ST FRANCIS COLLEGE



**Internship Report on**

**“CUSTOMER CHURN PREDICTION”**

## Submitted in partial fulfillment of the requirements for the award of a Degree of

**Bachelor of Computer Applications**

Submitted by:

**KEERTHANA S (U18IW22S0036)**

**MACHINE LEARNING INTERNSHIP AT**

**Prinston Smart Engineers**

**Under the Guidance of**

**Mr. Akash**

**Project Lead Prinston Smart Engineers**

**Department of Computer Science and Applications**



**ST FRANCIS COLLEGE**



**Department of Computer Science & Applications**

**CERTIFICATE**

Certified that the Project work entitled **“CUSTOMER CHURN PREDICTION”** carried out by **KEERTHANA S [U18IW22S0036]** is a bona-fide student of St Francis College, Bangalore in partial fulfillment of the requirement of VI semester (Machine Learning Project) during the year 2024 – 2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Internship Project report has been approved as it satisfies the academic requirements with respect of the Mini Project work prescribed for the said degree.

## MS SABITHA Dr. R.N. SUBHA RAO

HOD, Dept. of Computer Science Principal

External Viva

Name of the examiners: Signature with date

1.



**ACKNOWLEDGEMENT**

While presenting this Machine Learning Project on **“CUSTOMER CHURN PREDICTION”**, I feel that it is my duty to acknowledge the help rendered to us by various people.

I would like to thank our Principal **Dr. R.N Subha Rao**, ST FRANSCIS COLLEGE for his support though out this project.

I express my whole hearted gratitude to **Ms Sabitha**, who is our respectable Head of Dept. of Computer Science. I wish to acknowledge for his valuable help and encouragement.

I Sincerely Acknowledge Guidance and Constant Encouragement of my Internship Guide **Mr. Akash, Project Lead, Prinston Smart Engineers**, for his guidance and valuable advice at every stage of my project which helped us in the successful completion of project.

# KEERTHANA S [U18IW22S0036]



This project focuses on predicting customer churn in the telecommunications sector using machine learning techniques. The dataset is pre-processed by handling missing values, converting categorical variables, and engineering new features such as tenure bins. Imbalanced data is addressed using the Synthetic Minority Over-sampling Technique (SMOTE) to ensure equal representation in the target variable. The dataset is then split into training and testing sets, and various classification models, including Decision Trees and Random Forests, are trained. Model performance is evaluated using accuracy, confusion matrices, and classification reports to assess predictive capabilities. Feature importance analysis is conducted to identify key factors influencing churn, aiding business decision-making. The final model is saved using pickle for future use in deployment and real-time prediction. Overall, this study provides a comprehensive approach to churn prediction, helping businesses implement proactive retention strategies.

1. Introduction 1-21



* 1. [Problem Statement 21](#_TOC_250009)
  2. [Objective 21](#_TOC_250008)
  3. [Future Scope 21](#_TOC_250007)

1. Requirement Specification 22
   1. [Software Requirements 22](#_TOC_250006)
   2. [Hardware Requirements 22](#_TOC_250005)
2. System Definition 23-35
   1. [Project Description 23](#_TOC_250004)
   2. [Libraries Used 30](#_TOC_250003)
   3. Technology Used 31
   4. [Dataset 32](#_TOC_250002)

[3.5 Advantages 34](#_TOC_250001)

[3.6. Disadvantages 34](#_TOC_250000)

1. Implementation (Code) 36-40
2. Snapshots 41-44
3. Declaration 45
4. Conclusion/Future Enhancement 46
5. Reference 47

**CHAPTER - 1**

**INTRODUCTION**

Machine learning (ML) is a discipline of artificial intelligence (AI) that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention. Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data, and they can independently learn, grow, develop, and adapt. Machine learning derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process. ML algorithms use computation methods to learn directly from data instead of relying on any predetermined equation that may serve as a model.

Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming, coined the term “Machine Learning”. He defined machine learning as – a “Field of study that gives computers the capability to learn without being explicitly programmed”. The process starts with feeding good quality data and then training our machines(computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and what kind of task we are trying to automate.

The performance of ML algorithms adaptively improves with an increase in the number of available samples during the ‘learning’ processes. For example, deep learning is a sub-domain of machine learning that trains computers to imitate natural human traits like learning from examples. It offers better performance parameters than conventional ML algorithms.

Machine learning is used in many different applications, from image and speech recognition to natural language processing, recommendation systems, fraud detection, portfolio optimization, automated task, and so on. Machine learning models are also used to power autonomous vehicles, drones, and robots, making them more intelligent and adaptable to changing environments.

## How machine learning algorithms work

The lifecycle of a machine learning project involves a series of steps that include:

1. **Study the Problems:** The first step is to study the problem. This step involves understanding the business problem and defining the objectives of the model.
2. **Data Collection:** When the problem is well-defined, we can collect the relevant data required for the model. The data could come from various sources such as databases, APIs, or web scraping.
3. **Data Preparation:** When our problem-related data is collected. then it is a good idea to check the data properly and make it in the desired format so that it can be used by the model to find the hidden patterns. This can be done in the following steps:
   * Data cleaning
   * Data Transformation
   * Explanatory Data Analysis and Feature Engineering
   * Split the dataset for training and testing.
4. **Model Selection:** The next step is to select the appropriate machine learning algorithm that is suitable for our problem. This step requires knowledge of the strengths and weaknesses of different algorithms. Sometimes we use multiple models and compare their results and select the best model as per our requirements.
5. **Model building and Training:** After selecting the algorithm, we have to build the model.
6. In the case of traditional machine learning building mode is easy it is just a few hyper parameter tunings.
7. In the case of deep learning, we have to define layer-wise architecture along with input and output size, number of nodes in each layer, loss function, gradient descent optimizer, etc.
8. After that model is trained using the pre-processed dataset.
9. **Model Evaluation:** Once the model is trained, it can be evaluated on the test dataset to determine its accuracy and performance using different techniques like classification report, F1 score, precision, recall, ROC Curve, Mean Square error, absolute error, etc.
10. **Model Tuning:** Based on the evaluation results, the model may need to be tuned or optimized to improve its performance. This involves tweaking the hyperparameters of the model.
11. **Deployment:** Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing software system or creating a new system for the model.
12. **Monitoring and Maintenance:** Finally, it is essential to monitor the model’s performance in the production environment and perform maintenance tasks as required. This involves

monitoring for data drift, retraining the model as needed, and updating the model as new data becomes available.

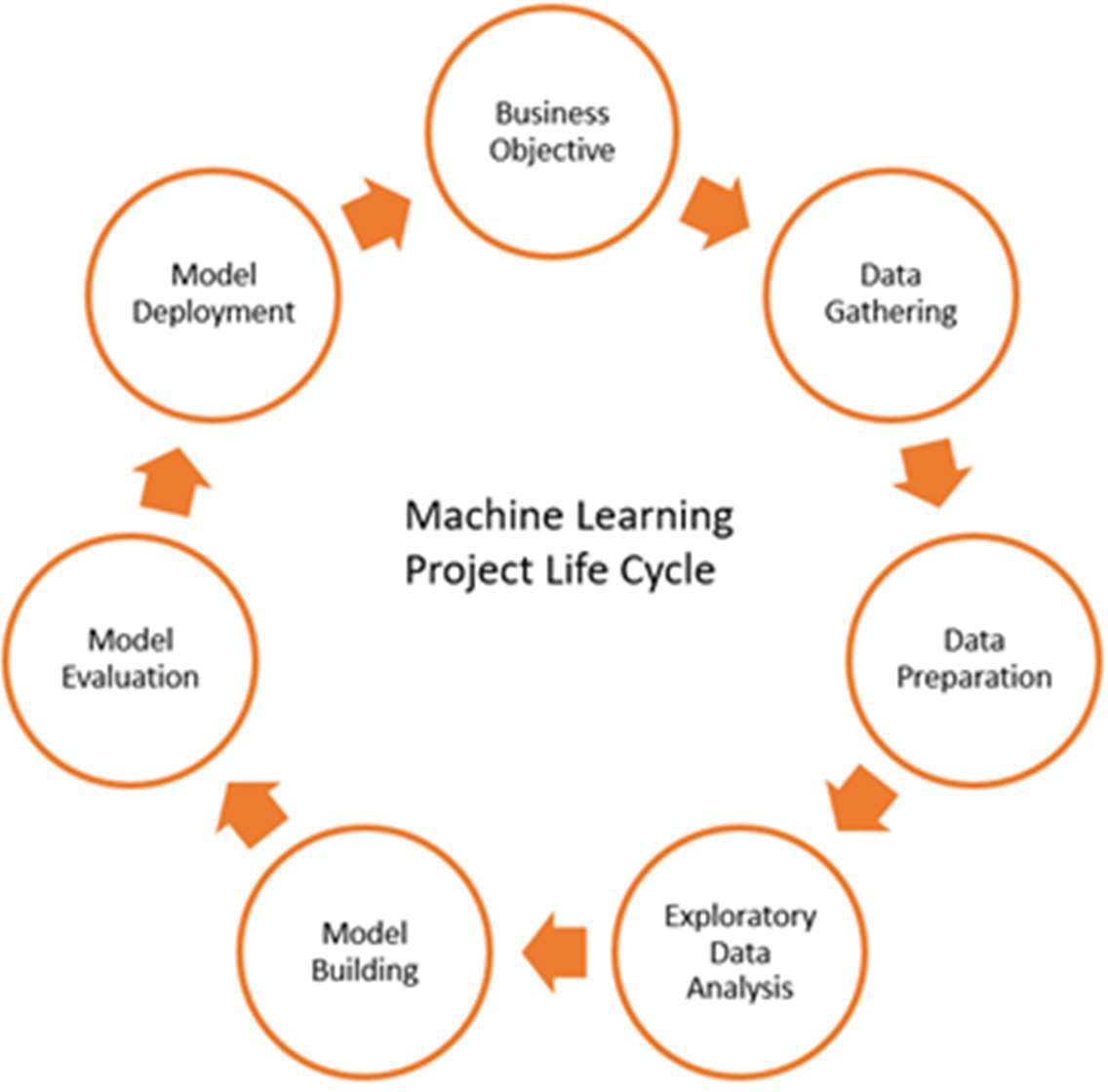


Figure 1.1 Machine Learning Life Cycle

## How we split data in Machine Learning?

Whenever a machine learning model is trained, we can’t train that model on a single dataset or even we train it on a single dataset then we will not be able to assess the performance of our model. For that reason, we split our source data into training, testing, and validation datasets. The data splitting procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Splitting the dataset is essential for an unbiased evaluation of prediction performance.

* + **Training Data:** The part of data we use to train our model. This is the data that your model actually sees (both input and output) and learns from.
  + **Validation Data:** The part of data that is used to do a frequent evaluation of the model, fit on the training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning). This data plays its part when the model is actually training.
  + **Testing Data:** Once our model is completely trained, testing data provides an unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values (without seeing actual output). After prediction, we evaluate our model by comparing it with

the actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.

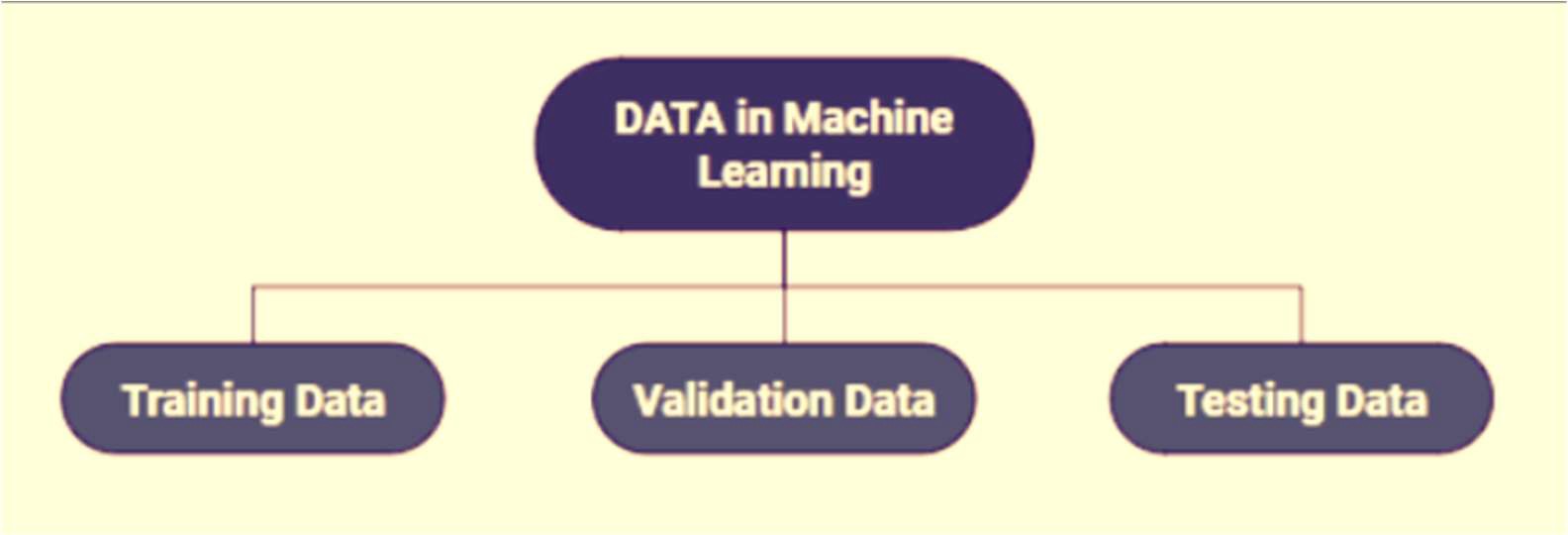


Figure 1.2 Data in Machine Learning

## Types of Machine Learning

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

## Supervised Machine Learning

Supervised learning is a type of machine learning in which the algorithm is trained on the labeled dataset. It learns to map input features to targets based on labeled training data. In supervised learning, the algorithm is provided with input features and corresponding output labels, and it learns to generalize from this data to make predictions on new, unseen data. There are two main categories of supervised learning that are mentioned below:

* + - Classification

Classification deals with predicting categorical target variables, which represent discrete classes or labels. For instance, classifying emails as spam or not spam, or predicting whether a patient has a high risk of heart disease. Classification algorithms learn to map the input features to one of the predefined classes.

Here are some classification algorithms:

* + - * Logistic Regression
      * Support Vector Machine
      * Random Forest
      * Decision Tree
      * K-Nearest Neighbours (KNN)
      * Naive Bayes
    - Regression

Regression, on the other hand, deals with predicting continuous target variables, which represent numerical values. For example, predicting the price of a house based on its size, location, and amenities, or forecasting the sales of a product. Regression algorithms learn to map the input features to a continuous numerical value.

Here are some regression algorithms:

* + - * Linear Regression
      * Polynomial Regression
      * Ridge Regression
      * Lasso Regression
      * Decision tree
      * Random Forest

## Advantages of Supervised Machine Learning

* Supervised Learning models can have high accuracy as they are trained on labeled data**.**
* The process of decision-making in supervised learning models is often interpretable.
* It can often be used in pre-trained models which saves time and resources when developing new models from scratch.

## Disadvantages of Supervised Machine Learning

* It has limitations in knowing patterns and may struggle with unseen or unexpected patterns that are not present in the training data.
* It can be time-consuming and costly as it relies on labeled data only.
* It may lead to poor generalizations based on new data.

## Unsupervised Machine Learning

Unsupervised Learning Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data. Unlike supervised learning, unsupervised learning doesn’t involve providing the algorithm with labeled target outputs. The primary goal of Unsupervised learning is often to discover hidden patterns, similarities, or clusters within the data, which can then be used for various purposes, such as data exploration, visualization, dimensionality reduction, and more.

There are two main categories of unsupervised learning that are mentioned below:

* + - Clustering

Clustering is the process of grouping data points into clusters based on their similarity. This technique is useful for identifying patterns and relationships in data without the need for labeled examples.

Here are some clustering algorithms:

* + - * K-Means Clustering algorithm
      * Mean-shift algorithm
      * DBSCAN Algorithm
      * Principal Component Analysis
      * Independent Component Analysis
    - Association

Association rule learning is a technique for discovering relationships between items in a dataset. It identifies rules that indicate the presence of one item implies the presence of another item with a specific probability.

Here are some association rule learning algorithms:

* + - * Apriori Algorithm
      * Eclat
      * FP-growth Algorithm

## Advantages of Unsupervised Machine Learning

* + It helps to discover hidden patterns and various relationships between the data.
  + Used for tasks such as customer segmentation, anomaly detection, and data exploration**.**
  + It does not require labeled data and reduces the effort of data labelling.

## Disadvantages of Unsupervised Machine Learning

* + Without using labels, it may be difficult to predict the quality of the model’s output.
  + Cluster Interpretability may not be clear and may not have meaningful interpretations.
  + It has techniques such as autoencoders and dimensionality reduction that can be used to extract meaningful features from raw data.

## Semi-Supervised Machine Learning

Semi-Supervised learning is a machine learning algorithm that works between the supervised and unsupervised learning so it uses both labeled and unlabeled data. It’s particularly useful when obtaining labeled data is costly, time-consuming, or resource-intensive. This approach is useful when the dataset is expensive and time-consuming. Semi-supervised learning is chosen when labeled data requires skills and relevant resources in order to train or learn from it.

We use these techniques when we are dealing with data that is a little bit labeled and the rest large portion of it is unlabeled. We can use the unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in the case of image data sets where usually all images are not labeled.

There are a number of different semi-supervised learning methods each with its own characteristics. Some of the most common ones include:

* + - **Graph-based semi-supervised learning:** This approach uses a graph to represent the relationships between the data points. The graph is then used to propagate labels from the labeled data points to the unlabeled data points.
    - **Label propagation:** This approach iteratively propagates labels from the labeled data points to the unlabeled data points, based on the similarities between the data points.
    - **Co-training:** This approach trains two different machine learning models on different subsets of the unlabeled data. The two models are then used to label each other’s predictions.
    - **Self-training:** This approach trains a machine learning model on the labeled data and then uses the model to predict labels for the unlabeled data. The model is then retrained on the labeled data and the predicted labels for the unlabeled data.
    - **Generative adversarial networks (GANs):** GANs are a type of deep learning algorithm that can be used to generate synthetic data. GANs can be used to generate unlabeled data for semi-supervised learning by training two neural networks, a generator and a discriminator.

## Advantages of Semi- Supervised Machine Learning

* It leads to better generalization as compared to supervised learning, as it takes both labeled and unlabeled data.
* Can be applied to a wide range of data.

## Disadvantages of Semi- Supervised Machine Learning

* Semi-supervised methods can be more complex to implement compared to other approaches.
* It still requires some labeled data that might not always be available or easy to obtain.
* The unlabeled data can impact the model performance accordingly.

## Reinforcement Learning

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

## Positive reinforcement

* + - * Rewards the agent for taking a desired action.
      * Encourages the agent to repeat the behaviour.
      * Examples: Giving a treat to a dog for sitting, providing a point in a game for a correct answer.

## Negative reinforcement

* + - * Removes an undesirable stimulus to encourage a desired behaviour.
      * Discourages the agent from repeating the behaviour.
      * Examples: Turning off a loud buzzer when a lever is pressed, avoiding a penalty by completing a task.

## Advantages of Reinforcement Machine Learning

* It has autonomous decision-making that is well-suited for tasks and that can learn to make a sequence of decisions, like robotics and game-playing.
* This technique is preferred to achieve long-term results that are very difficult to achieve.
* It is used to solve a complex problem that cannot be solved by conventional techniques.

## Disadvantages of Reinforcement Machine Learning

* Training Reinforcement Learning agents can be computationally expensive and time-consuming.
* Reinforcement learning is not preferable to solving simple problems.
* It needs a lot of data and a lot of computation, which makes it impractical and costly.

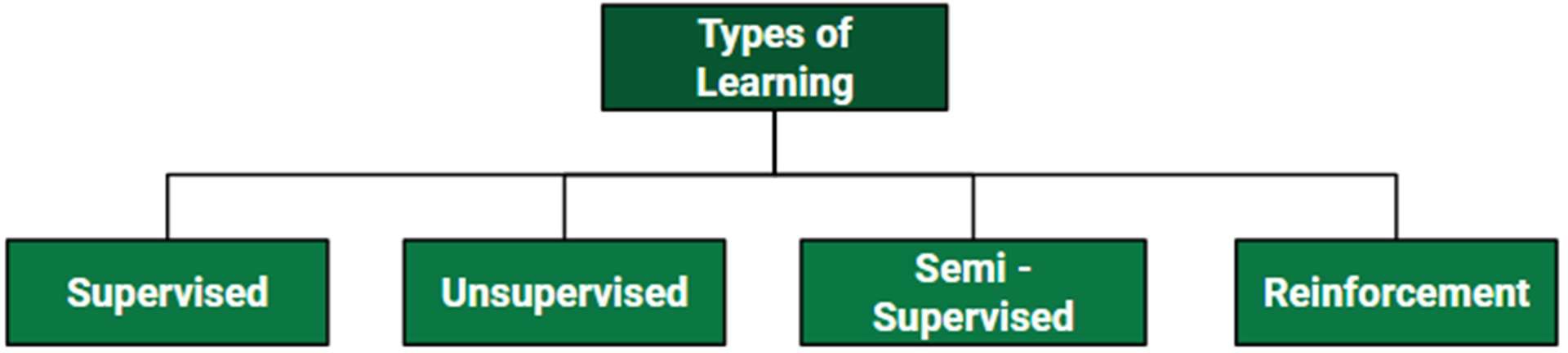


Figure 1.3 Types of Machine Learning

## Need for machine learning:

Machine learning is important because it allows computers to learn from data and improve their performance on specific tasks without being explicitly programmed. This ability to learn from data and adapt to new situations makes machine learning particularly useful for tasks that involve large amounts of data, complex decision-making, and dynamic environments.

Here are some specific areas where machine learning is being used:

* **Predictive modelling:** Machine learning can be used to build predictive models that can help businesses make better decisions. For example, machine learning can be used to predict which customers are most likely to buy a particular product, or which patients are most likely to develop a certain disease.
* **Natural language processing:** Machine learning is used to build systems that can understand and interpret human language. This is important for applications such as voice recognition, chatbots, and language translation.
* **Computer vision:** Machine learning is used to build systems that can recognize and interpret images and videos. This is important for applications such as self-driving cars, surveillance systems, and medical imaging.
* **Fraud detection:** Machine learning can be used to detect fraudulent behaviour in financial transactions, online advertising, and other areas.
* **Recommendation systems:** Machine learning can be used to build recommendation systems that suggest products, services, or content to users based on their past behaviour and preferences.

Overall, machine learning has become an essential tool for many businesses and industries, as it enables them to make better use of data, improve their decision-making processes, and deliver more personalized experiences to their customers.

## Comparison of Machine Learning Algorithms

Comparing machine learning algorithms is important in itself, but there are some not so-obvious benefits of comparing various experiments effectively.

## Better performance

The primary objective of model comparison and selection is definitely better performance of the machine learning software/solution. The objective is to narrow down on the best algorithms that suit both the data and the business requirements.

## Longer lifetime

High performance can be short-lived if the chosen model is tightly coupled with the training data and fails to interpret unseen data. So, it’s also important to find the model that understands underlying data patterns so that the predictions are long-lasting and the need for re-training is minimal.

## Easier retraining

When models are evaluated and prepared for comparisons, minute details, and metadata get recorded which come in handy during retraining. For example, if a developer can clearly retrace the reasons behind choosing a model, the causes of model failure will immediately pop out and re-training can start with equal speed.

## Speedy production

With the model details available at hand, it’s easy to narrow down on models that can offer high processing speed and use memory resources optimally. Also, during production,

several parameters are required to configure the machine learning solutions. Having production-level data can be useful for easily aligning with the production engineers. Moreover, knowing the resource demands of different algorithms, it will also be easier to check their compliance and feasibility with respect to the organization’s allocated assets.

## Performance Metrics in Machine Learning

Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model. To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics. These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyper-parameters. Each ML model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset. In machine learning, each task or problem is divided into classification and Regression. Not all metrics can be used for all types of problems; hence, it is important to know and understand which metrics should be used. Different evaluation metrics are used for both Regression and Classification tasks.

## Performance metrics for Classification

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data into classes or groups based on the training. It predicts class labels as the output, such as *Yes or No, 0 or 1, Spam or Not Spam,* etc. To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

## Accuracy

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

It can be formulated as:



## Confusion Matrix

A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.



Figure 1.4. Confusion Matrix We can determine the following from the above matrix:

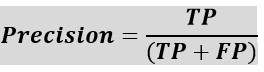
* + - * + In the matrix, columns are for the prediction values, and rows specify the Actual values. Here Actual and prediction give two possible classes, Yes or No. So, if we are predicting the presence of a disease in a patient, the Prediction column with Yes means, Patient has the disease, and for NO, the Patient doesn't have the disease.
        + In this example, the total number of predictions are 165, out of which 110 time predicted yes, whereas 55 times predicted No.
        + However, in reality, 60 cases in which patients don't have the disease, whereas 105 cases in which patients have the disease.

In general, the table is divided into four terminologies, which are as follows:

* + - * + True Positive (TP): In this case, the prediction outcome is true, and it is true in reality, also.
        + True Negative (TN): in this case, the prediction outcome is false, and it is false in reality, also.
        + False Positive (FP): In this case, prediction outcomes are true, but they are false in actuality.
        + False Negative (FN): In this case, predictions are false, and they are true in actuality.

## Precision

The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive or predictions that are actually true to the total positive predictions (True Positive and False Positive).



## Recall

It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

The formula for calculating Recall is given below:



## F-Score

F-score or F1 Score is a metric to evaluate a binary classification model on the basis of predictions that are made for the positive class. It is calculated with the help of Precision

and Recall. It is a type of single score that represents both Precision and Recall. So, the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.

The formula for calculating the F1 score is given below:



## AUC (Area Under the Curve)-ROC

Sometimes we need to visualize the performance of the classification model on charts; then, we can use the AUC-ROC curve. It is one of the popular and important metrics for evaluating the performance of the classification model.

Firstly, let's understand ROC (Receiver Operating Characteristic curve) curve. ROC represents a graph to show the performance of a classification model at different threshold levels. The curve is plotted between two parameters, which are:

* + - * + True Positive Rate
        + False Positive Rate

TPR or true Positive rate is a synonym for Recall, hence can be calculated as:



FPR or False Positive Rate can be calculated as:



## Performance metrics for Regression

Regression is a supervised learning technique that aims to find the relationships between the dependent and independent variables. A predictive regression model predicts a numeric or

discrete value. The metrics used for regression are different from the classification metrics. It means we cannot use the Accuracy metric (explained above) to evaluate a regression model; instead, the performance of a Regression model is reported as errors in the prediction. Following are the popular metrics that are used to evaluate the performance of Regression models.

## Mean Absolute Error

Mean Absolute Error or MAE is one of the simplest metrics, which measures the absolute difference between actual and predicted values, where absolute means taking a number as Positive.

To understand MAE, let's take an example of Linear Regression, where the model draws a best fit line between dependent and independent variables. To measure the MAE or error in prediction, we need to calculate the difference between actual values and predicted values. But in order to find the absolute error for the complete dataset, we need to find the mean absolute of the complete dataset.

The below formula is used to calculate MAE:



## Mean Squared Error

Mean Squared error or MSE is one of the most suitable metrics for Regression evaluation. It measures the average of the Squared difference between predicted values and the actual value given by the model. Since in MSE, errors are squared, therefore it only assumes non- negative values, and it is usually positive and non-zero. Moreover, due to squared differences, it penalizes small errors also, and hence it leads to over-estimation of how bad the model is. MSE is a much-preferred metric compared to other regression metrics as it is differentiable and hence optimized better.

The formula for calculating MSE is given below:



## R2 Score

R squared error is also known as Coefficient of Determination, which is another popular metric used for Regression model evaluation. The R-squared metric enables us to compare our model with a constant baseline to determine the performance of the model. To select the constant baseline, we need to take the mean of the data and draw the line at the mean.

The R squared score will always be less than or equal to 1 without concerning if the values are too large or small.



## Adjusted R2

Adjusted R squared, as the name suggests, is the improved version of R squared error. R square has a limitation of improvement of a score on increasing the terms, even though the model is not improving, and it may mislead the data scientists.

To overcome the issue of R square, adjusted R squared is used, which will always show a lower value than R². It is because it adjusts the values of increasing predictors and only shows improvement if there is a real improvement.

We can calculate the adjusted R squared as follows:



## Data Pre-Processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data preprocessing is a technique that is used to convert the raw data into a clean

data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

## Why do We Need Data Preprocessing?

* + Improving Data Quality: Data preprocessing is essential for enhancing the quality of data by handling inconsistencies, inaccuracies, and errors, which is critical for ensuring reliable and robust analytics.
  + Dealing with Missing Values: Data preprocessing includes techniques like imputation that are critical for dealing with missing data effectively, as datasets often have missing values which can significantly hinder the performance of machine learning models.
  + Normalizing and Scaling: Data preprocessing helps in normalizing or scaling features, which is especially important for algorithms that are sensitive to the scale of the input. This ensures that all the features are on a comparable scale, which is crucial for the accurate performance of many machine learning algorithms.
  + Handling Outliers: Through data preprocessing, outliers can be identified and managed appropriately. This is important as outliers can have a disproportionate effect on the modelling process and can lead to misleading results.
  + Dimensionality Reduction: Data preprocessing includes techniques such as Principal Component Analysis (PCA) for reducing the number of input features, which not only helps in improving the performance of models but also makes the dataset more manageable and computationally efficient.

## Steps in Data Preprocessing

Data preprocessing is a step that involves transforming raw data so that issues owing to the incompleteness, inconsistency, and/or lack of appropriate representation of trends are resolved so as to arrive at a dataset that is in an understandable format. The steps used in data preprocessing include the following:

1. Data profiling. Data profiling is the process of examining, analyzing and reviewing data to collect statistics about its quality. It starts with a survey of existing data and its characteristics. Data scientists identify data sets that are pertinent to the problem at hand, inventory its significant attributes, and form a hypothesis of features that might be relevant for the proposed analytics or machine learning task. They also relate data sources to the relevant business concepts and consider which preprocessing libraries could be used.
2. Data cleansing. The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.
3. Data reduction. Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.
4. Data transformation. Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring unstructured data, combining salient variables when it makes sense or identifying important ranges to focus on.
5. Data enrichment. In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organized to achieve the optimal balance between the training time for a new model and the required compute.
6. Data validation. At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model. This second step helps identify any problems in the hypothesis used in the cleaning and feature engineering of the data. If the data scientists are satisfied with the results, they can push the preprocessing task to a data engineer who figures out how to scale it for production. If not, the data scientists can go back and make changes to the way they implemented the data cleansing and feature engineering steps.

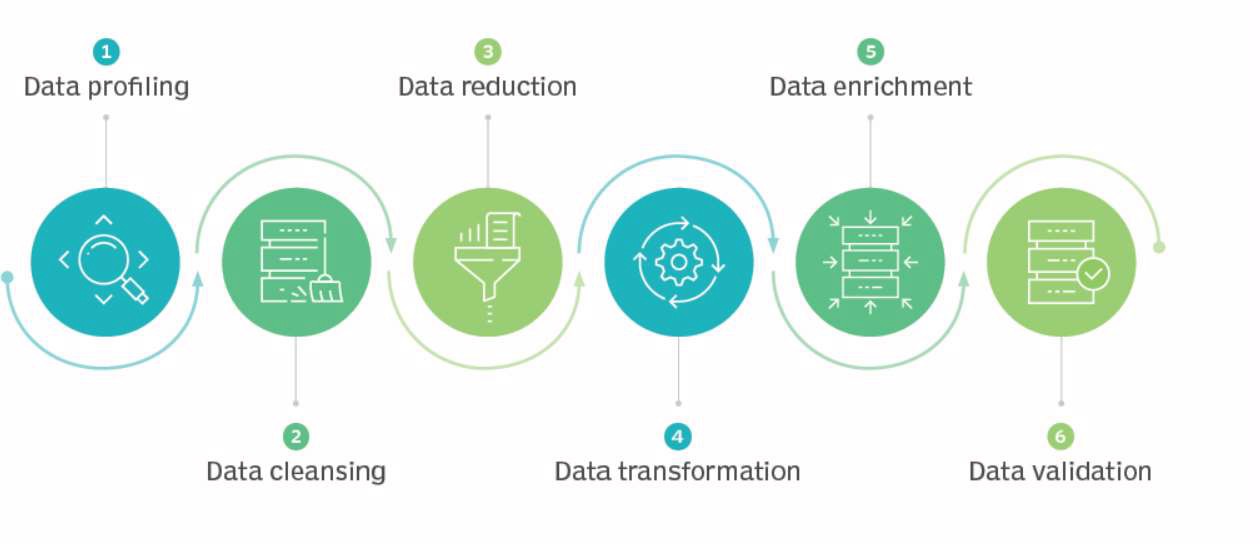


Figure 1.5. Steps in Data Pre-Processing

## Feature Encoding

Feature encoding is the process of transforming data into a format that can be used by machine learning algorithms. This is often necessary when working with real-world data, which can be messy and unstructured.

Machine learning models can only work with numerical values. For this reason, it is necessary to transform the categorical values of the relevant features into numerical ones. This process is called feature encoding.

Here are some of the more well-known and widely used encoding techniques:

* + **Label encoding:** Label encoding is a method of encoding variables or features in a dataset. It involves converting categorical variables into numerical variables.

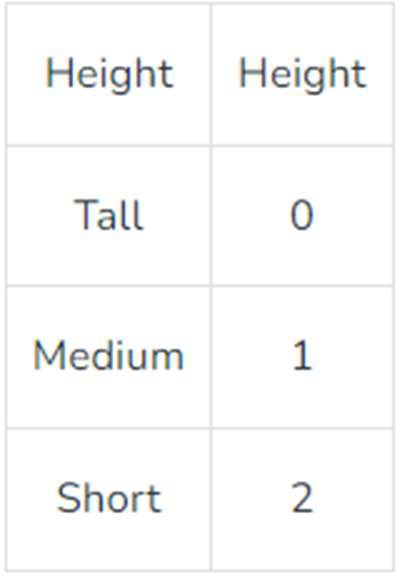
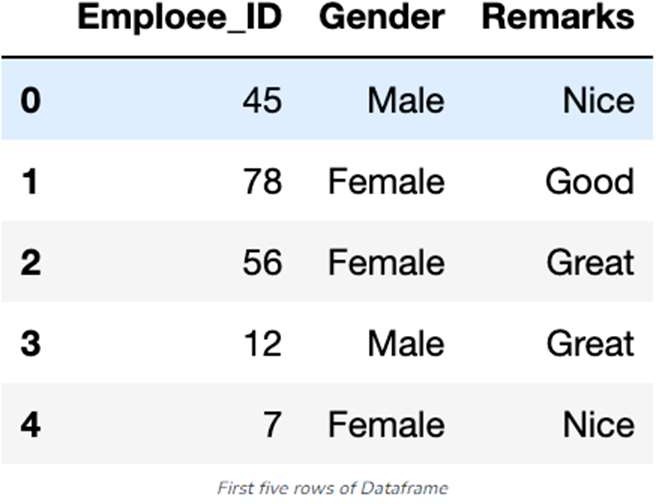
Suppose we have a column *Height* in some dataset that has elements as Tall, Medium, and short. To convert this categorical column into a numerical column we will apply label encoding to this column. After applying label encoding, the Height column is converted into a numerical column having elements 0,1, and 2 where 0 is the label for tall, 1 is the label for medium, and 2 is the label for short height.

Figure 1.6. Label Encoding

* + **One-hot encoding:** One-hot encoding is the process by which categorical variables are converted into a form that can be used by ML algorithms.



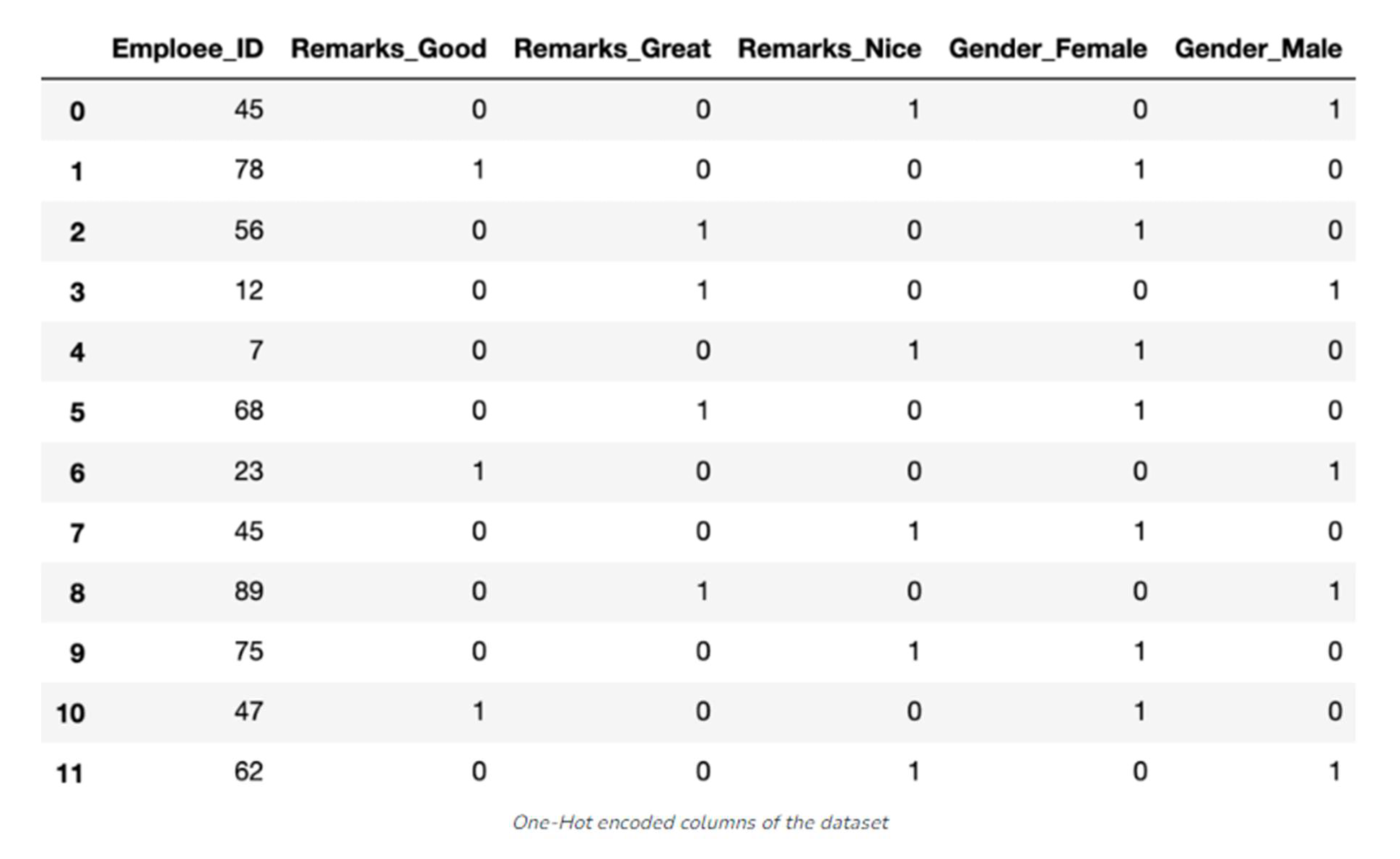


Figure 1.7. One-Hot Encoding

* + **Binary encoding:** Binary encoding is the process of encoding data using the binary code. In binary encoding, each character is represented by a combination of 0s and 1s.

For Example:

o 0000 - 0

o 0001 - 1

o 0010 - 2

o 0011 - 3

o 0100 - 4

o 0101 - 5

o 0110 - 6

o 0111 - 7

o 1000 - 8

o 1001 - 9

o 1010 – 10

# PROBLEM STATEMENT

To show the Implementation, Classification and Regression Analysis on the Customer Churn Prediction Dataset by Kaggle.

# OBJECTIVE

* **Classification**: In the classification task, we aim to predict whether a customer will churn or stay based on various demographic, behavioural, and service-related factors. This involves assigning customers into categories such as "Churn" and "Not Churn", helping businesses identify at-risk customers. Features like contract type, tenure, payment method, and monthly charges play a crucial role in determining customer churn behaviour.
* Regression: In the regression task, our goal is to predict the probability of churn or the time until a customer churns using factors such as monthly charges, tenure, total charges, and service subscriptions. Instead of simply labeling a customer as likely to churn, regression helps estimate how soon a customer is expected to leave or their churn risk score, allowing businesses to implement personalized retention strategies based on urgency.

# FUTURE SCOPE

During the course of this project, I recognized the **critical importance of feature scaling** in improving the performance of machine learning models. Since different features in the dataset, such as **tenure, monthly charges, and total charges**, have varying numerical ranges, ensuring they are on a consistent scale is essential for optimal model performance. Feature scaling plays a particularly crucial role in algorithms that rely on distance calculations, such as **Support Vector Machines (SVM) and K-Nearest Neighbours (KNN)**. Without proper scaling, models may give disproportionate importance to features with larger numerical values, leading to biased predictions.

As a future direction, we can further enhance the **churn prediction system** by exploring **advanced feature scaling techniques**, such as **log transformation, power transformation, or adaptive normalization methods**. Additionally, testing the impact of different scaling methods, such as **Min-Max Scaling, Standardization (Z-score), or Robust Scaling**, on model accuracy and robustness could yield valuable insights. By refining these preprocessing steps, we can improve the **efficiency, interpretability, and generalization** of our machine learning models, leading to more reliable churn predictions and better business decision-making.

**CHAPTER - 2**

**REQUIREMENTS SPECIFICATION**

# SOFTWARE REQUIREMENTS

* + - Operating system – Windows 7/8/10/11
    - Google Collab Environment
    - Libraries – NumPy, Scikit-Learn, Matplotlib and Pandas
    - Language used is Python

# HARDWARE REQUIREMENTS

* Processor: Intel Core i3 or higher
* Processor Speed: 1 GHz or above
* Memory (RAM): Minimum 2 GB
* Storage: 1TB Hard Disk Drive
* Input Devices: Standard Keyboard and Mouse or any other pointing device
* Display: Color Monitor for visualization and analysis

**CHAPTER - 3**

**SYSTEM DEFINITION**

# PROJECT DESCRIPTION

Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

## Classification

Classification is a technique for determining which class the dependent belongs to, based on one or more independent variables.

A classifier is a type of machine learning algorithm that assigns a label to a data input. Classifier algorithms use labeled data and statistical methods to produce predictions about data input classifications. Here, we employ logistic regression as the primary classification algorithm.

## Logistic Regression

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class. It is used for classification algorithms its name is logistic regression. it’s referred to as regression because it takes the output of the linear regression function as input and uses a sigmoid function to estimate the probability for the given class.

Firstly, linear regression is performed on the relationship between variables to get the model.



The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function.



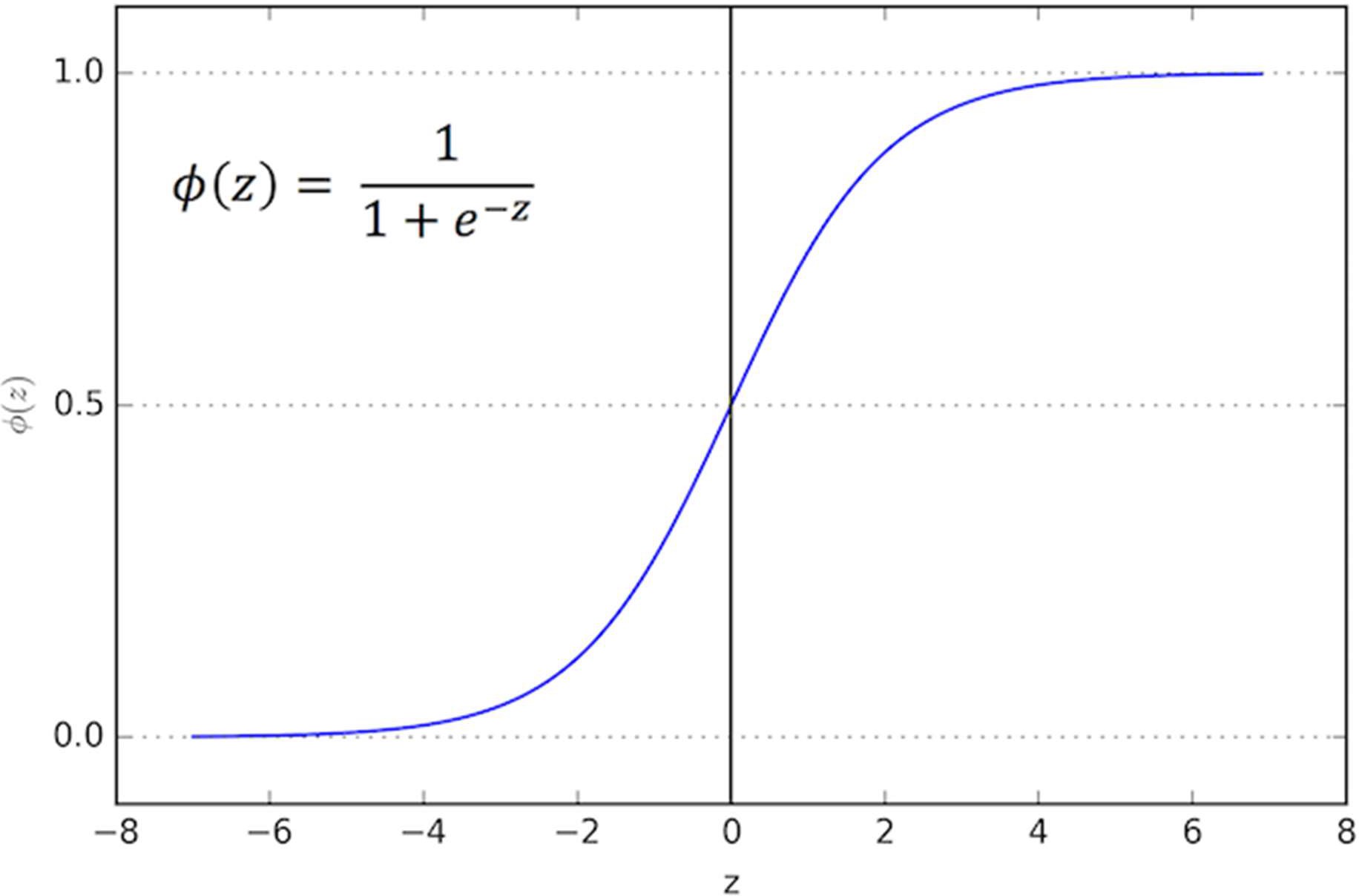


Figure 3.1. Logistic Regression

## Regression

Regression is a technique in machine learning that predicts outcomes by finding relationships between dependent and independent variables. It's a supervised learning algorithm that uses labeled training data to create models. A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”.

Linear regression, decision tree regression, and random forest regression are the chosen algorithms in this project.

## Linear regression

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

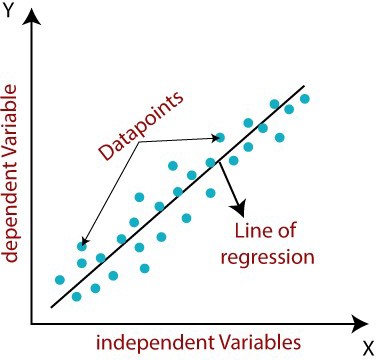


Figure 3.2. Linear Regression Mathematically, we can represent a linear regression as:

y= a0+a1x+ ε

Here,

Y= Dependent Variable (Target Variable)

X= Independent Variable (predictor Variable)

a0= intercept of the line (Gives an additional degree of freedom)

a1 = Linear regression coefficient (scale factor to each input value). ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

## Decision Tree Regression

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.

The process of splitting starts at the root node and is followed by a branched tree that finally leads to a leaf node (terminal node) that contains the prediction or the final outcome of the algorithm. Construction of decision trees usually works top-down, by choosing a variable at each step that best splits the set of items. Each sub-tree of the decision tree model can be represented as a binary tree where a decision node splits into two nodes based on the conditions.

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

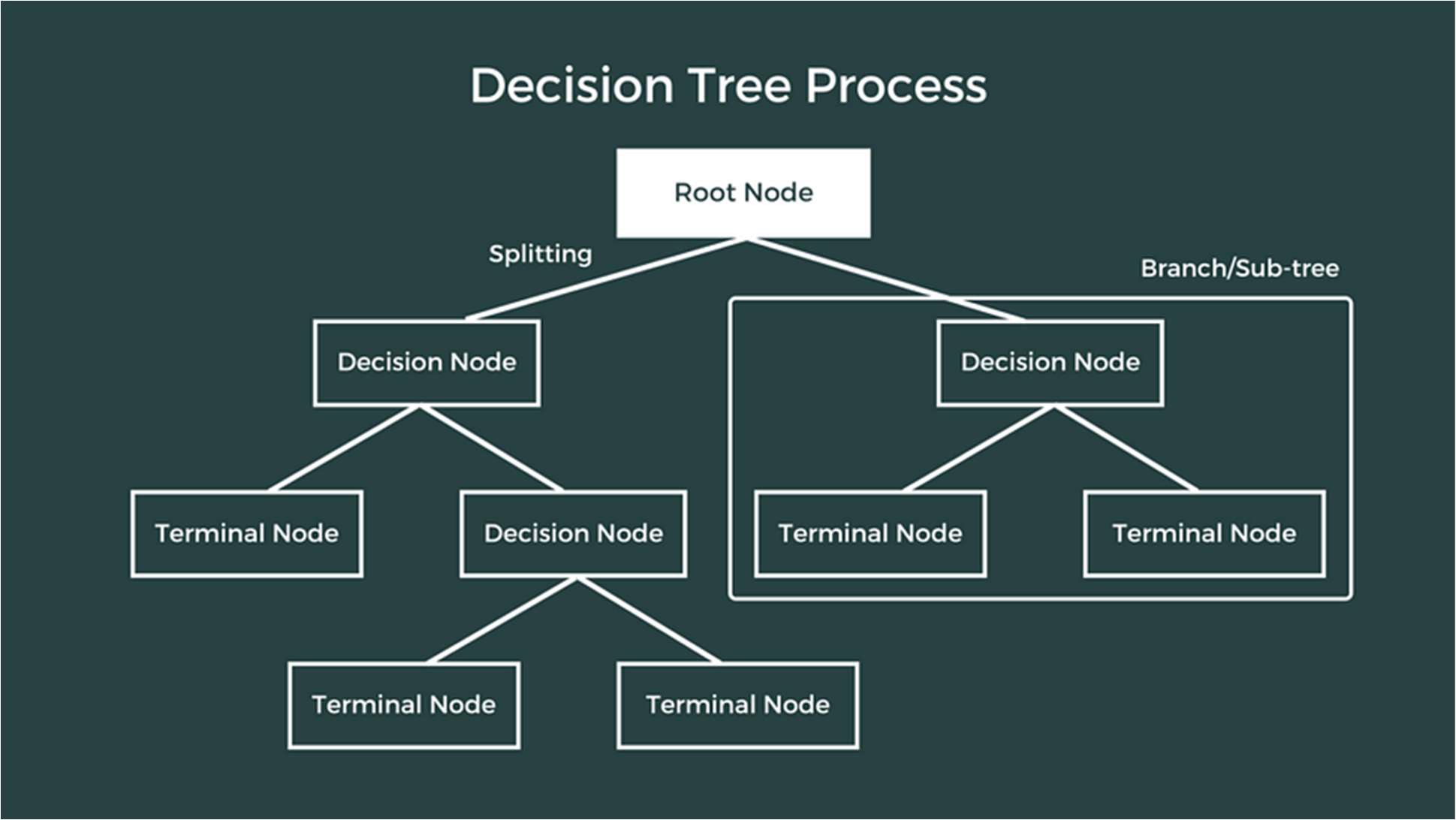


Figure 3.3. Decision Tree Regression

## Random Forest Regression

Random forests or random decision forests are an ensemble learning method that uses multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms mostly for solving classification and regression problems.

Random Forest Regression algorithms are a class of Machine Learning algorithms that use the combination of multiple random decision trees each trained on a subset of data. The use of multiple trees gives stability to the algorithm and reduces variance. The random forest regression algorithm is a commonly used model due to its ability to work well for large and most kinds of data.

The algorithm creates each tree from a different sample of input data. At each node, a different sample of features is selected for splitting and the trees run in parallel without any interaction. The predictions from each of the trees are then averaged to produce a single result which is the prediction of the Random Forest.

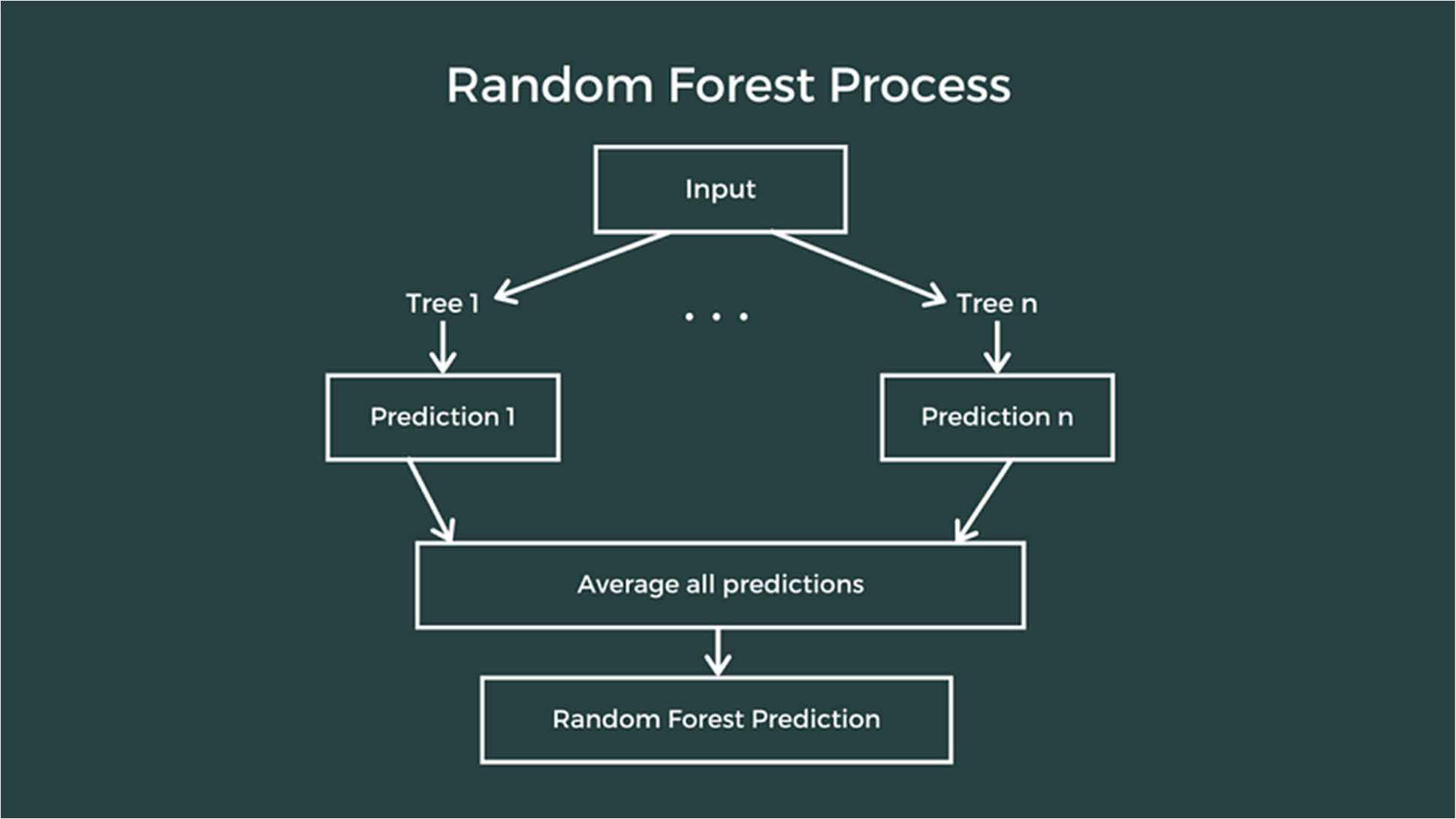


Figure 3.4. Random Forest Regression

# WORKING DESCRIPTION

The project focuses on an in-depth analysis of student performance, utilizing machine learning techniques to explore the complex interactions between demographic variables and academic outcomes. The central dataset, denoted as 'StudentsPerformance.csv,' serves as the foundation for investigating student scores in math, reading, and writing, alongside a myriad of personal, social, and economic factors.

# STUDENTS PERFORMANCE IN EXAMS

Education, being a cornerstone of societal progress, demands continuous refinement and adaptation to meet the evolving needs of learners. In this pursuit, the amalgamation of data science and education emerges as a potent force, capable of unravelling intricate patterns and providing nuanced insights into student performance. The project at hand delves into this intersection,

employing advanced machine learning techniques to analyses a comprehensive dataset, 'StudentsPerformance.csv,' with the overarching goal of understanding and predicting academic outcomes.

In contemporary educational landscapes, the availability of vast datasets presents an unprecedented opportunity to decipher the multifaceted dynamics that influence student success. This project sets out to explore these dynamics, leveraging the wealth of information encapsulated in the 'StudentsPerformance.csv' dataset. By scrutinizing demographic attributes, parental educational backgrounds, and performance across various subjects, it is aimed to uncover hidden correlations and predictive relationships.

The project methodology encompasses data preprocessing, feature engineering, and the application of machine learning models. By employing techniques such as one-hot encoding for categorical variables and standard scaling for numerical features, it is ensured the data is primed for meaningful analysis. Subsequently, classification and regression models are deployed, aided by the scikit-learn library, to derive predictive insights.

The significance of this project extends beyond the realm of academic inquiry. It holds the potential to inform educational policy, guide interventions, and contribute to the ongoing discourse on personalized learning.

## Context of Dataset

The data set consists of the marks secured by the students in various subjects. This data set includes scores from three exams of High school students from the United States and a variety of personal, social, and economic factors that have interaction effects upon them.

## Data Pre-Processing

Data preprocessing is an important step before using it. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific model to train. In this dataset, there are both numerical and categorical features. The categorical features need to be converted to numerical as the models takes only the numerical values.

Key steps in this process include handling missing data, encoding categorical variables, and scaling numerical features.

## Handling Missing Data:

Identify and address missing values in the dataset, employing strategies such as imputation or removal to maintain data integrity.

## Encoding Categorical Variables:

Utilize one-hot encoding to convert categorical variables into a format suitable for machine learning models, enhancing their interpretability and effectiveness.

## Scaling Numerical Features:

Standardize numerical features to a common scale using techniques like Standard Scaler, preventing certain variables from dominating the modelling process.

By implementing these pre-processing steps, it is ensured that the dataset is primed for meaningful analysis and model development.

## Training and Testing split

To gauge the performance and generalizability of our machine learning models, we employ a training and testing split. This involves partitioning the dataset into subsets dedicated to model training and evaluation.

Before splitting the data for training and testing, we have to assign the response variable and predictor variable to Y and X respectively. Now we have to split the data in an 80:20 ratio. 80% of the data will be used for training the models and 20% of the data will be used for testing.

## Performing Classification

The classification aspect of our project involves predicting student grades based on a combination of demographic and educational factors. Logistic regression is employed as the primary classification algorithm.

Through classification modelling, it is aimed to uncover the intricate relationships between predictor variables and academic grades, offering a predictive framework for educators.

## Performing Regression

The regression component of our project focuses on predicting math scores using a suite of features. Linear regression, decision tree regression, and random forest regression are the chosen algorithms.

With the prepared model, test that with the 20% (Test) testing data and assign that to the yapped variable Now test the performance of the model using root squared mean error and r2 score.

# LIBRARIES USED

In this project, several essential Python libraries are harnessed to seamlessly process, analyses, and model complex educational data.

## NumPy

NumPy is a general-purpose array-processing package. NumPy served as the backbone for scientific computing, facilitating numerical operations and array manipulations critical for data preprocessing. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data.

## Pandas

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high- performance & productivity for users. It allows to efficiently handle and explore the 'StudentsPerformance.csv' dataset, organizing the information into a structured and analyzable format.

## Matplotlib

Matplotlib is a plotting library that is ideal for creating visualizations in Python. It provides a range of tools for creating line plots, scatter plots, histograms, and more. Matplotlib is utilized for creating visualizations to better understand the distribution of scores and relationships between different features.

## Scikit-learn

Scikit-learn is a popular library for machine learning in Python. It provides a wide range of tools for classification, regression, and clustering. Scikit-learn also includes tools for data preprocessing, model selection, and model evaluation. Scikit-learn provides a wide array of tools enabling us to predict students' grades and math scores based on various demographic and educational features.

# TECHNOLOGIES USED

## Machine Learning

Machine learning (ML) constitutes the exploration of computer algorithms designed to enhance their performance autonomously through experience and the utilization of data. Positioned within the realm of artificial intelligence, ML algorithms construct models based on sample data, commonly referred to as "training data." This process enables them to make predictions or decisions without explicit programming for each scenario.

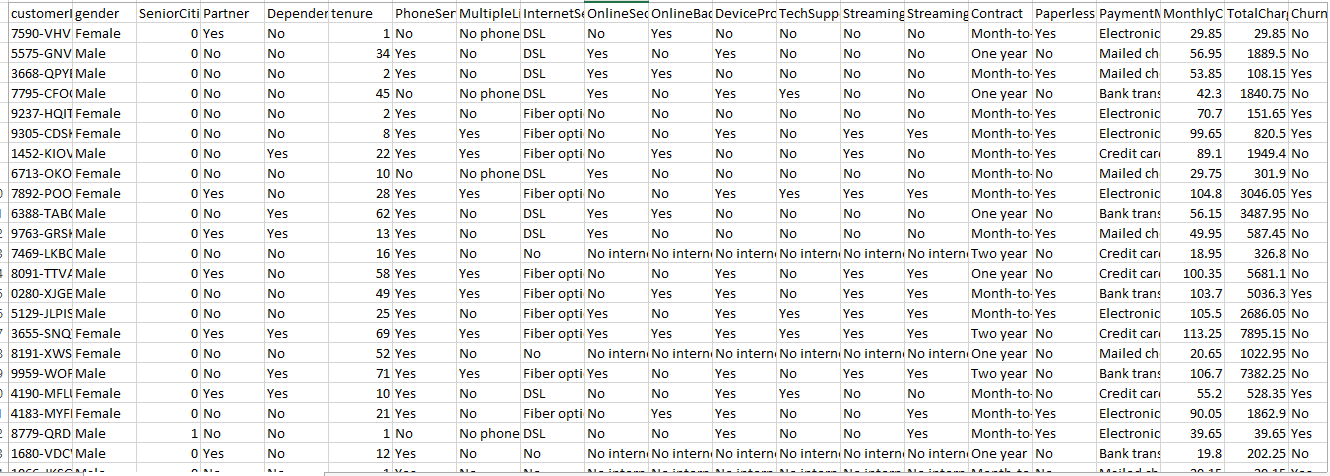
The application of machine learning spans a diverse range of fields, including medicine, email filtering, speech recognition, and computer vision. Its significance lies in its ability to tackle tasks that are challenging or impractical to address using traditional algorithms. In essence, machine learning empowers systems to learn and adapt, offering valuable solutions across various domains.

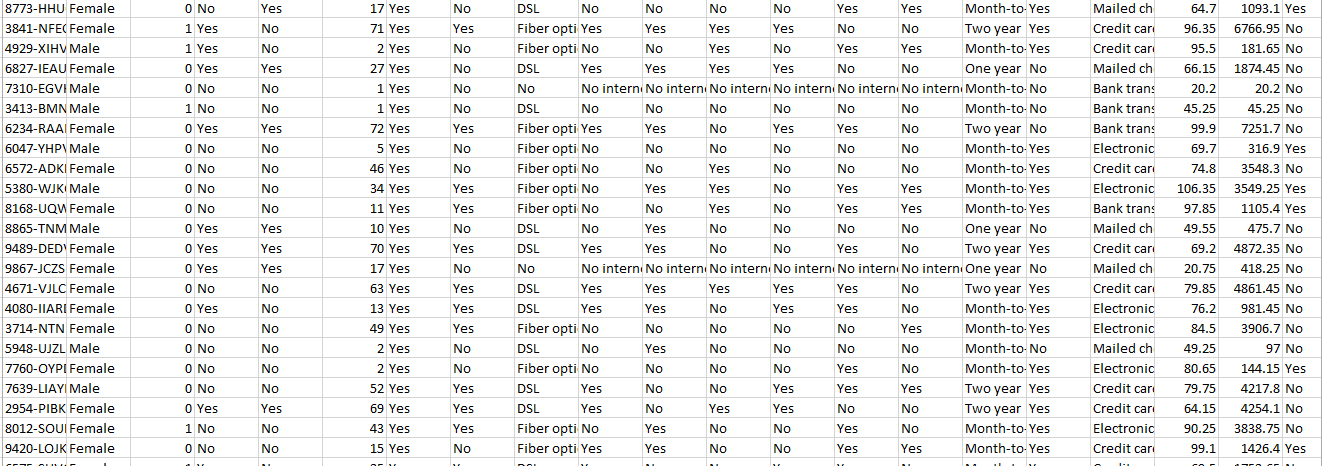
## Python

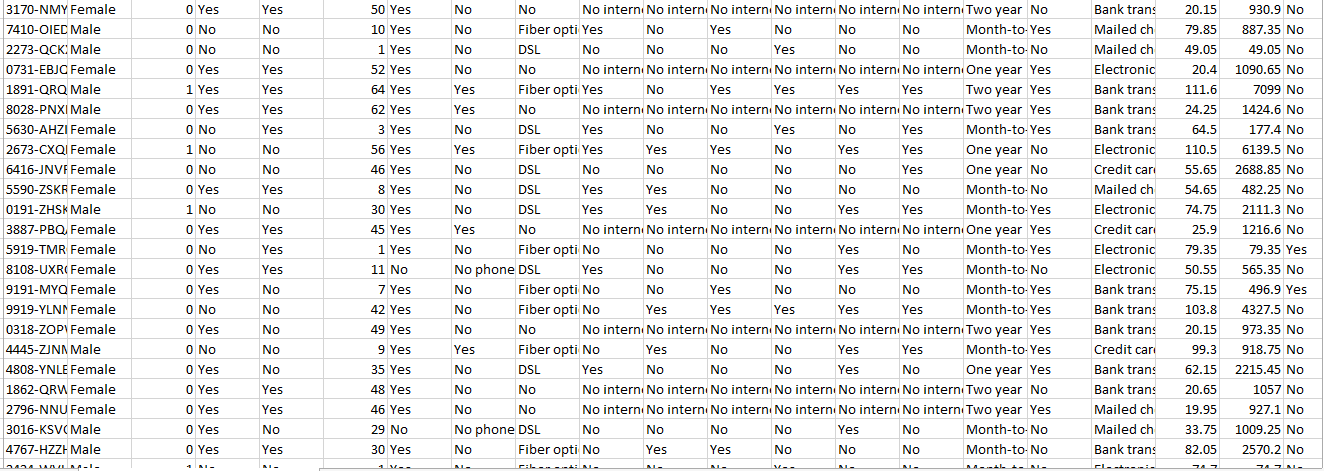
Python stands out as a high-level, versatile, and exceedingly popular programming language. Widely employed in diverse domains, the latest iteration, Python 3, finds applications in web development, machine learning, and various cutting-edge technologies within the software industry. Its adaptability makes it an ideal choice for beginners entering the programming landscape and proves equally advantageous for seasoned programmers with expertise in other languages such as C++ and Java. The language's versatility and broad adoption contribute to its standing as a key player in contemporary software development.

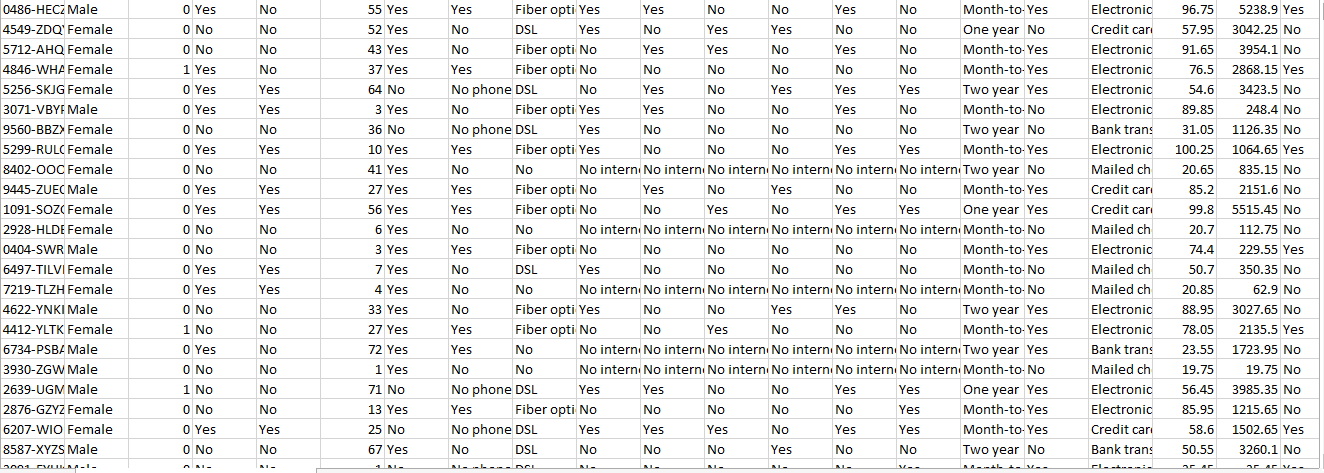
# DATASET

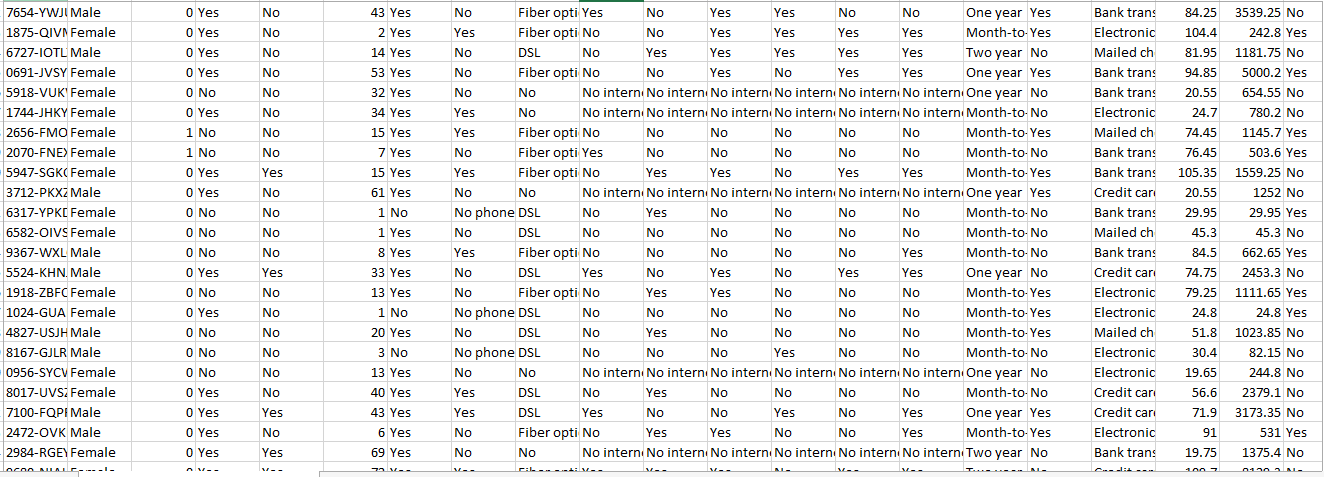
For this project, I have used the dataset extracted from Kaggle. The dataset given by the source is fairly accurate and it is the marks secured by students in high school in the United States. The dataset has 7044 rows and 21 columns. Snapshot of part of the dataset is given below:











# ADVANTAGES

## Data-Driven Insights:

The project leverages machine learning techniques to extract valuable insights from the dataset, providing a data-driven understanding of factors influencing student performance.

## Predictive Modelling:

By employing classification and regression models, the project enables the prediction of student grades and math scores. This can be valuable for identifying students at risk or understanding the impact of various factors on academic outcomes.

## Educational Decision Support:

The project's findings can potentially inform educational decision-making processes. Schools and educational institutions may benefit from insights into the relationships between demographic factors and academic success.

## Applicability:

The use of widely adopted Python libraries such as NumPy, pandas, Matplotlib, and scikit-learn ensures the project's applicability and accessibility to a broad audience in the data science and education communities.

## Interdisciplinary Relevance:

The project bridges the gap between machine learning and education, demonstrating the interdisciplinary relevance of data science in addressing challenges and opportunities within the educational domain.

# DISADVANTAGES

## Bias and Fairness:

The dataset itself may carry biases, and the machine learning models could inadvertently perpetuate or amplify these biases. It's crucial to be aware of potential ethical considerations related to fairness and bias in predictions.

## Limited Scope:

The project focuses on a specific dataset related to student performance in exams. The findings may not be universally applicable to all educational contexts, and the scope is limited to the features present in the dataset.

## Data Quality and Completeness:

The accuracy of the models heavily depends on the quality and completeness of the dataset. Incomplete or inaccurate data may lead to biased model outcomes and limit the generalizability of the findings.

## Assumption of Causation:

While machine learning models can identify correlations, they do not establish causation. Correlations found in the analysis should be interpreted with caution, and further research may be needed to validate causal relationships.

**CHAPTER - 4**

**IMPLEMENTATION (CODE)**

## Import Necessary Libraries

#Basic import import numpy as np import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import mean\_squared\_error, r2\_score from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score #Import csv Dataset

df = pd.read\_csv('/content/StudentsPerformance.csv’)

## Data preprocessing

#Create average column

avg = df.iloc[:, 5:8].mean(axis = 1) df["Average Score"] = round(avg, 1) #Create grade column

# 0-30.9 = Fail #40-49.9 = Pass

#50-69.9 = Second Class Lower

#70-79.9 = Second Class Upper #80-100 = First Class

criteria = (df["Average Score"].between(0.0, 39.9), df["Average Score"].between(40.0, 49.9),

df["Average Score"].between(50.0, 69.9), df["Average Score"].between(70.0, 79.9), df["Average Score"].between(80.0, 100))

values = ["fail", "pass", "second class lower", "second class upper", "first class"] #Add grade column

df['Grade'] = np.select(criteria, values, 0)

## Classification

# Split data into features and target variables

X = df.drop(columns=['Average Score', 'Grade'],axis=1) y = df['Grade']

num\_features = X.select\_dtypes(exclude="object").columns cat\_features = X.select\_dtypes(include="object").columns

from sklearn.preprocessing import OneHotEncoder, StandardScaler from sklearn.compose import ColumnTransformer numeric\_transformer = StandardScaler()

oh\_transformer = OneHotEncoder() preprocessor = ColumnTransformer( [

("OneHotEncoder", oh\_transformer, cat\_features), ("StandardScaler", numeric\_transformer, num\_features),

]

)

X = preprocessor.fit\_transform(X)

# Split data into training and testing sets

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.2) lcla = LogisticRegression(max\_iter=500)

lcla.fit(X\_train, y\_train) pred=lcla.predict(X\_test) # evaluate predictions

accuracy = accuracy\_score(y\_test, pred) print('Accuracy: %.2f' % (accuracy\*100))

## Regression

# Split data into features and target variables

X = df.drop(columns=['math score', 'Average Score', 'Grade'],axis=1) y = df['math score']

num\_features = X.select\_dtypes(exclude="object").columns cat\_features = X.select\_dtypes(include="object").columns

from sklearn.preprocessing import OneHotEncoder, StandardScaler from sklearn.compose import ColumnTransformer numeric\_transformer = StandardScaler()

oh\_transformer = OneHotEncoder() preprocessor = ColumnTransformer( [

("OneHotEncoder", oh\_transformer, cat\_features),

("StandardScaler", numeric\_transformer, num\_features),

]

)

X = preprocessor.fit\_transform(X) # separate dataset into train and test

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.2) def evaluate\_model(true, predicted):

mse = mean\_squared\_error(true, predicted)

rmse = np.sqrt(mean\_squared\_error(true, predicted)) r2\_square = r2\_score(true, predicted)

return rmse, r2\_square models = {

"Linear Regression": LinearRegression(), "Decision Tree": DecisionTreeRegressor(),

"Random Forest Regressor": RandomForestRegressor()

}

for i in range(len(list(models))): # Train model

model = list(models.values())[i] model.fit(X\_train, y\_train)

# Make predictions

y\_test\_pred = model.predict(X\_test) # Evaluate model

model\_test\_rmse, model\_test\_r2 = evaluate\_model(y\_test, y\_test\_pred) print(list(models.keys())[i])

print('Model performance')

print("- Root Mean Squared Error: {:.4f}".format(model\_test\_rmse)) print("- R2 Score: {:.4f}".format(model\_test\_r2))

print('='\*35)

print('\n’) #result

lin\_model = LinearRegression(fit\_intercept=True) lin\_model = lin\_model.fit(X\_train, y\_train) y\_pred = lin\_model.predict(X\_test)

score = r2\_score(y\_test, y\_pred)\*100

print(" Accuracy of the model is %.2f" %score)

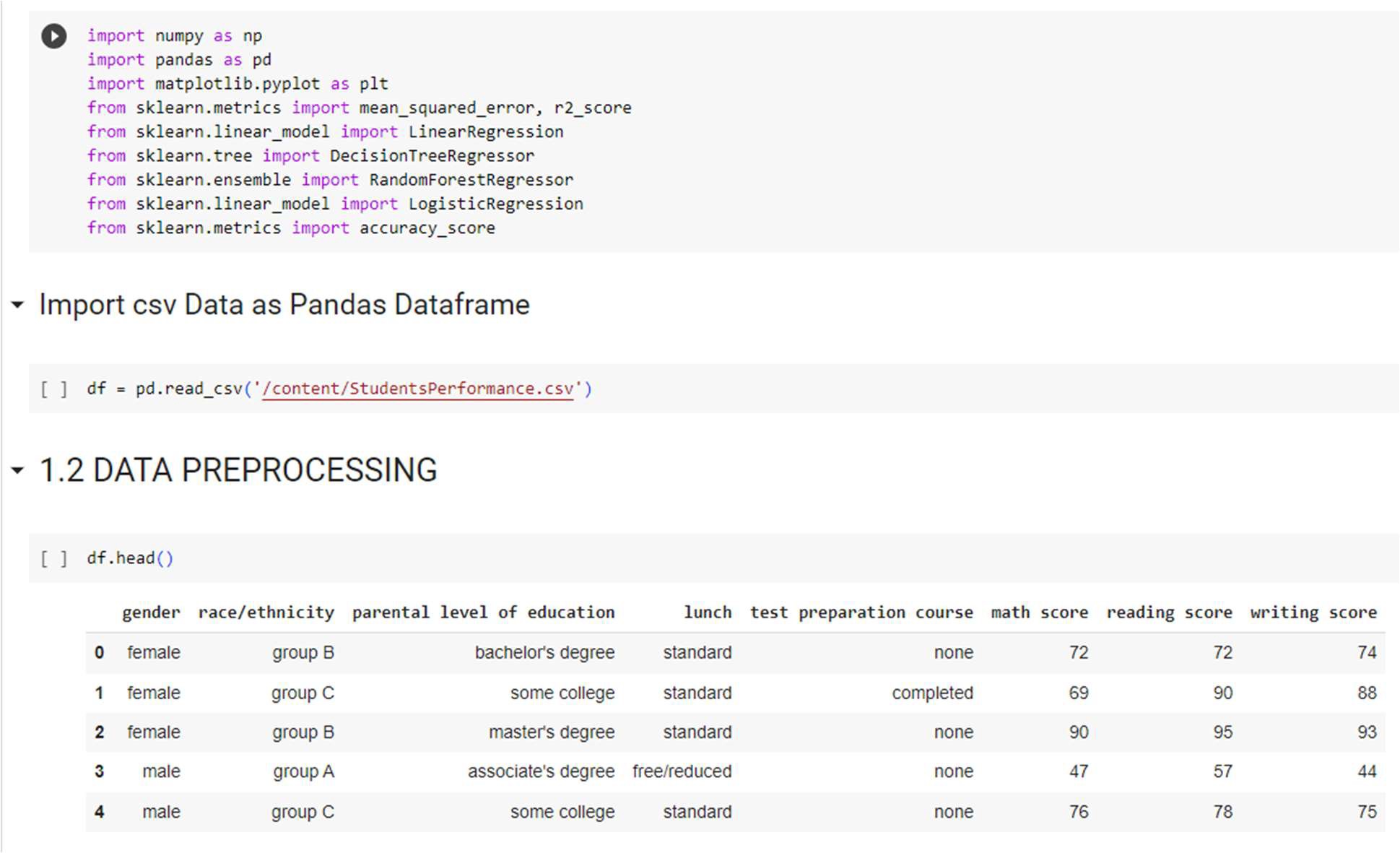
pred\_df=pd.DataFrame({'Actual Value':y\_test, 'Predicted Value':y\_pred, 'Difference':y\_test- y\_pred})

print(pred\_df) plt.scatter(y\_test,y\_pred) plt.xlabel('Actual') plt.ylabel('Predicted')

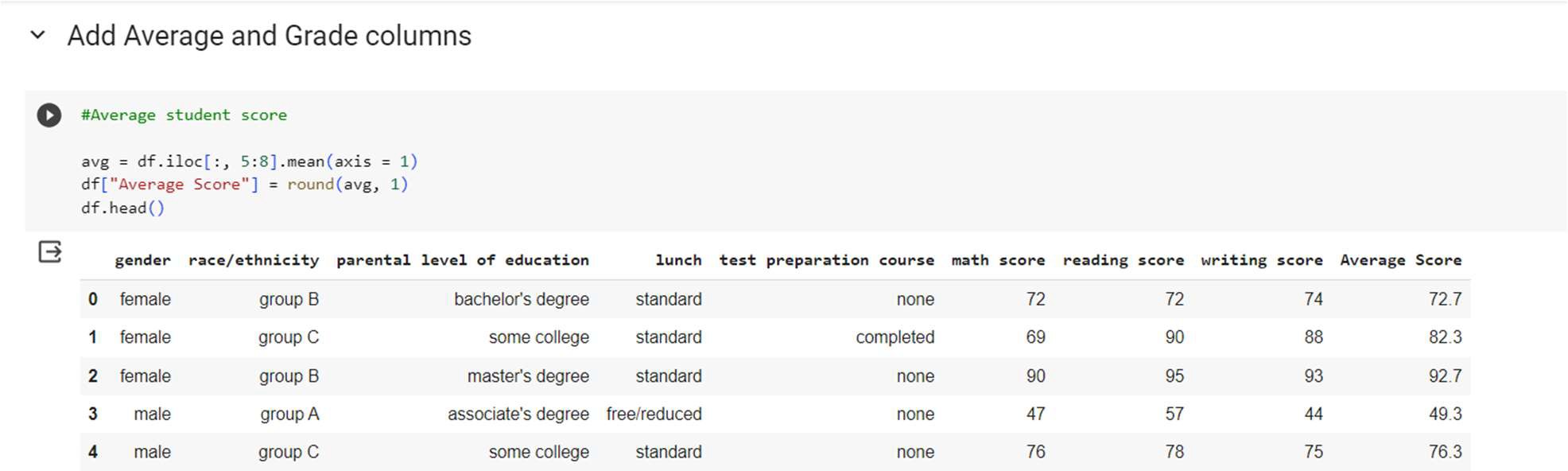
**CHAPTER - 5**

**SNAPSHOTS**

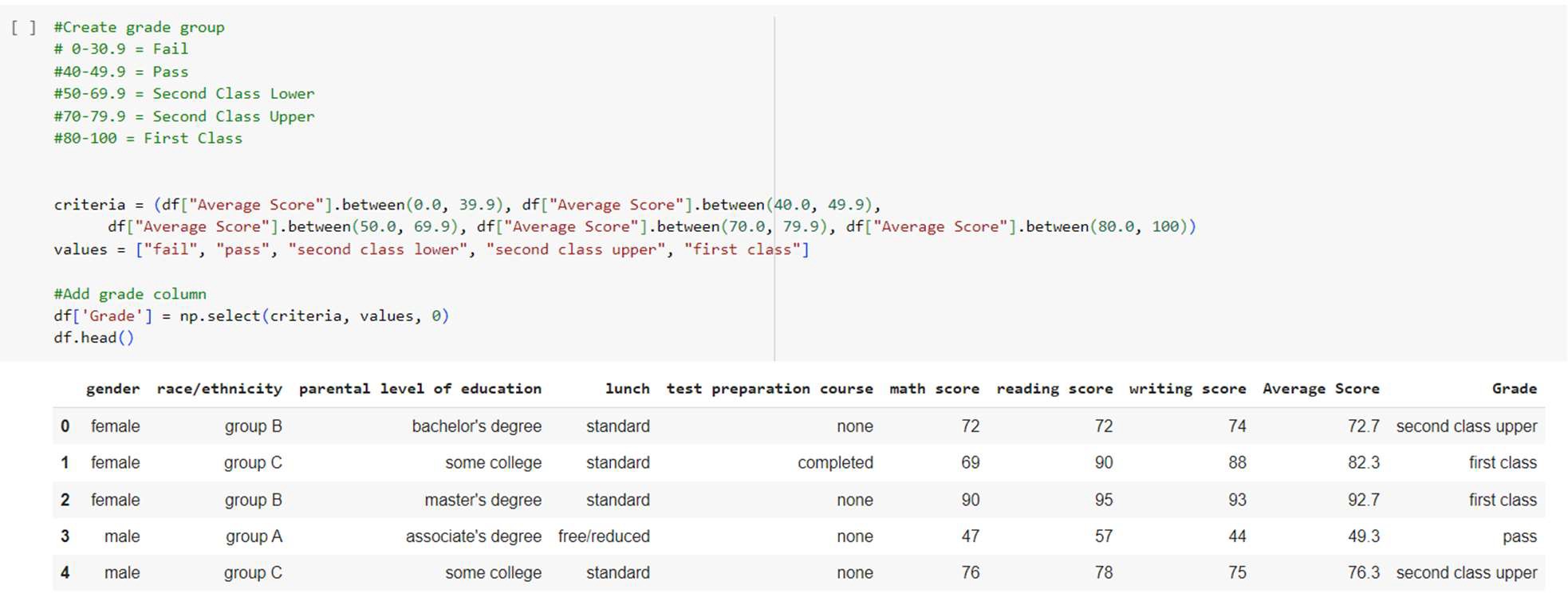
These lines import the necessary libraries, including NumPy, pandas, and scikit-learn modules for various machine learning algorithms.



This calculates the average score from the columns 'math score', 'reading score', and 'writing score' for each student and adds a new column 'Average Score' to the Data Frame.

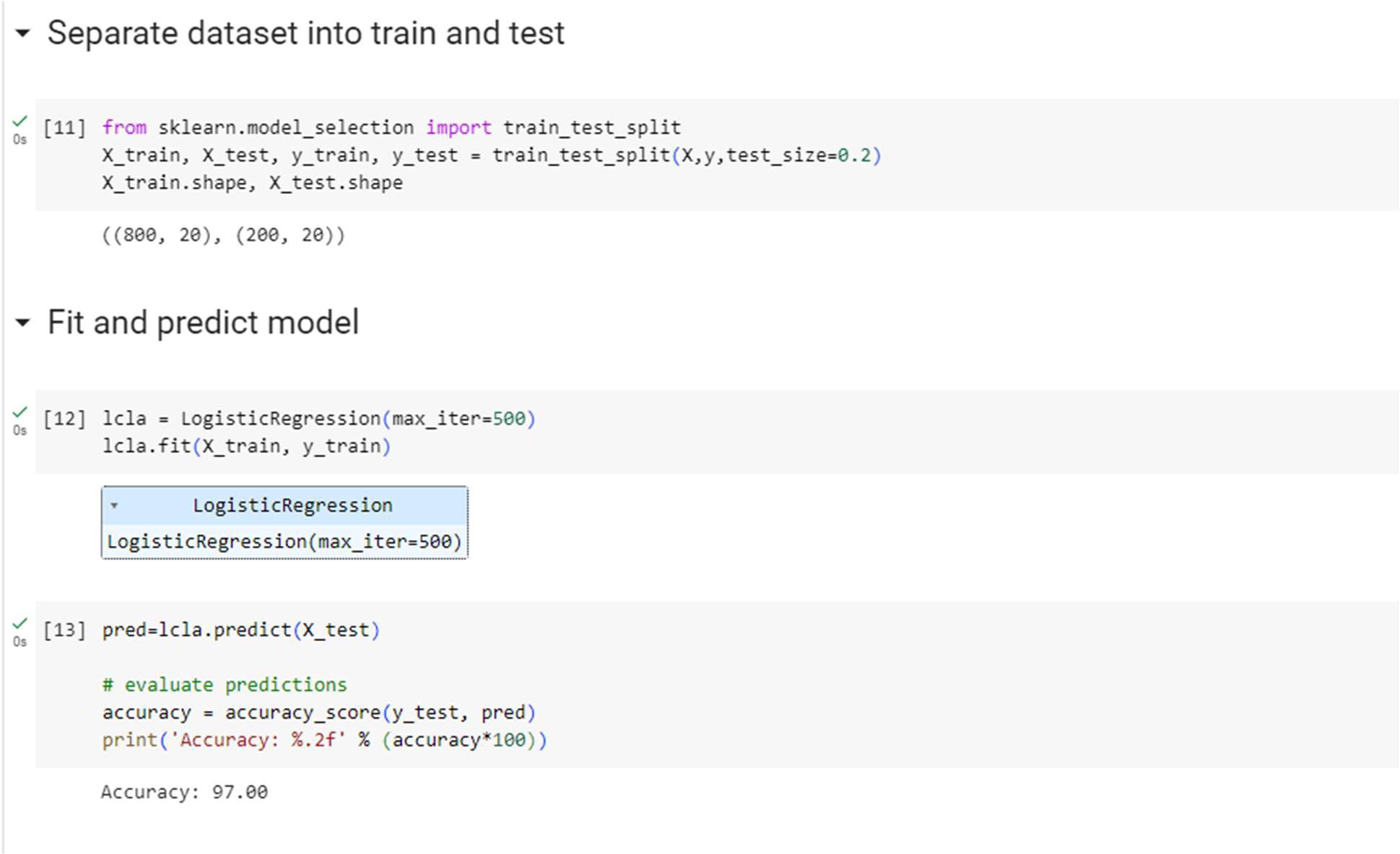


Setting up a grading system based on the average score



This splits the dataset into training and testing sets for classification and trains a logistic regression model for classification and makes predictions on the test set.

Thus, it evaluates the accuracy of the classification model.



This loop trains different regression models (linear regression, decision tree, random forest) and evaluates their performance.

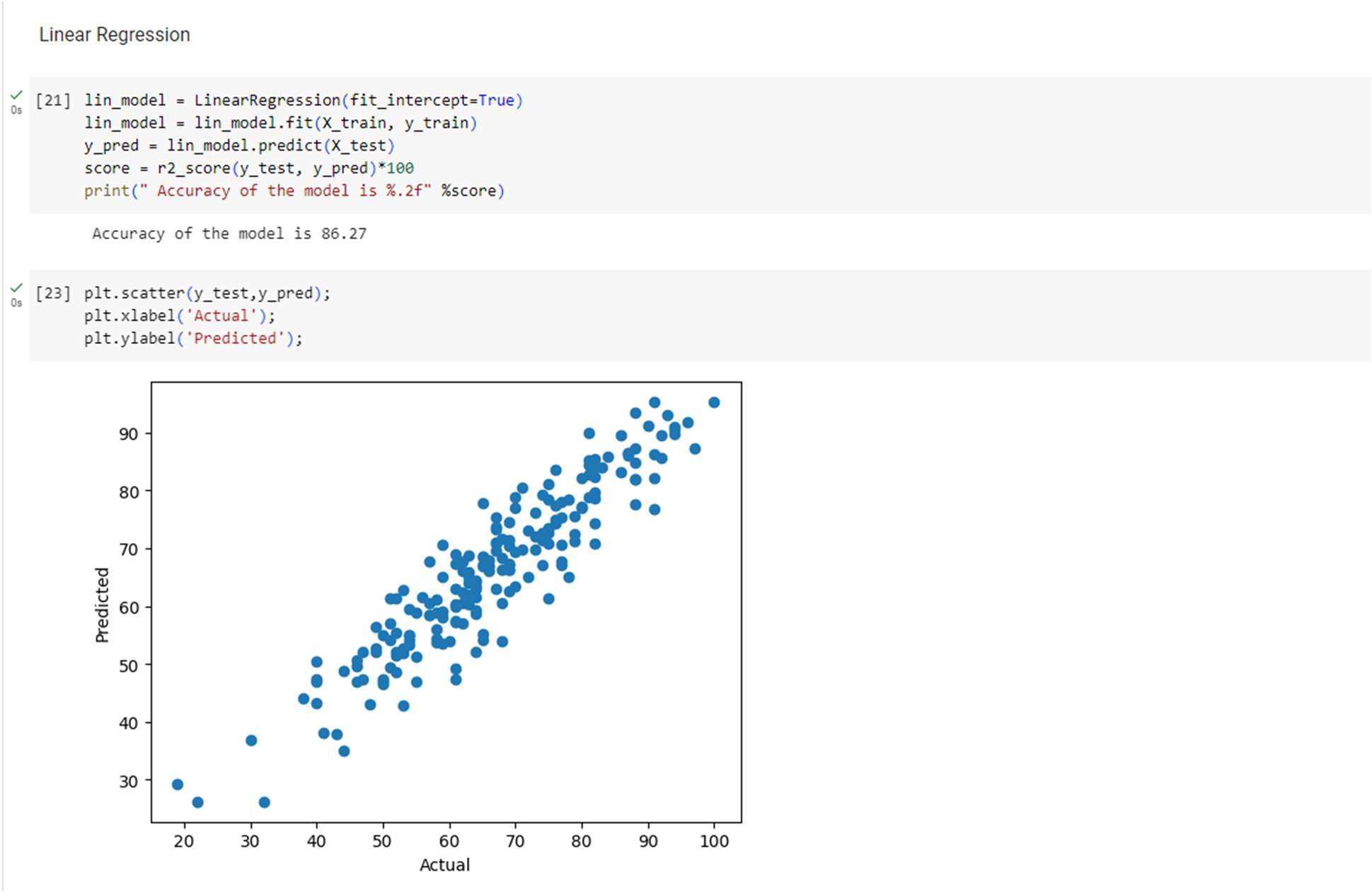


Since, the r2 score of the Linear regression is found to be higher, we further analyze the data using Linear Regression. The Accuracy of the model is calculated and printed.

A DataFrame (pred\_df) using pandas, which contains three columns:

* 'Actual Value': The actual values from the test set (y\_test).
* 'Predicted Value': The predicted values obtained from the regression model (y\_pred).
* 'Difference': The absolute difference between the actual and predicted values.

Using Matplotlib, a scatter plot of actual vs predicted values for regression is created.



**CHAPTER – 6**

**DECLARATION**

I, Keerthana S a student of 6th semester BCA, Computer Science and Application department, Bangalore Institute of Technology , Bengaluru hereby declare that internship project work entitled "CUSTOMER CHURN PREDICTION" has been carried out by me at Prinston Smart Engineers, Bengaluru and submitted in partial fulfilment of the course requirement for the award of the degree of Bachelor of Computer Science and Application of St Francis College, Bengaluru, during the academic year 2024-2025.

I also declare that, to the best of my knowledge and belief, the work reported here is not from the part of dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Place: Bengaluru

**KEERTHANA S [1BI21IS024]**

**CHAPTER - 7**

**CONCLUSION**

In conclusion, the endeavour to analyse student performance through the lens of machine learning has yielded significant insights into the intricate interplay of various factors influencing academic outcomes. This project, cantered around the 'StudentsPerformance.csv' dataset, embarked on a comprehensive exploration of demographic information and exam scores, employing both classification and regression models to unravel patterns and predictions.

The application of classification models facilitated the anticipation of students' grades, providing a nuanced understanding of how demographic and educational features contribute to academic success. Concurrently, regression models were adept at predicting math scores, shedding light on the nuanced relationships between specific variables and performance metrics.

The strategic utilization of Python libraries, including NumPy, pandas, Matplotlib, and scikit- learn, underscored the project's accessibility and relevance within the broader data science and educational communities. Leveraging these tools allowed for efficient data manipulation, exploration, and the implementation of machine learning algorithms, contributing to the project's methodological robustness.

As the project concludes, its outcomes offer a valuable contribution to the understanding of student performance and, by extension, the potential enhancement of educational practices. The predictive capabilities of machine learning models open avenues for early identification of at-risk students and tailored interventions, thereby fostering a data-driven approach to educational decision-making.

In the ever-evolving landscape of data science and education, this project stands as a testament to the power of interdisciplinary collaboration, providing a bridge between advanced analytics and the nuanced challenges faced in educational settings. Moving forward, the insights gained from this endeavour not only deepen our understanding of student performance but also underscore the importance of ethical considerations and the ongoing pursuit of knowledge at the intersection of data science and education.

**CHAPTER - 8**

**REFERENCES**

* https://pandas.pydata.org/docs/
* https://matplotlib.org/stable/index.html
* Regression Analysis with Python by Luca Massaron, Alberto Boschetti · 2016