Name:- Purva Deepak Kolhe

Qualification:- pursuing M.Sc(Data Science)

Email:- purvakolhe2002@gmail.com

Candidate No.:- AB-Tech 0052

• Introduction

➢ What is Data Science?

Data Science is a multidisciplinary field that involves the extraction of insights and knowledge from data using a combination of statistical, computational, and domain-specific techniques. It involves collecting, processing, analyzing, and interpreting large and complex datasets to discover patterns, trends, and insights that can be used to make informed decisions.

Data science draws on a wide range of disciplines, including statistics, computer science, mathematics, and domain-specific fields such as biology, social science, and finance. It typically involves the use of specialized software tools and programming languages such as Python, R, and SQL to clean and transform data, build predictive models, and visualize results.

Data science has become increasingly important in recent years as more and more organizations have recognized the value of data-driven decision-making. It is used in a variety of industries and applications, including healthcare, finance, marketing, and scientific research, among others.

➢ Introduction to Python

Python is a high-level, interpreted programming language that is known for its simplicity, readability, and versatility. It was first created by Guido van Rossum in the late 1980s and has since become one of the most popular programming languages in the world.

Python is used in a wide range of applications, including data science, web development, automation, machine learning, and artificial intelligence. It is easy to learn and has a large and supportive community of developers who contribute to its development and share resources and tools.

One of the key features of Python is its emphasis on code readability, which is achieved through the use of whitespace indentation rather than brackets. This makes it easy for developers to read and understand each other's code, which can be particularly useful in collaborative projects.

➢ What is Data Extraction

Data extraction is the process of collecting data from different sources, transforming it into a usable format, and storing it in a central location such as a database or data warehouse. The goal of data extraction is to gather information from multiple sources and consolidate it into a single location for analysis and reporting purposes.

Data extraction can involve various techniques depending on the type and structure of the data. For example, structured data, which is organized into a predefined format such as a table or spreadsheet, can be extracted using SQL queries or tools that automatically extract data from databases. Unstructured data, which does not have a predefined format and includes text, images, and multimedia content, requires more sophisticated techniques such as natural language processing and machine learning algorithms to extract information.

➢ Types of Data

Structured Data: Structured data is highly organized and easily searchable. It is usually stored in a database, spreadsheet or table format and can be easily analyzed using SQL queries. Examples of structured data include sales data, financial records, and customer data.

Unstructured Data: Unstructured data is data that is not organized in a specific format and does not fit into traditional relational databases. This type of data includes text, images, audio and video files, and social media data. Examples of unstructured data include emails, social media posts, and customer feedback.

Semi-Structured Data: Semi-structured data is a combination of structured and unstructured data. It has some organizational structure, but also contains elements that do not fit into a standard database structure. Semi-structured data can be found in formats such as XML or JSON. Examples of semi-structured data include log files, sensor data, and machine data.

➢ Data Processing

Data processing is the process of transforming raw data into meaningful information that can be used for decision-making. It involves various operations such as cleaning, organizing, filtering, validating, and transforming data to make it useful for analysis or storage.

The main steps involved in data processing include:

Data Cleaning: This involves removing errors, duplicates, and inconsistencies from the data to ensure that it is accurate and reliable.

Data Organization: This involves structuring the data in a way that makes it easier to search, sort, and retrieve. This may involve using a database or spreadsheet to organize the data.

Data Filtering: This involves selecting a subset of data that meets specific criteria. For example, filtering data by date range, geographic location, or customer segment.

Data Validation: This involves checking the accuracy and completeness of the data to ensure that it is reliable and free from errors.

Data Transformation: This involves converting data from one format to another, such as converting text data to numerical data, or aggregating data to a higher level of granularity.

➢ Data Wrangling

Data Transformation: This involves converting data from one format to another, such as changing the data type, scaling, or normalizing the data, and creating new variables.

Data Merging: This involves combining data from multiple sources, such as merging data from different databases or joining data from different tables.

Data Aggregation: This involves summarizing the data to a higher level of granularity, such as calculating the mean, median, or mode of a variable.

➢ Overview of Data Analytics Techniques

Descriptive Analytics: Descriptive analytics involves the analysis of past data to understand what happened and why it happened. This type of analysis includes techniques such as data visualization, summary statistics, and exploratory data analysis. The goal of descriptive analytics is to gain insights into historical trends, patterns, and relationships in the data.

Predictive Analytics: Predictive analytics involves the use of statistical models and machine learning algorithms to predict future outcomes based on historical data. This type of analysis includes techniques such as regression analysis, decision trees, and neural networks. The goal of predictive analytics is to identify patterns in the data that can be used to make predictions about future events.

Prescriptive Analytics: Prescriptive analytics involves the use of optimization techniques to identify the best course of action based on the predicted outcomes. This type of analysis includes techniques such as linear programming, integer programming, and nonlinear programming. The goal of prescriptive analytics is to make recommendations about the best course of action to achieve a specific outcome.

➢ Business Intelligence

Improved Decision Making: BI enables decision-makers to make data-driven decisions based on accurate and timely information.

Increased Efficiency: BI helps to streamline business operations by identifying areas for improvement and optimizing processes.

Competitive Advantage: BI provides organizations with insights into market trends, customer behavior, and competitor activities, giving them a competitive edge.

Better Financial Performance: BI can help organizations reduce costs, increase revenue, and improve profitability by identifying opportunities for growth and optimization.

• Python Fundamentals

Variables and Data Types: In Python, you can store data in variables, which can hold different data types, such as integers, floats, strings, and boolean values.

Operators: Python supports a range of operators, such as arithmetic, comparison, and logical operators. Operators are used to perform mathematical operations, compare values, and manipulate data.

Control Structures: Control structures, such as if-else statements and loops, are used to control the flow of a program. They allow you to execute code based on certain conditions or to repeat code multiple times.

Functions: Functions are used to group code that performs a specific task. Functions make your code more modular, easier to read and maintain, and can be reused in different parts of your program.

● Seaborn Python Library

Built-in datasets: Seaborn comes with several built-in datasets, including iris, titanic, and tips, which can be used for exploring and practicing data visualization.

Styling options: Seaborn provides a range of styling options, such as color palettes, grid styles, and font sizes, which can be used to customize the appearance of your plots.

Plot types: Seaborn provides a range of plot types, such as scatter plots, line plots, bar plots, and histograms, which can be used to visualize different types of data.

Statistical models: Seaborn provides several statistical models, such as linear regression and logistic regression, which can be used to fit and visualize statistical models.

Faceting: Seaborn provides the ability to create multiple plots based on subsets of the data, known as faceting. Faceting can be used to create informative and easy-to-read plots that highlight differences in the data.

● Matplotlib Python Library

Customization: Matplotlib provides a range of customization options, such as color maps, line styles, markers, and text formatting, which allow you to create visually appealing plots that highlight key insights in your data.

Subplots: Matplotlib provides the ability to create multiple plots within a single figure using subplots. This is useful for comparing multiple datasets or for creating complex visualizations that require multiple plots.

Backends: Matplotlib provides several backends, including PDF, SVG, and PNG, which allow you to save your plots in different file formats. Matplotlib also provides interactive backends, such as Qt5 and Tk, which allow you to create interactive plots.

Integration: Matplotlib integrates with other Python libraries, such as NumPy, Pandas, and Seaborn, making it easy to use with other data analysis tools.

● NumPy

➢ Array Creation

```
import numpy as np


my_list = [1, 2, 3, 4, 5]

my_array = np.array(my_list)


print(my_array)
```

➢ Data Types Objects

Numeric: Numeric data types are used to represent numbers in Python. The most common numeric data types are integers, floating-point numbers, and complex numbers.

String: String data types are used to represent text in Python. Strings are created using single quotes ('...') or double quotes ("...").

Boolean: Boolean data types are used to represent true or false values. In Python, the values True and False are used to represent boolean data.

List: List data types are used to represent a collection of values. Lists are created using square brackets ([...]) and can contain any type of data.

Tuple: Tuple data types are similar to lists, but they are immutable, meaning that their values cannot be changed after they are created. Tuples are created using parentheses ((...)).

Dictionary: Dictionary data types are used to represent a collection of key-value pairs. Dictionaries are created using curly braces ({...}) and can contain any type of data.

Set: Set data types are used to represent a collection of unique values. Sets are created using curly braces ({...}) or using the set() function.

- Basic Slicing

```
    my_list = [0, 1, 2, 3, 4, 5]


# Slice the list from index 1 to index 4

slice1 = my_list[1:4]


# Slice the list from index 0 to index 3

slice2 = my_list[:3]


# Slice the list from index 3 to the end of the list

slice3 = my_list[3:]


print(slice1)  # Output: [1, 2, 3]

print(slice2)  # Output: [0, 1, 2]

print(slice3)  # Output: [3, 4, 5]
```

Sorting, Searching

```
        my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5]
sorted_list = sorted(my_list)
print(sorted_list)  # Output: [1, 1, 2, 3, 4, 5, 5, 6, 9]
```

Pandas DataFrame and Analysis

Pandas DataFrame

A Pandas DataFrame is a two-dimensional tabular data structure provided by the Pandas library in Python. It is similar to a table or spreadsheet with rows and columns, where each column can contain different types of data (e.g., numbers, strings, dates).

Some key features of a Pandas DataFrame include:

Structure: The DataFrame consists of rows and columns, and each column has a name (column header) that uniquely identifies it.

Flexible indexing: Each row and column in a DataFrame has a unique label or index that can be used to access or manipulate the data.

Data handling: DataFrames can handle various types of data, including missing values, and provide methods for data cleaning, filtering, reshaping, and aggregating.

Efficient operations: Pandas DataFrame supports efficient operations on data, such as vectorized computations, grouping, sorting, and merging/joining with other DataFrames.

To work with a Pandas DataFrame, you typically import the Pandas library and create a DataFrame object by reading data from various sources, such as CSV files, databases, or by converting other data structures like lists or dictionaries into DataFrames. Once created, you can perform various operations on the DataFrame, such as selecting subsets of data, applying computations, aggregating data, merging or joining DataFrames, and visualizing the data using plots and charts.

Data Analysis

Data collection: Gathering relevant data from various sources, such as databases, files, APIs, or surveys.

Data cleaning: Preprocessing and cleaning the data to handle missing values, outliers, and inconsistencies. This step often involves data validation, normalization, and transformation.

Exploratory data analysis (EDA): Exploring and visualizing the data to gain insights, understand patterns, relationships, and distributions. This step helps in formulating hypotheses and identifying trends or outliers.

Statistical analysis: Applying statistical techniques to quantify and analyze relationships between variables, test hypotheses, and derive meaningful conclusions from the data. This may involve descriptive statistics, hypothesis testing, regression analysis, or other statistical methods.

Data modeling and machine learning: Building predictive models or applying machine learning algorithms to make predictions, classifications, or clustering based on the available data.

Interpretation and communication: Interpreting the results of the analysis, drawing conclusions, and effectively communicating the findings to stakeholders or decision-makers using visualizations, reports, or presentations.

Python, with libraries such as Pandas, NumPy, and scikit-learn, provides powerful tools for data analysis. These libraries offer functionalities for data manipulation, exploratory analysis, statistical analysis, and machine learning, making Python a popular choice for data analysis tasks.

• Statistics Fundamentals

➢ Scatter Plot

A scatter plot is a type of data visualization that represents the relationship between two variables by displaying individual data points on a Cartesian plane. Each data point is represented by a marker or dot, with the position on the plot determined by the values of the variables being compared.

Scatter plots are useful for visualizing the distribution, pattern, or correlation between two continuous variables. They can reveal the presence of clusters, trends, outliers, or the strength of the relationship between the variables.

Here's an example using Matplotlib to create a scatter plot:

```
import matplotlib.pyplot as plt

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

plt.scatter(x, y)
plt.show()
```

➢ Random Variables

A random variable can take on different values based on the underlying probability distribution. There are two types of random variables: discrete and continuous.

Discrete Random Variable: A discrete random variable can take on a countable number of distinct values. Examples of discrete random variables include the number of heads obtained when flipping a coin multiple times or the number of cars passing through an intersection in a given time period. The probability distribution of a discrete random variable can be described using a probability mass function (PMF), which assigns probabilities to each possible value.

Continuous Random Variable: A continuous random variable can take on any value within a specified range. It is associated with a continuous probability distribution. Examples of continuous random variables include the height of individuals, the time taken to complete a task, or the temperature in a given location. The probability distribution of a continuous random variable can be described using a probability density function (PDF), which specifies the relative likelihood of different values occurring.

➢ Mean, Variance, Standard Deviation

Mean: The mean, also known as the average, is a measure of central tendency. It is calculated by summing all the values in a dataset and dividing the sum by the number of values. The mean represents the typical or average value of the data.

Variance: Variance quantifies the spread or dispersion of a dataset. It measures how far each data point is from the mean. The variance is calculated by taking the average of the squared differences between each data point and the mean. A higher variance indicates greater variability in the data.

Standard Deviation: The standard deviation is the square root of the variance. It is often used as a more interpretable measure of dispersion because it is in the same unit as the original data. The standard deviation provides a measure of how much the data deviates from the mean. A larger standard deviation indicates a wider spread of data points.

In mathematical notation:

Mean (μ): $\mu = (x1 + x2 + ... + xn) / n$

Variance (σ^2): $\sigma^2 = [(x1 - \mu)^2 + (x2 - \mu)^2 + ... + (xn - \mu)^2] / n$

Standard Deviation (σ): $\sigma = sqrt(\sigma^2)$

➢ Probability Distribution Functions

Probability distribution functions (PDFs) are mathematical functions that describe the probabilities of various outcomes or values occurring in a random variable. PDFs are used to model and analyze random phenomena and are essential in probability theory and statistics. They provide a framework for understanding the likelihood of different outcomes and help in making predictions and conducting statistical inference.

Here are some common probability distribution functions:

Probability Mass Function (PMF): The PMF is used to describe the probability distribution of a discrete random variable. It gives the probability of each possible value that the random variable can take. The PMF assigns a probability to each value in the range of the random variable, and the sum of all probabilities is equal to 1.

Probability Density Function (PDF): The PDF is used to describe the probability distribution of a continuous random variable. Unlike the PMF, which assigns probabilities to individual values, the PDF assigns probabilities to intervals or ranges of values. The area under the PDF curve over a given range represents the probability of the random variable falling within that range. The PDF integrates to 1 over the entire range of the random variable.

Cumulative Distribution Function (CDF): The CDF gives the cumulative probability of a random variable taking on a value less than or equal to a given value. It provides information about the probability of observing a value less than or equal to a specific point in the distribution.

➢ Normal Distribution

The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution that is often used to model natural phenomena. The normal distribution is symmetric and bell-shaped, and it is characterized by two parameters: the mean (μ) and the standard deviation (σ).

The probability density function (PDF) of the normal distribution is given by:

$f(x) = (1 / (\sigma * sqrt(2\pi))) * exp(-((x-\mu)^2)/(2\sigma^2))$

The normal distribution has several important properties, such as:

It is symmetric about the mean, with the highest point of the curve located at the mean.

The area under the curve is equal to 1, meaning that the total probability of all possible outcomes is 1.

The mean and the standard deviation completely determine the shape of the distribution.

The normal distribution is continuous and extends from negative infinity to positive infinity, although the probability of observing values far from the mean is very low.

About 68% of the values in a normal distribution fall within one standard deviation of the mean, and about 95% of the values fall within two standard deviations of the mean.

➢ Bayes Theorem

Bayes' theorem has applications in various fields, including statistics, machine learning, artificial intelligence, medical diagnosis, spam filtering, and more. It enables us to make probabilistic inferences, perform Bayesian inference, and estimate unknown quantities based on available data and prior knowledge.

Mathematically, Bayes' theorem is stated as:

P(A|B) = (P(B|A) * P(A)) / P(B)

where:

P(A|B) represents the probability of event A given event B (the posterior probability).

P(B|A) represents the probability of event B given event A (the likelihood).

P(A) represents the prior probability of event A (the initial belief or probability before considering any evidence).

P(B) represents the probability of event B (the marginal probability or evidence).

• ML with Python

➢ Applications of ML:-

Image and Object Recognition: ML is used for tasks such as image classification, object detection, and facial recognition. It has applications in fields like computer vision, self-driving cars, security systems, and medical imaging.

Natural Language Processing (NLP): ML techniques are applied to analyze and understand human language. NLP is used in machine translation, sentiment analysis, chatbots, voice assistants, and information retrieval systems.

Fraud Detection: ML algorithms can identify patterns and anomalies in financial transactions to detect fraudulent activities. This is used in credit card fraud detection, insurance fraud detection, and cybersecurity.

Recommendation Systems: ML is used to build recommendation systems that provide personalized recommendations to users. These systems are used in e-commerce platforms, streaming services, and content recommendation.

Predictive Analytics: ML algorithms are used for predicting future outcomes based on historical data. This is applied in areas such as sales forecasting, demand prediction, financial market analysis, and predictive maintenance.

Healthcare: ML is used for medical diagnosis, disease prediction, drug discovery, and personalized medicine. It helps analyze medical images, genomics data, electronic health records, and clinical data.

➢ Supervised vs Unsupervised Learning:-

Supervised Learning: Supervised learning involves training a model on labeled data, where the input data is paired with corresponding target labels or outcomes. The goal is to learn a mapping function that can make accurate predictions or classifications for new, unseen data.

Supervised learning algorithms include various techniques such as linear regression, logistic regression, decision trees, support vector machines (SVM), random forests, and neural networks. Examples of supervised learning applications include email spam classification, sentiment analysis, image recognition, and medical diagnosis.

Unsupervised Learning: Unsupervised learning, in contrast, deals with unlabeled data, where the input features are provided without any corresponding target labels. The objective is to discover hidden patterns, structures, or relationships within the data.

Unsupervised learning is useful for exploratory data analysis, anomaly detection, data visualization, and data preprocessing. It can help identify market segments, recommend related items, detect unusual patterns in network traffic, and more.

➢ Best Python Libraries For ML

scikit-learn: scikit-learn is a versatile and comprehensive library for machine learning. It provides a wide range of algorithms for classification, regression, clustering, dimensionality reduction, and model evaluation. It is known for its user-friendly API and extensive documentation, making it a great choice for beginners.

Pandas: Pandas is a powerful data manipulation and analysis library. It provides data structures such as DataFrames, which are highly useful for handling structured data. Pandas simplifies data preprocessing and exploration tasks, making it an essential tool in machine learning workflows.

Matplotlib: Matplotlib is a plotting library for creating visualizations in Python. It offers a wide range of plots and customization options, making it useful for visualizing data, model performance, and other analysis results.

Seaborn: Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a higher-level interface and offers additional statistical visualizations and styling options. Seaborn is particularly handy for creating attractive and informative statistical graphics.

- # Regression

➢ Data Science Linear Regression:-

Linear regression is a widely used statistical technique in data science and machine learning for modeling the relationship between a dependent variable and one or more independent variables. It is a type of supervised learning algorithm used for regression tasks, where the goal is to predict a continuous numeric value.

In linear regression, the relationship between the dependent variable (often denoted as "y") and independent variables (often denoted as "x") is assumed to be linear. The aim is to find the best-fitting line that minimizes the differences between the predicted values and the actual values of the dependent variable. This line is determined by estimating the slope and intercept of the linear equation.

The simple linear regression model with one independent variable can be represented as:

$y = \beta 0 + \beta 1 * x + \varepsilon$

where:

y is the dependent variable to be predicted.

x is the independent variable.

β0 is the y-intercept or constant term.

β1 is the coefficient or slope that represents the change in y for a one-unit change in x.

ε is the error term or residual, which captures the variability that is not explained by the linear relationship.

The goal of linear regression is to estimate the values of β0 and β1 that minimize the sum of squared differences between the predicted and actual values of y. This is typically done using a method called ordinary least squares (OLS), which calculates the best-fit line by minimizing the sum of squared residuals.

➢ Introduction to Logistic Regression:-

Logistic regression is a statistical modeling technique used to predict binary outcomes or perform binary classification tasks. It is a type of supervised learning algorithm that is widely used in various fields, including data science, machine learning, and statistics.

Unlike linear regression, which predicts continuous numeric values, logistic regression predicts the probability of an event or the likelihood of a binary outcome. The dependent variable in logistic regression is categorical and takes one of two possible values, typically represented as 0 and 1.

The logistic regression model applies the logistic function (also known as the sigmoid function) to transform the linear regression equation into a bounded range between 0 and 1. The logistic function is defined as:

$p = 1 / (1 + e^{(-z)})$

where:

p represents the probability of the binary outcome (e.g., the probability of an event occurring).

z is the linear combination of the independent variables and their respective coefficients.

The logistic regression equation can be represented as:

$\log(p / (1 - p)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n * x_n$