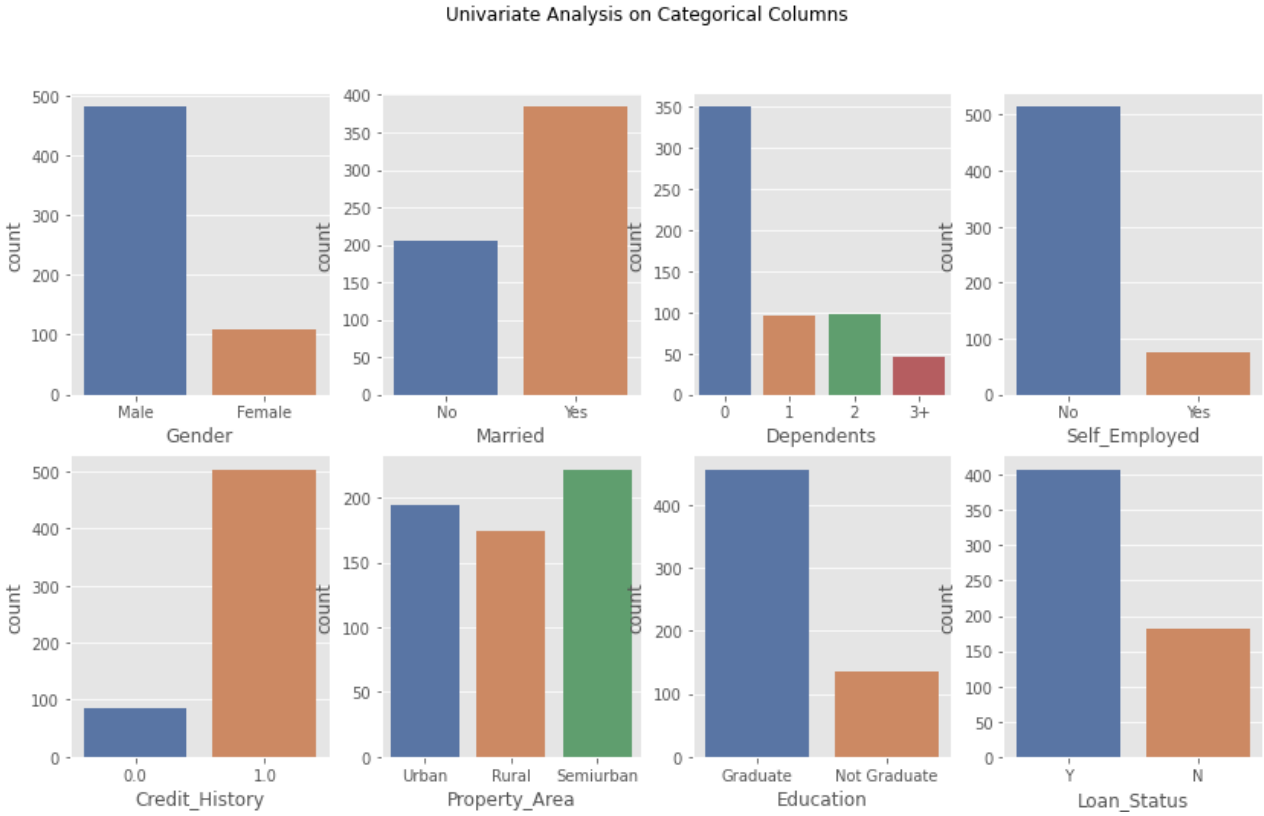


Figure 31: Visualization of distribution of categorical columns on dataset using bar plot.

We can observe the distribution of different categories in several columns. As we can observe, there are more male applications than female applicants. Similar to how married applications are more significant than single applicants, etc. Such a large variety of hidden patterns exist.

Using bivariate analysis to determine relationships between variables:[37] Bivariate analysis is one of the most basic types of quantitative analysis. It involves two variables to determine their empirical relationship and is useful in testing simple hypotheses of association.

To understand the relationship between two variables in a data set, three types of bivariate analysis can be used,

1. Categorical vs Numerical

2. Categorical vs Categorical

3. Numerical vs Numerical

Categorical vs Numerical columns analysis: As we know, our target column is categorical, and we will perform bivariate analysis only on the target data. So, first, we will analyze the impact of numerical columns on the target column using categorical vs numerical analysis.

Using a box plot, we will analyze the impact of applicant and co-applicant income on loan status.

```
#Lets check the impact of applicant income and co-applicant income on loan status,
plt.rcParams['figure.figsize'] = (10,4)

plt.subplot(1,2,1)
sns.boxplot(data['Loan_Status'], data['ApplicantIncome'], palette = 'deep' )

plt.subplot(1,2,2)
sns.boxplot( data['Loan_Status'],data['CoapplicantIncome'], palette = 'deep')

plt.suptitle('Impact of applicant & co-applicant income on loan status', fontsize=15)
plt.show()
```

Output:

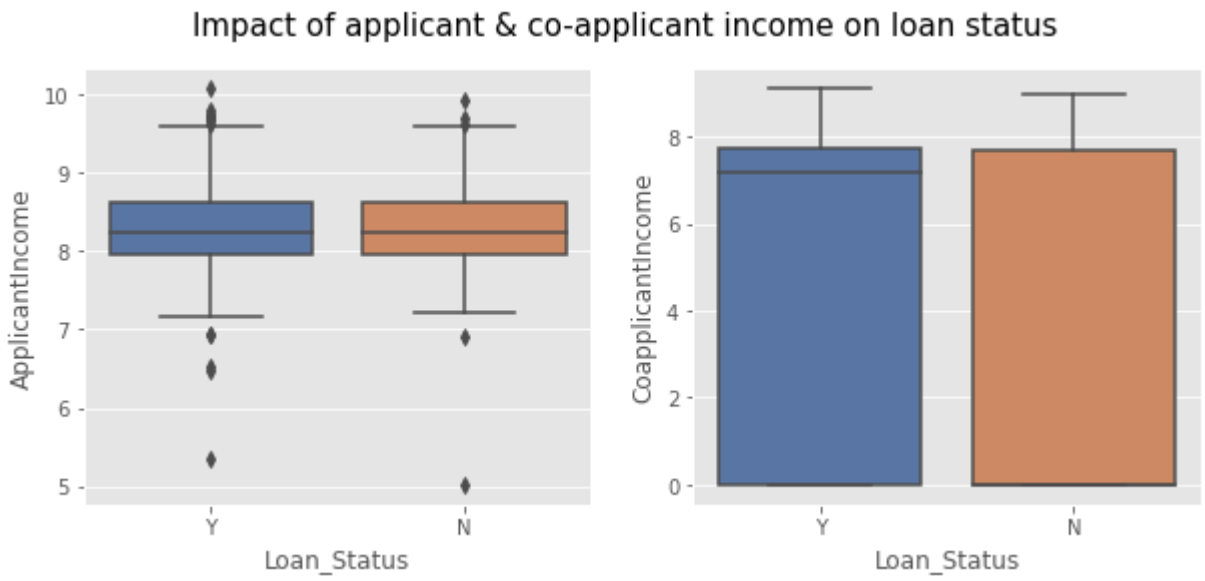


Figure 32: Impact of applicant and co-applicant income on loan status using box plot.

We can see from the plots that there is no clear pattern. Let's use the boxen plot again to analyze the impact of loan amount and loan amount term,

```
#Lets check the impact of  income on loan status,
plt.rcParams['figure.figsize'] = (10,4)

plt.subplot(1,2,1)
sns.boxenplot(data['Loan_Status'], data['LoanAmount'], palette = 'deep' )

plt.subplot(1,2,2)
sns.boxenplot( data['Loan_Status'],data['Loan_Amount_Term'], palette = 'deep')

plt.suptitle('Impact of loan amount and loan amount term on loan status', fontsize=15)
plt.show()
```

Output:

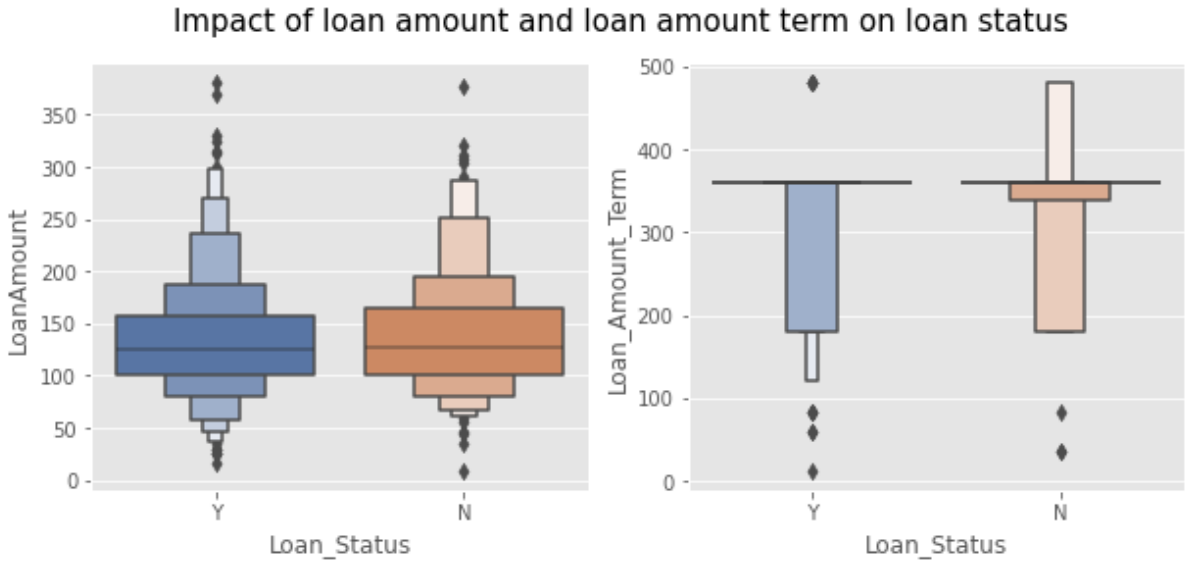


Figure 33: Impact of loan amount and loan amount term on loan status.

We can see that the loan amount has no clear pattern, but it is clear that the longer the loan amount term, the lower chances of getting a loan approved. So, we analyzed the effect of numerical columns on the target column (Loan_Status).

Categorical vs Categorical columns analysis: Now, we are checking the impact of categorical columns on the target column using the **crosstab()** function, which returns across tabulation of two categorical columns,

```
# Check all categorical columns to target column loan status,

print('Impact of marraige on loan status')
print(pd.crosstab(data['Loan_Status'], data['Married']), '\n')

print('Impact of dependents on loan status')
print(pd.crosstab(data['Loan_Status'], data['Dependents']), '\n')

print('Impact of education on loan status')
print(pd.crosstab(data['Loan_Status'], data['Education']), '\n')

print('Impact of employment on loan status')
print(pd.crosstab(data['Loan_Status'], data['Self_Employed']), '\n')

print('Impact of property area on loan status')
print(pd.crosstab(data['Loan_Status'], data['Property_Area']), '\n')

print('Impact of property area on loan status')
print(pd.crosstab(data['Loan_Status'], data['Credit_History']))
```

Output:

```
Impact of marraige on loan status
Married       No  Yes
Loan_Status
N             76  106
Y            130  278

Impact of dependents on loan status
Dependents     0   1   2  3+
Loan_Status
N            110  33  24  15
Y            240  63  74  31
```

According to the first column, married candidates had their loan rejected 76 times and approved 130 times. Similarly, the married candidates were approved 278 times and rejected 130 times.

We can now understand the impact of marriage on loan status, even though the difference in loan eligibility between married and unmarried candidates is very small.

However, married candidates have an opportunity over unmarried candidates. Likewise, there are relationships between all of the columns and the target column like graduated applicant more eligible than the non-graduate applicants.

Output:

```
Impact of education on loan status
Education    Graduate  Not Graduate
Loan_Status
N                 130            52
Y                 326            82
```

```
Impact of property area on loan status
Property_Area  Rural  Semiurban  Urban
Loan_Status
N                 66         51     65
Y                108        171    129
```

```
Impact of property area on loan status
Credit_History  0.0  1.0
Loan_Status
N                80  102
Y                 6  402
```

We can see that in the dependent column. The impact of having 1, 2, 3+ dependents is quite similar. That means we can merge these three categories. Also, we can observe that educated applicants are eligible to grant to loan and applicant with credit history 1.0 and loan approved 402 time, it seems more influential factor to approve loan. There are so many patterns hidden in these cross tabulations to understand data insights.

Visualization of Correlation: Let us now analyze the correlation between all variables. To visualize the correlation, we will use a heat map. Heatmaps use color variations to visualize data. The variables with a darker color have a stronger correlation.

```
matrix = data.corr()
f, ax = plt.subplots(figsize=(9, 11))
sns.heatmap(matrix, vmax=.8, square=True, cmap="BuPu",annot=True);
```

Output:

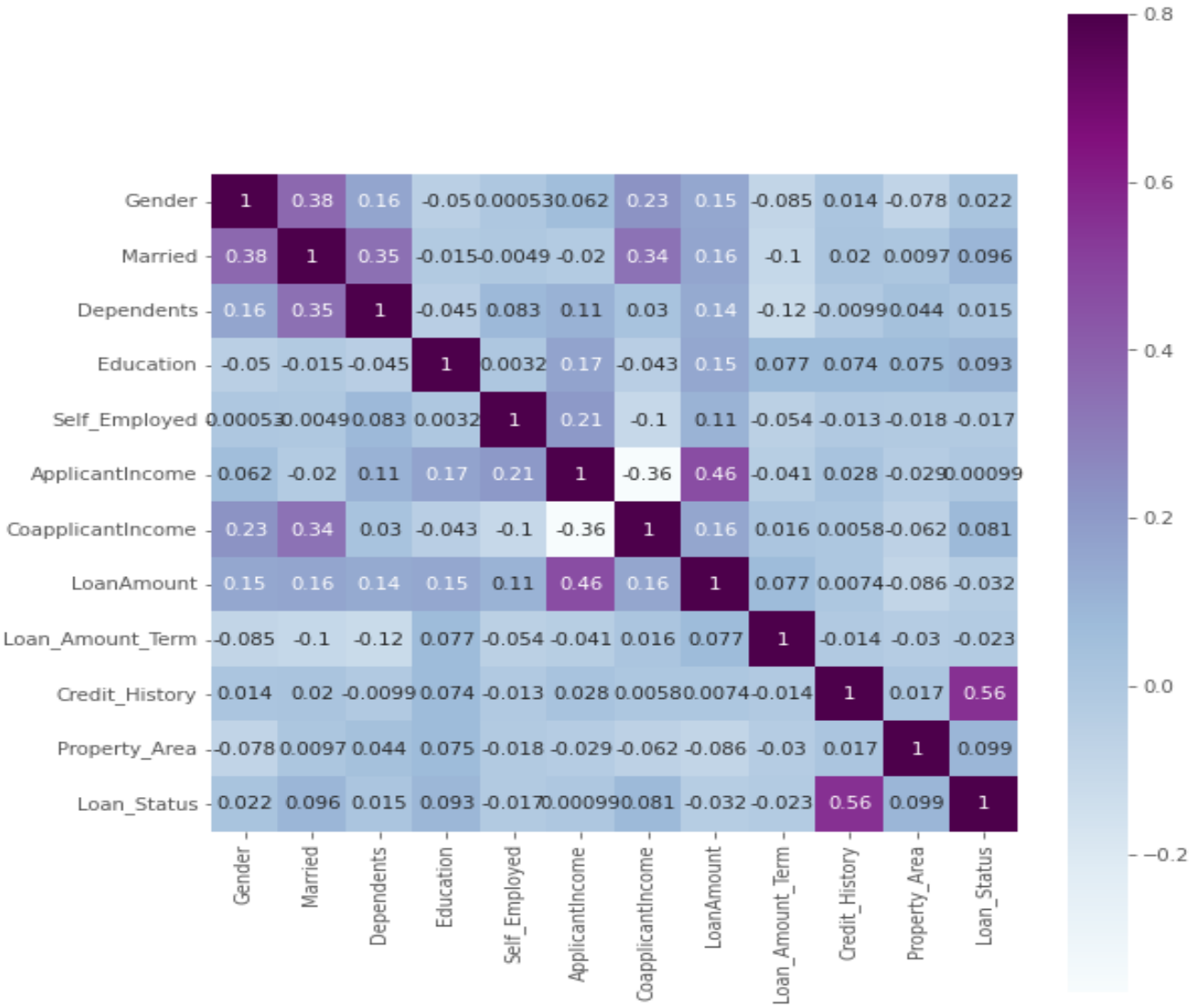


Figure 34: Correlation heatmap for understanding relationship between variables.

We can see that (ApplicantIncome - LoanAmount) and (Credit History - Loan Status) are the most correlated variables.

5.1.4 Data Preparation for Modelling

Now we will prepare the data for fitting into a machine learning model. We already know that strings cannot be accepted by a machine learning model. For that, we'll need to encode all of the categorical columns.

```
#Lets check the data type of strings / objects,
data.select_dtypes('object').head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	Urban	Y
4	LP001008	Male	No	0	Graduate	No	Urban	Y

Remove Loan_ID Column: We can see that there is a loan ID column, which is useless for predicting a candidate's loan status. So, let's use the drop function to remove this column,

```
#Now delete the Loan_ID column,
print('Before data shape: ', data.shape)

data = data.drop(['Loan_ID'], axis=1)
print('After data shape: ', data.shape)
```

```
Before data shape:  (590, 13)
After data shape:  (590, 12)
```

Encoding categorical columns: Let's now start encoding the remaining categorical columns. We will apply the **replace()** function to do that. We can see that the categories of gender, marriage, education, self-employment, and loan status are encoded with a '1' for the first category and a '0' for the second.

```
# lets encode all columns with 1 and 0,

data['Gender'] = data['Gender'].replace(('Male','Female'),(1, 0))
data['Married'] = data['Married'].replace(('Yes','No'),(1, 0))
data['Education'] = data['Education'].replace(('Graduate','Not Graduate'), (1, 0))
data['Self_Employed'] = data['Self_Employed'].replace(('Yes','No'), (1, 0))
data['Loan_Status'] = data['Loan_Status'].replace(('Y','N'), (1, 0))

#As seen above that Urban and Semi Urban Property have very similar impact on Loan Status,
data['Property_Area'] = data['Property_Area'].replace(('Urban','Semiurban', 'Rural'),(1, 1, 0))

#As seen above that apart from 0 dependents, all are similar hence, we merge them,
data['Dependents'] = data['Dependents'].replace(('0', '1', '2', '3+'), (0, 1, 1, 1))

# lets check whether there is any object column left
data.select_dtypes('object').columns
```

```
Index([], dtype='object')
```

Separate target column from dataset: Finally, we will split the target column from the dataset. It's a significant stage because if it's left in the data set, the machine learning model will pick up all the patterns from the target column as well. But as we all know, the target column is absent in real-world cases. We employ these prediction models for that reason. We can see that we have been successful in separating the target column from the data and placing it in a variable named Y.

```
# Lets split the target column from the dataset,
Y = data['Loan_Status']
x = data.drop(['Loan_Status'], axis = 1)

#Now check shape of data Y and x,
print('shape of Y target column: ', Y.shape)
print('shape of x dataset deleting target column: ', x.shape)
```

```
shape of Y target column:  (590,)
shape of x dataset deleting target column:  (590, 11)
```

Resampling: We previously observed that the target column is highly imbalanced. We must balance the data using statistical approaches. We may resample the data using a variety of statistical approaches. Resampling is a statistical method that includes drawing repeated samples from the original samples, such as over sampling, cluster-based sampling, and under sampling. In data analysis, oversampling and under sampling are techniques for adjusting the class distribution of a data set.

To minimize data loss, we will employ over sampling rather than under sampling in this scenario. The **imblearn** library will be used to perform over sampling. To perform oversampling, we will utilize the SMOTE algorithm from the imblearn library.

```
# Use Over Sampling Technique to resample the data.
# lets import the SMOTE algorithm to do the same.

from imblearn.over_sampling import SMOTE
oversample = SMOTE()

x_resample, Y_resample  = oversample.fit_resample(x, Y.ravel())

# lets check the shape of x dataset and Y column after resampling,
print('Use over sampling technique dataset x:', x_resample.shape)
print('Use over sampling technique target column Y:',  Y_resample.shape)
```

```
Use over sampling technique dataset x: (816, 11)
Use over sampling technique target column Y: (816,)
```

After implementing an over sampling, we will count the number of applicants whose loan has been refused and accepted to determine the discrepancy.

```
#Lets check the target variables after resampling,

print("Before resampling Y target value counts:")
print(Y.value_counts(), '\n')

print("After resampling Y target value counts:")
Y_resample = pd.DataFrame(Y_resample)
print(Y_resample[0].value_counts())
```

```
Before resampling Y target value counts:
1    408
0    182
Name: Loan_Status, dtype: int64

After resampling Y target value counts:
1    408
0    408
Name: 0, dtype: int64
```

We can clearly see that, before to the resampling, we had 408 applicants whose loan applications got accepted but only 108 candidates whose loan applications got rejected.

However, after resampling, the number of applicants who had their loan application granted and refused remained at 408.

Prepare train and test dataset for modeling: Finally, We also split the dataset into train and test using the **train_test_split()** utility from scikit-learn, and check the shape of all newly formed dataset to ensure they are correctly formed.

```
# lets split the test data and the training data,

from sklearn.model_selection import train_test_split

x_train, x_test, Y_train, Y_test = train_test_split(x_resample, Y_resample, test_size = 0.2, random_state = 0)

# lets print the shapes again,
print("Shape of the x Train :", x_train.shape)
print("Shape of the Y Train :", Y_train.shape)
print("Shape of the x Test :", x_test.shape)
print("Shape of the Y Test :", Y_test.shape)
```

```
Shape of the x Train : (652, 11)
Shape of the Y Train : (652, 1)
Shape of the x Test : (164, 11)
Shape of the Y Test : (164, 1)
```

We can observe that the x train and Y train contain 652 records each, but the x test and Y test have 164 records each. Also, the Y train and Y test contain only one column, which is the target column.

5.1.5 Modeling and Evaluation

Logistic Regression Modeling: Now, we are going to train dataset on a machine learning predictive model. We will implement logistic regression algorithm to logistic to train the model. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. Now let's import the logistic regression algorithm from **sklearn.linear_model** package. After that, we will define a model and then train our training dataset that is x train and Y train using the **.fit()** function.

```
# lets apply Logistic Regression

from sklearn.linear_model import LogisticRegression

model = LogisticRegression(random_state = 0)
model.fit(x_train, Y_train)

y_pred = model.predict(x_test)

print("Training Accuracy :", model.score(x_train, Y_train))
print("Testing Accuracy :", model.score(x_test, Y_test))
```

```
Training Accuracy : 0.7730061349693251
Testing Accuracy : 0.8170731707317073
```

Now, as the model has been trained on the training data. It's time to perform some predictive analysis using the predict function.

We are going to make the predictions on the testing set that is x test. Finally, we can check the training and testing accuracy using the score function. We can see that the training accuracy for a logistic regression model is around 77 percent, whereas the testing accuracy comes out to be around 81 percent.

Performance Metrics for Logistics Regression: Let's import the **confusion_matrix** and **classification_report** from the **sklearn.metrics** package and check them as well,

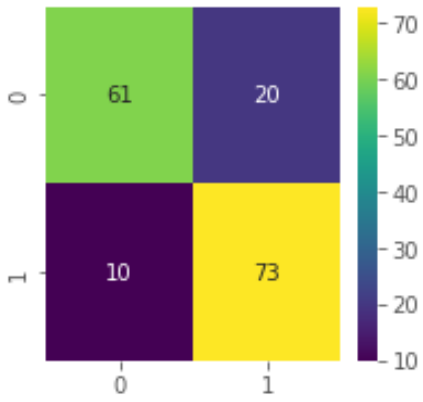
```
# lets analyze the performance using confusion matrix,

from sklearn.metrics import confusion_matrix, classification_report

cm = confusion_matrix(Y_test, y_pred)
plt.rcParams['figure.figsize'] = (3, 3)
sns.heatmap(cm, annot = True, cmap = 'viridis', fmt = '.8g')
plt.show()

# lets also use classification report for performance analysis,
cr = classification_report(Y_test, y_pred)
print(cr)
```

Output:



	precision	recall	f1-score	support
0	0.86	0.75	0.80	81
1	0.78	0.88	0.83	83
accuracy			0.82	164
macro avg	0.82	0.82	0.82	164
weighted avg	0.82	0.82	0.82	164

Figure 35: Confusion matrix for logistics regression with accuracy, precision, Recall, F1 Score.

We can see that the logistic regression model recognizes the loan rejection cases incorrectly 20 times while correctly identifying the loan acceptance cases only 10 times. If we look at the precision, we can see that class '0' has a precision of 86%, while class '1' has a precision of 78%. The recall rate for classes "0" and "1" is also 75% and 88%, respectively. This means that logistic regression is not the best model for this data set.

We trained on a logistic regression model previously, and it performed reasonably well. However, we are going to try one more machine learning algorithm that is more advanced than the logistic regression model.

Gradient Boosting Modeling: We are going to try a gradient boosting model and learn more about this model in the boosting model scores. First, let's import the gradient boosting algorithm from **sklearn.ensemble** package. After that will define a model and then train our training data set that is x train and Y train using the **.fit()** function.

```
# lets apply Gradient Boosting classifer,

from sklearn.ensemble import GradientBoostingClassifier

model =  GradientBoostingClassifier()
model.fit(x_train, Y_train)

y_pred = model.predict(x_test)

print("Training Accuracy :", model.score(x_train, Y_train))
print("Testing Accuracy :", model.score(x_test, Y_test))
```

Now as the model has been trained on the training data. It's time to perform some predictive analysis using the predict function. We're going to make the predictions on the testing set, that is x test, finally, we can check the training and testing accuracy using the score function.

Output:

```
Training Accuracy : 0.9187116564417178
Testing Accuracy : 0.8353658536585366
```

We can see that the training accuracy of a gradient boosting model is slightly higher than that of a logistic regression model.

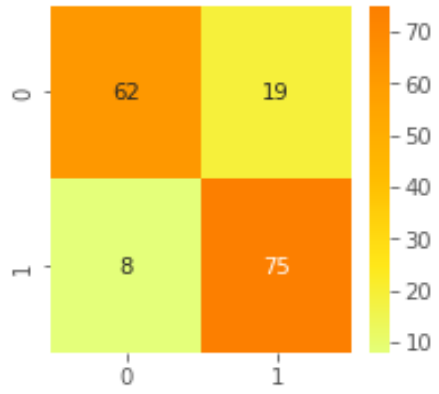
Performance Metrics for Gradient Boosting: Let us also look at the **confusion_matrix** and **classification_report** to ensure that this model performs better the previous one,

```
# lets analyze the Performance using Confusion matrix

cm = confusion_matrix(Y_test, y_pred)
plt.rcParams['figure.figsize'] = (3, 3)
sns.heatmap(cm, annot = True, cmap = 'Wistia', fmt = '.8g')
plt.show()

# lets also use classification report for performance analysis
cr = classification_report(Y_test, y_pred)
print(cr)
```

Output:



	precision	recall	f1-score	support
0	0.89	0.77	0.82	81
1	0.80	0.90	0.85	83
accuracy			0.84	164
macro avg	0.84	0.83	0.83	164
weighted avg	0.84	0.84	0.83	164

Figure 36: Confusion matrix for Gradient Boosting with accuracy, precision, Recall, F1 Score.

Let us also look at the confusion matrix and classification report to ensure that this model performs better the previous one.

If we look at the precision and recall, they are quite similar. We can check the cross-validation scores also and we can see that the scores are not bearing too much, meaning that the model has been trained well.

```
from sklearn.model_selection import cross_val_score

clf = GradientBoostingClassifier(random_state = 0)
scores = cross_val_score(clf, x_train, Y_train, cv=10)
print(scores)
```

Output:

```
[0.74242424 0.86363636 0.81538462 0.90769231 0.81538462 0.73846154
 0.8        0.8        0.83076923 0.81538462]
```

We cannot expect this model to work very well as the number of records in the training data is very less. This means that gradient boosting is a good model to go with.

Feature Importance: Let us now determine the feature importance, that is, which features are most important to solve the business problem. We will do it with sklearn's feature importance.

```
model_feature_importances = pd.Series(data= model.feature_importances_ ,index=x_train.columns)
model_feature_importances = model_feature_importances.sort_values()
model_feature_importances
```

```
Self_Employed        0.001289
Gender               0.003919
Dependents           0.008078
Loan_Amount_Term     0.018064
Married              0.018688
Education            0.025197
Property_Area        0.036805
CoapplicantIncome    0.056295
ApplicantIncome      0.107861
LoanAmount           0.157355
Credit_History       0.566449
dtype: float64
```

```
fig = px.bar(model_feature_importances)
fig.show()
```

Output:

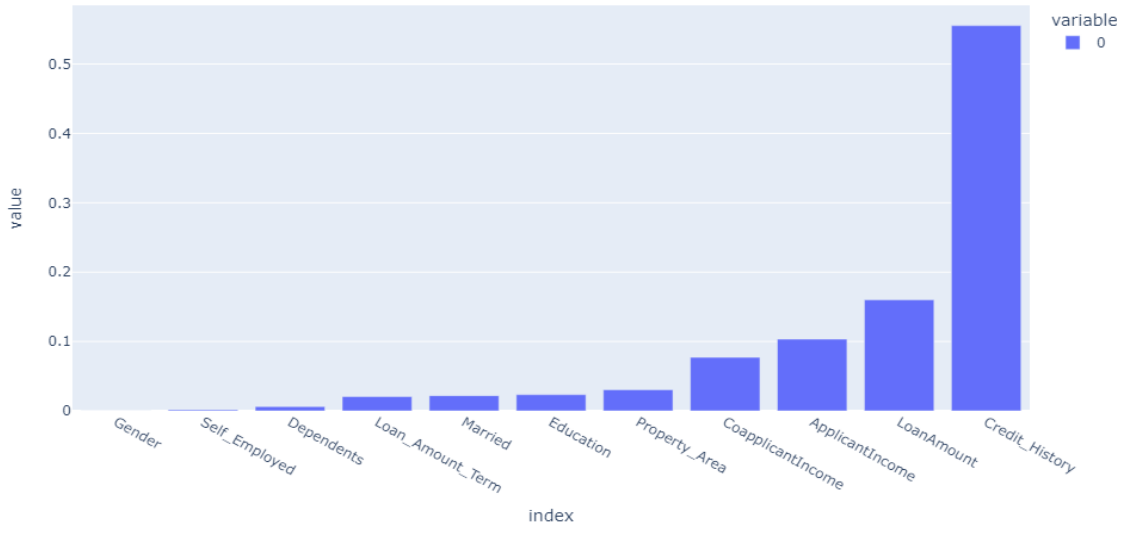


Figure 37: Feature importance obtain from our trained model.

After creating the bar plot and observe that, we can see that the 'Credit History' feature is the most important. As a result, feature engineering guided us in predicting our target variable.

5.1.6 Summaries in Practical Part

01. To clearly understand the business issue statement that must be solved by data mining tasks such as customer segment classification problem and the use of ML models to automate the system or real-time prediction system for finance organization.

02. To evaluate the general frequency of data in a dataset using descriptive statistics.

03. Determine missing data and impute/replace them with statistical values such as mean, median, or mode.

04. Using the box plot visualization, identify outliers and eliminate them from the dataset for smooth bivariate and univariate analysis.

05. In univariate analysis, find the skewness in categorical columns and eliminate it with the log transformation.

06. Use bivariate analysis to discover hidden patterns and the impact of other factors on loan status target value.

07. To use linear regression and gradient boosting techniques, separate the target variable from the dataset as well as the train and test data.

08. Our predictions are almost 0.773 accurate, i.e., we have identified 77% of the loan status correctly for our logistics regression trained model

09. And our predictions are almost 0.918 accurate, i.e., we have identified 91% of the loan status correctly for implemented gradient boosting model.

10. According to the performance matrix gradient boosting model accuracy is better than logistics regression model and which factors are more essential in granting a loan based on our findings credit history on trained ML model.

Finally, Loan prediction is a relatively common real-life challenge that every retail bank experiences. If the model is performed correctly, it can save a lot of man hours at the end of a retail bank.

Although this practical section is primarily designed to provide a walkthrough of the Loan Prediction analysis, we may gain a full understanding of how to solve a classification problem and create a machine learning model to enhance business performance.

6. Conclusion

The goal of the diploma thesis was to research related literature, scientific papers, and online resources to learn about data mining tasks and machine learning algorithms, as well as various Python libraries for understanding and best use cases of ML and DM applicants, as well as determine loan prediction analysis using historical data.

This diploma thesis represents the iterative cycle of the CRISM-DM, DM and ML project, which includes various stages such as accessing datasets, data cleaning, different statistical analysis and visualization, outlier detection and removal, machine learning methods and algorithms, and finally model creation. In the second chapter of this thesis, I explored the numerous studies of data mining standard process, data mining - environment, architectures, and in detail techniques used to get desired outcomes from this process.

The three and four chapters of the research present a literature review to understand theoretical concepts of topics such as machine learning techniques and algorithms, business use cases, the Python eco - systems and its libraries, and literature.

In the fifth chapter of this research, I completed a practical part based on approaches to achieve the thesis targets. whilst solving a single business problem through customer segmentation data mining with the development of a machine learning model for the analysis of business data and making important business decisions.

To summarize, every day a massive amount of data is recorded in a database in any enterprise. We can identify important factors for business progress based on past data. Utilizing data

techniques and machine learning algorithms, it is possible to discover hidden patterns and knowledge through data analysis, making it simple to make business decisions and advance technical advancements for improved corporate performance. I believe that every sector or corporation should approach new data-driven technology. Whether a company is a new startup or a large corporation, whether it is a product-based company or a service-based company, it should utilize data properly to make appropriate decisions that impact on financially.