INTERNSHIP REPORT

A report submitted in partial fulfillment of the requirements of the Award of Degree of

**BACHELOR OF TECHNOLOGY**

**In**

**Computer Science and Engineering**

**By**

# MUKESH MAHATO DHANUK

**REGD\_NO: 21781A05K0**

## Domain: DATA SCIENCE AND MACHINE LEARNING

**(Duration - 2023)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SRI VENKATESWARA COLLEGE OF ENGINEERING & TECHNOLOGY**

**(An Autonomous Institution)**

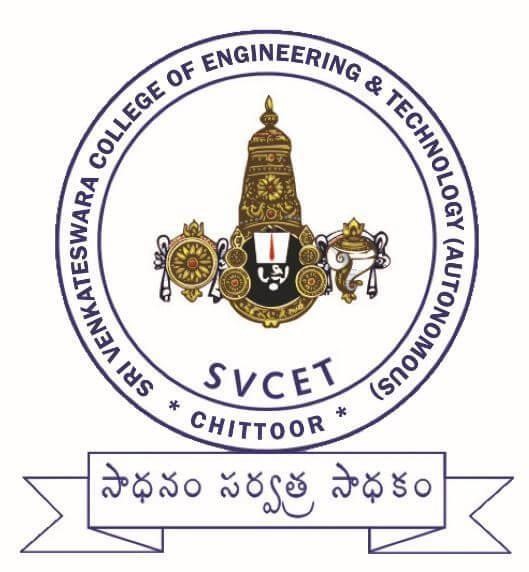
Approved by AICTE, Permanently affiliated to JNTU, Anantapur **R.V.S.NAGAR,CHITOOR.**  **ANDHRA PRADESH.**

1

**SRI VENKATESWARA COLLEGE OF ENGINEERING & TECHNOLOGY**

(An Autonomous Institution)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the “**Internship report”** submitted by

**NITESH KUMAR YADAV** (Regd. No.: 21781A05K1) is work

done by her and submitted during 2022 – 2023 academic year, in partial

fulfillment of the requirements for the award of the degree of BACHELOR OF

TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING, at ADS

Indo-Euro Synchronization – Tadepalle - Vijayawada.

**Mr. P. Thirumurugan Dr. P Jyotheeswari**

Coordinator of Internship Head Of Department

# CERTIFICATE OF INTERNSHIP



ACKNOWLEDGEMENT

First, I would like to thank Dr.V.V.N.Raj, HR, Head, of **ADS IndoEuro Synchronization**, for giving me the opportunity to do an internship within the organization.

I also would like all the people that worked along with me **NANOMINDZ, Vishakapatanam** with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am highly indebted to Director

Dr.R. Venkataswamy and Principal Dr.M. MOHAN BABU for the

facilities provided to accomplish this internship. I would like to thank my Head of the Department **Dr.P. Jyotheeswari** for his constructive criticism throughout my internship.

I would like to thank College internship coordinator **Mr.P. Thirumurugan**

**M.E..**internship coordinator Department of CSE for their support and advices to get and complete internship in above said organization.

I am extremely great full to my department staff members and friends who helped me in successful completion of this internship.

**NITESH KUMAR YADAV**

**(21781A05K1)**

# ABSTRACT

Data science encompasses a set of principles, problem definitions, algorithms, and processes for extracting nonobvious and useful patterns from large data sets. Many of the elements of data science have been developed in related fields such as machine learning and data mining. In fact, the terms data science, machine learning, and data mining are often used interchangeably. The commonality across these disciplines is a focus on improving decision making through the analysis of data. However, although data science borrows from these other fields, it is broader in scope. Machine learning (ML) focuses on the design and evaluation of algorithms for extracting patterns from data. Data mining generally deals with the analysis of structured data and often implies an emphasis on commercial applications. Data science takes all of these considerations into account but also takes up other challenges, such as the capturing, cleaning, and transforming of unstructured social media and web data; the use of big-data technologies to store and process big, unstructured data sets; and questions related to data ethics and regulation.

## INDEX

**S.no CONTENTS**

1. Introduction

1.1 Data Science Overview

1. EXCEL
2. SQL
3. Python Programming
4. NumPy
5. Tableau
6. Power BI

**WEEKLY OVERVIEW**

**OF**

**INTERNSHIP ACTIVITIES**

|  |  |  |
| --- | --- | --- |
| S.No | Module Name | Date |
|  | Introduction | 18/03/2023 |
| 1 | Excel | 21/03/2023 |
| 2 | SQL | 11/04/2023 |
| 3 | Python Programming | 25/04/2023 |
| 4 | NumPy | 16/05/2023 |
| 5 | Tableau | 13/06/2023 |
| 6 | Power BI | 25/07/2023 |

**s**

**INTRODUCTION**

Data Science is about data gathering, analysis and decisionmaking.

Data Science is about finding patterns in data, through analysis, and make future predictions.

By using Data Science, companies are able to make:

* Better decisions (should we choose A or B)
* Predictive analysis (what will happen next?)
* Pattern discoveries (find pattern, or maybe hidden information in the data)

# 1.EXCEL

**Spreadsheets | Importing Data to Excel**

**Spread Sheet:**

Spreadsheets are a powerful tool in Excel for organizing and analyzing data. They provide a way to store data in rows and columns, perform calculations and analysis, and create visual representations of the data. In this section, we will cover some basic concepts of spreadsheets in Excel that relate to the questions mentioned earlier.

**Working With Cells And Data:**

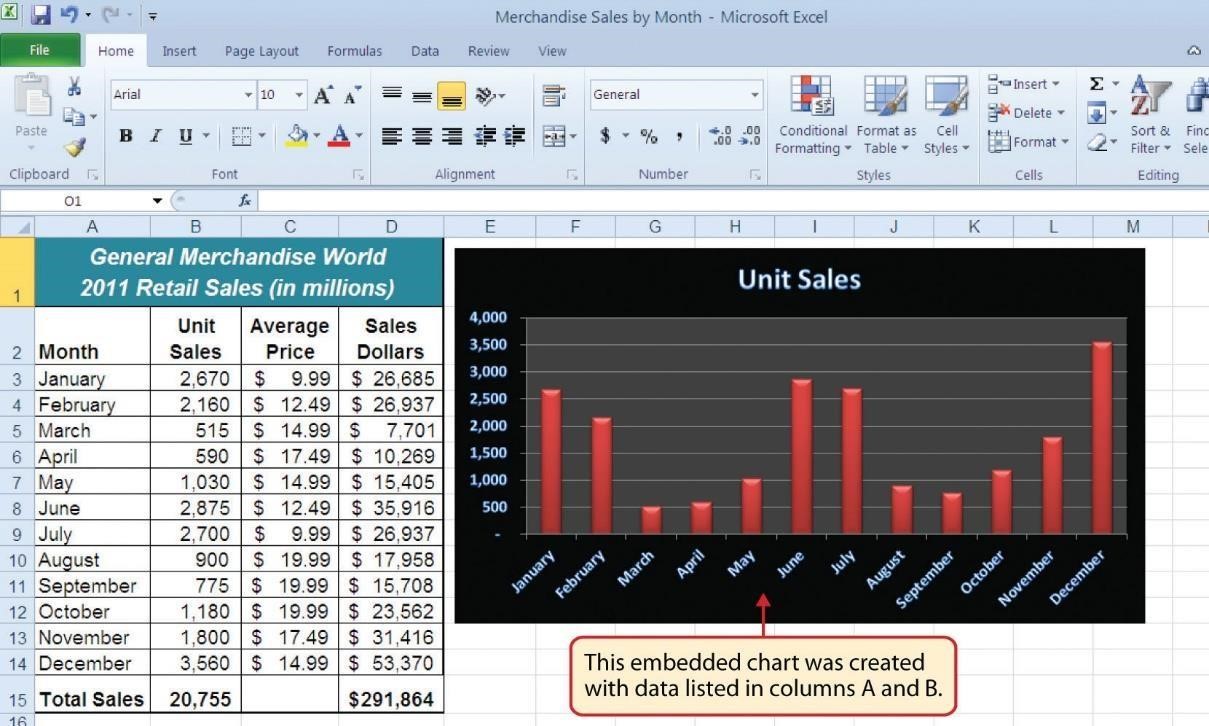
* In Excel, a spreadsheet is made up of cells, which are identified by their column and row reference, such as Al or B5, Each cell can contain text, numbers, formulas, or functions. To enter data into a cell, simply click on the cell and start typing. You can also copy and paste data into cells, or use the fill handle to drag and fill cells with a pattern or series of data.
* Example: To enter data into a cell, click on the cell and start typing. For example, to enter the value "10" into cell Al, click on cell Al and type 10.

**Basic Formulas And Functions:**

* Formulas and functions are used in Excel to perform calculations and analysis on data.

Formulas are expressions that begin with an equal sign (=) and can include arithmetic operators (+... /), cell references, and functions. Functions are pre-built formulas that can perform specific tasks, such as calculating averages, sums, and counts. Some common functions used in Excel include SUM, AVERAGE, COUNT, MAX, and MIN.

* Example: To calculate the sum of a range of numbers, use the SUM function. For example, to calculate the sum of the numbers in cells Al through A5, enter =SUM(A1:A5)" in a cell.



**Formatting And Styling:**

* Formatting and styling are used in Excel to make data more visually appealing and easier to read. You can format cells to change the appearance of text, numbers, and dates, such as changing font size, color, or style. You can also apply number formatting, such as currency or percentage, and use conditional formatting to highlight cells that meet certain criteria, such as values that are above or below a certain threshold.

Example: To apply number formatting to a cell, select the cell and click on the "Number Format drop-down menu in the Home tab. Choose the desired number format, such as Percentage".

**Sorting And Filtering :**

Sorting and filtering are used in Excel to organize and analyze data. You can sort data in ascending or descending order based on the values in one or more columns. You can also filter data to display only the rows that meet certain criteria, such as values that are greater than a certain threshold or dates that fall within a specific range.

* Example: To sort data in ascending or descending order, select the data range and click on the "Sort & Filter" button in the Home tab. Choose the desired sorting option, such as "Sort A to Z or "Sort Largest to Smallest".

**Charts And Graphs :**

Charts and graphs are used in Excel to create visual representations of data. You can create a variety of charts and graphs, such as bar charts, line charts, pie charts, and scatter plots. You can also customize the appearance of charts and graphs, such as changing the colors, labels, and axis titles.

Example: To create a bar chart, select the data range and click on the "Insert" tab. Click on the "Bar" chart type and choose the desired chart style. Customize the appearance of the chart by changing the colors, labels, and axis titles.

**Collaboration And Sharing :**

* Excel provides features for collaboration and sharing, such as sharing a workbook with multiple users, protecting a workbook with a password, and tracking changes made by other users. You can also use Excel Online to collaborate with others in real time and share works with anyone who has access to the internet.

Example: To share a workbook with multiple users, click on the "Share" button in the top right corner of the screen. Choose the desired sharing option, such as "Invite People" or "Get a Sharing Link.

**Importing Data Into Excel:**

* Excel allows you to import data from a variety of sources, including text files, CSV files, databases, and other Excel workbooks. Importing data into Excel can save you time and effort in entering data manually, and it can also help you to work with data from multiple sources

**Steps To Import Data Into Excel:**

* Open a new or existing workbook in Excel.
* Click on the "Data" tab in the ribbon at the top of the screen.
* Click on the "From Text/CSV" or "From Other Sources" button in the "Get & Transform Data
* Choose the file or data source you want to import, such as a text file, database, or web page.
* Follow the prompts to specify the import settings, such as the delimiter character, encoding, and data type.
* Preview the data to ensure it has been imported correctly.
* Click on the "Load" button to import the data into Excel.

**Spreadsheet Functions to organize Data |Filtering |Pivot Tables | Charts:**

SPREAD SHEET FUNCTIONS:

1.SORT function: This function is used to sort a range of data in either ascending or descending order based on one or more columns.

Example: =SORT(A2:E20, 3, FALSE)

This formula sorts the range A2:E20 by the values in the third column (C) in descending order.

2.FILTER function: This function is used to extract data from a range based on specified criteria. Example: FILTER(A2:E20, D2:D20 > 100)

This formula extracts all the rows from the range A2:E20 where the value in column D is greater than 100.

3.SUMIF function: This function is used to sum values in a range that meet specified criteria. This formula sums all the values in column B where the corresponding value in column A is "Apples".

4.COUNTIF function: This function is used to count the number of cells in a range that meet specified criteria.

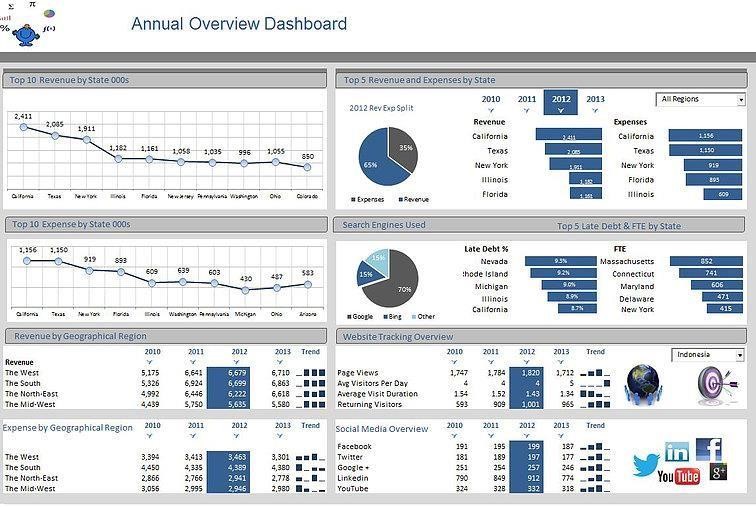
5.VLOOKUP function: This function is used to find a value in one table and return a corresponding value from another table.

6.INDEX/MATCH function combination: This function combination is used to find a value in one table and return a corresponding value from another table.

7.CONCATENATE function: This function is used to join two or more strings of text together. Example: CONCATENATE(A2, B2)

This formula joins the values in cells A2 and B2 together with a space between them.

8.LEFT/RIGHT/MID functions: These functions are used to extract a specified number of characters from the left, right, or middle of a cell



**FILTERING DATA IN MICROSOFT EXCEL:**

Filtering data in Excel allows you to display only the rows that meet certain criteria. This can be useful when you have a large dataset and you only want to focus on specific information.

* To apply a filter to a range of data in Excel:
* Select the range of cells that you want to filter.
* On the Home tab, click the Filter button in the Editing group. This will add filter arrows to each column header in your selected range.
* Click the filter arrow for the column that you want to filter, and select the criteria that 'you want to use to filter your data. • Excel will display only the rows that meet your filter criteria.

**PIVOT TABLE IN MICROSOFT EXCEL:**

A pivot table is a powerful tool in Excel that allows you to summarize and analyze large amounts of data in a concise and meaningful way. It allows you to group, filter, and analyze data based on different criteria, and provides a dynamic view of your data that can help you identify trends and patterns.

To create a pivot table in Excel:

* Select the range of cells that contains your data.
* Click the "Insert" tab on the ribbon.
* Click the "Pivot Table" button. This will open the "Create PivotTable" dialog box.
* DATA VALIDATION IN MICROSOFT EXCEL
* Data validation is a feature in Excel that allows you to control the type and range of data that can be entered into a cell. It helps you to ensure that your data is accurate and consistent, and prevents errors and inconsistencies in your worksheets.

To apply data validation in Excel:

* Select the cell or range of cells that you want to apply validation to.
* Click the "Data" tab on the ribbon.
* Click the "Data Validation" button.
* Choose the type of validation you want to apply, such as "Whole Number", "Decimal", "List", or "Date". Set the criteria for validation by choosing the rules that you want to apply. For example, you can set a rule to allow only values between a certain range, or values that are in a specific list.

1.SORT function: This function is used to sort a range of data in either ascending or descending order based on one or more columns.

Example: =SORT(A2:E20, 3, FALSE)

This formula sorts the range A2:E20 by the values in the third column (C) in descending order.

2.FILTER function: This function is used to extract data from a range based on specified criteria.

Example: FILTER(A2:E20, D2:D20 > 100)

This formula extracts all the rows from the range A2:E20 where the value in column D is greater than 100.

3.SUMIF function: This function is used to sum values in a range that meet specified criteria. This formula sums all the values in column B where the corresponding value in column A is "Apples".

4.COUNTIF function: This function is used to count the number of cells in a range that meet specified criteria.

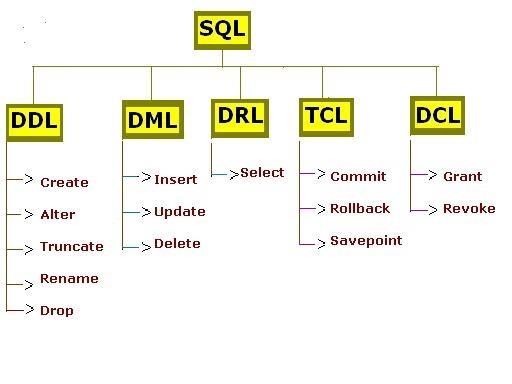
2.**SQL**

**SQL Overview | Relational Data Base Concepts :**

SQL, or Structured Query Language, is a programming language used to manage and manipulate relational databases. It is a standard language used by most database systems, including MySQL, Oracle, SQL Server, and PostgreSQL.

SQL allows users to create and manage databases, as well as perform tasks such as inserting, updating, and retrieving data from a database. It uses a set of commands and syntax to communicate with the database and manipulate the data stored within it. Some of the most commonly used SQL commands include:

* SELECT: retrieves data from one or more tables in the database.
* INSERT: adds new data to a table in the database.
* UPDATE: modifies existing data in a table in the database.
* DELETE: removes data from a table in the database.
* CREATE: creates a new table or database in the database system.
* DROP: deletes a table or database from the database system. Here are some examples of how these commands are used in SQL:



1.SELECT:

The SELECT command is used to retrieve data from one or more tables in the database. Here's an example of a simple SELECT statement:

SELECT FROM customers;

This statement retrieves all the data from the "customers table in the database. The\*\*\* character is a wildcard that tells SQL to retrieve all the columns in the table.

2. INSERT:

The INSERT command is used to add new data to a table in the database.

Here's an example of a simple INSERT statement:

INSERT INTO customers (first\_name, last\_name, email)

VALUES ('John', 'Doe',’johndoe@example.com’);

3.UPDATE:

The UPDATE command is used to modify existing data in a table in the database.

1. DELETE:

The DELETE command is used to remove data from a table in the database.

Here's an example of a simple DELETE statement; DELETE FROM customers WHERE id = 1;

This statement deletes the row with an "id" value of 1 from the "customers table in the database.

1. CREATE:

The CREATE command is used to create a new table or database in the database system.

Here's an example of a simple CREATE statement:

CREATE TABLE orders ( id INT PRIMARY KEY, customer id INT, order date DATE, total DECIMAL(10, 2)

):

**RELATIONAL DATABASE CONCEPTS:**

A relational database is a type of database that organizes data into tables consisting of rows and columns. Each table in a relational database has a unique name and a set of columns that define the types of data that can be stored in that table. Each row in a table represents a single record, and each column represents a particular attribute of that record.

**SQL.** (Structured Query Language) is a programming language that is used to manage and manipulate data in a relational database. SQL provides a standardized way to interact with databases, making it easy to create, modify, and retrieve data.

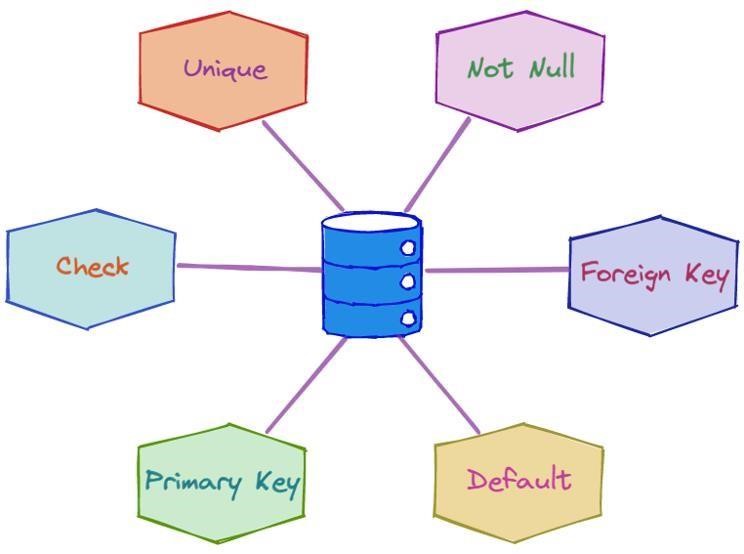
**Here are some common SQL concepts that you should be familiar with:**

**Tables:** Tables are the basic building blocks of a relational database. They consist of rows and columns, where each row represents a record and each column represents a specific attribute of that record.

Example: Consider a table called "employees" that contains information about employees in a company. It might have columns for employee ID, name, email, phone number, and department. **Primary keys:** Primary keys are unique identifiers for each record in a table. They help to ensure that each record in a table is unique and can be easily accessed.

**Foreign keys**: Foreign keys are used to establish relationships between tables in a relational database. A foreign key in one table refers to the primary key in another table.

Queries: Queries are used to retrieve data from a database. SQL provides a variety of commands that can be used to query a database, such as SELECT, INSERT, UPDATE



**SQL Data Grouping and Summarizing**

SQL (Structured Query Language) is a standard programming language used to manage relational databases. Grouping and summarizing data in SQL is a common requirement in data analysis and reporting. In SQL, we use the GROUP BY clause to group data based on one or more columns and the aggregate functions such as SUM, COUNT, AVG, MAX, and MIN to summarize the data.

**Grouping by a single column:**

Suppose we have a table called "sales" with the following columns: "date", "product", "price", and "quantity". We want to group the data by the "product" column and find the total sales for each product.

ELECT product, SUM(price quantity) as total\_sales

FROM sales

GROUP BY product;

**Grouping by multiple columns:**

Suppose we want to group the data by both "date" and "product" columns and find the total sales for each combination.

SELECT date, product, SUM(price quantity) as total\_sales

FROM sales

GROUP BY date, product,

Using aggregate functions:

Suppose we want to find the total sales, average price, and maximum quantity sold for each product.

SELECT product, SUM(price quantity) as total\_sales, AVG(price) as avg\_price, MAX(quantity) as max quantity

FROM sales

GROUP BY product;

Filtering the results:

Suppose we want to find the total sales for each product but only include the products with total sales greater than $10,000.

SELECT product, SUM(price quantity) as total\_sales

FROM sales

GROUP BY product

HAVING SUM(price quantity)> 10000;

Grouping by date and finding the total sales for each month:

Suppose we want to find the total sales for each month by grouping the data by the "date" column. SELECT DATE TRUNC('month, date) as month, SUM(price quantity) as total sales FROM sales GROUP BY DATE\_TRUNC(month, date);

Grouping by date and product and finding the average price for each product per month: Suppose we want to find the average price for each product per month by grouping the data by the "date" and "product" columns.

Example:

SELECT DATE TRUNC('month, date) as month, product, AVG(price) as avg price FROM sales

GROUP BY DATE\_TRUNC('month, date), product;

This will give us a result set with three columns: "month", "product", and "avg price. Each row will represent a product, a month, and its average price.

Finding the top-selling products:

Suppose we want to find the top-selling products by grouping the data by the "product" column and sorting the result by the total sales in descending order.

SELECT product, SUM(price quantity) as total sales

FROM sales

GROUP BY product

ORDER BY total\_sales DESC

LIMIT 10;

This will give us a result set with two columns: "product" and "total\_sales". Only the top 10 products with the highest total sales will be included in the result set.

**Conclusion:**

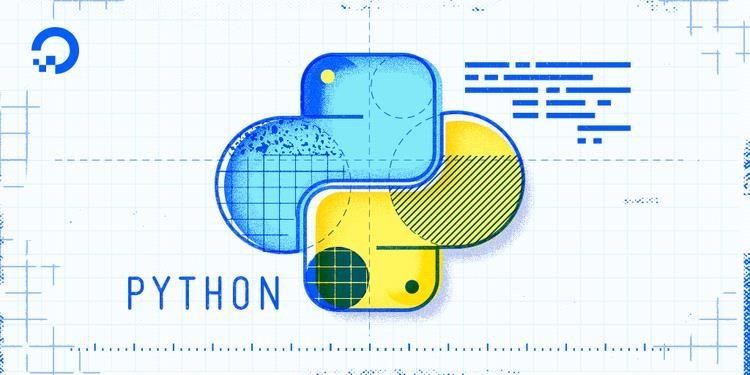
In conclusion, grouping and summarizing data in SQL is a powerful feature that allows us to analyze and report on large datasets. we can easily group data by one or more columns and perform various calculations on the grouped data.

### 3.PYTHON PROGRAMMING

**Importance of Conditional Statements:**

There are no programs which cannot be executed without conditional statements. Any problem statement given to you must be solved using these statements and hence these statements are very important in programming.

All companies which are looking for the programming skills will check from this basic level coding.



**Let's Warm Up:**

We may require to execute some lines of code under some conditions. For example, I can allow a person to vote only when his age is above 18years. A student gets promoted only when his attendance is above 75%. In order to satisfy the statements I need to check with some conditions. And hence we use conditional statements in programming languages.

Conditional statements are also known as decision-making statements. Let us clearly understand this with a simple example. A number is given to me and I need to check whether the given number is even or odd.

If the given number is divisible by 2, it is a even number else it is an odd number. Here, divisibility is the condition check for a number and for this condition check we use conditional statements

Here are some conditional statements in python: **If statement:**

We use if statement to check whether the condition is True or not. If the condition is True then the statements inside if clause we get executed. Indentation places a main role here, Statements which follow indentation are inside if clause and will be executed if the statement is True Syntax:

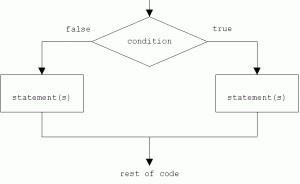
if <condition>:

#statement-inside

#statement-inside

#statement-inside

#statement-outside #statement-outside **Flowchart:**



**Example:**

a=10 b=10

if a==b: print("Inside If") print("Outside If") The condition is a==b.

If a value, 10 is equal to b value 10 then the statement inside if statement, print("Inside If") will be executed and then the next statement will be executed, Hence the output is: Inside If Outside If >>>

**If - else statement:**

We use if-else clause to check whether the given condition is True or False. If the condition is True then if clause will be executed and if the condition is False then else clause will be executed. The best example for if-else is even or odd program. If the given number is divisible by 2 then the number is even number else the number is odd number.

Syntax:

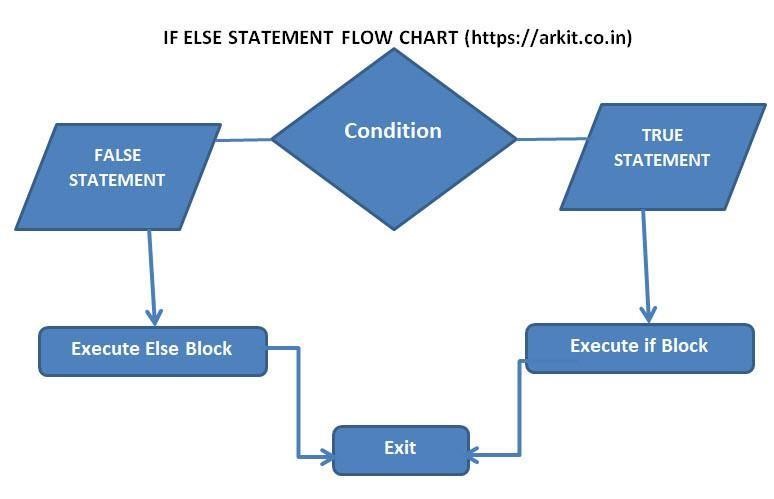
if <condition>:

#statement-if #statement-if else:

#statement-else

#statement-else

**Flowchart:**



Let us consider even or odd program, n

= 5 if n%2==0: print("Even") else: print("Odd")Here, n value is given as 5. If statement is holding a condition to check whether n is divisible by 2. If True, print("Even") will be executed else print("Odd") will be executed.

IMPORTANCE OF Dictionary:

A dictionary is a collection which is unordered, changeable and indexed.

Many organizations like Cognizant, Microsoft, IBM, Accenture, Infosys Infytq ask questions on Tuples.

LET'S WARM UP:

• In python dictionaries are written with curly braces, and they have keys and values. Creating a Dictionary:

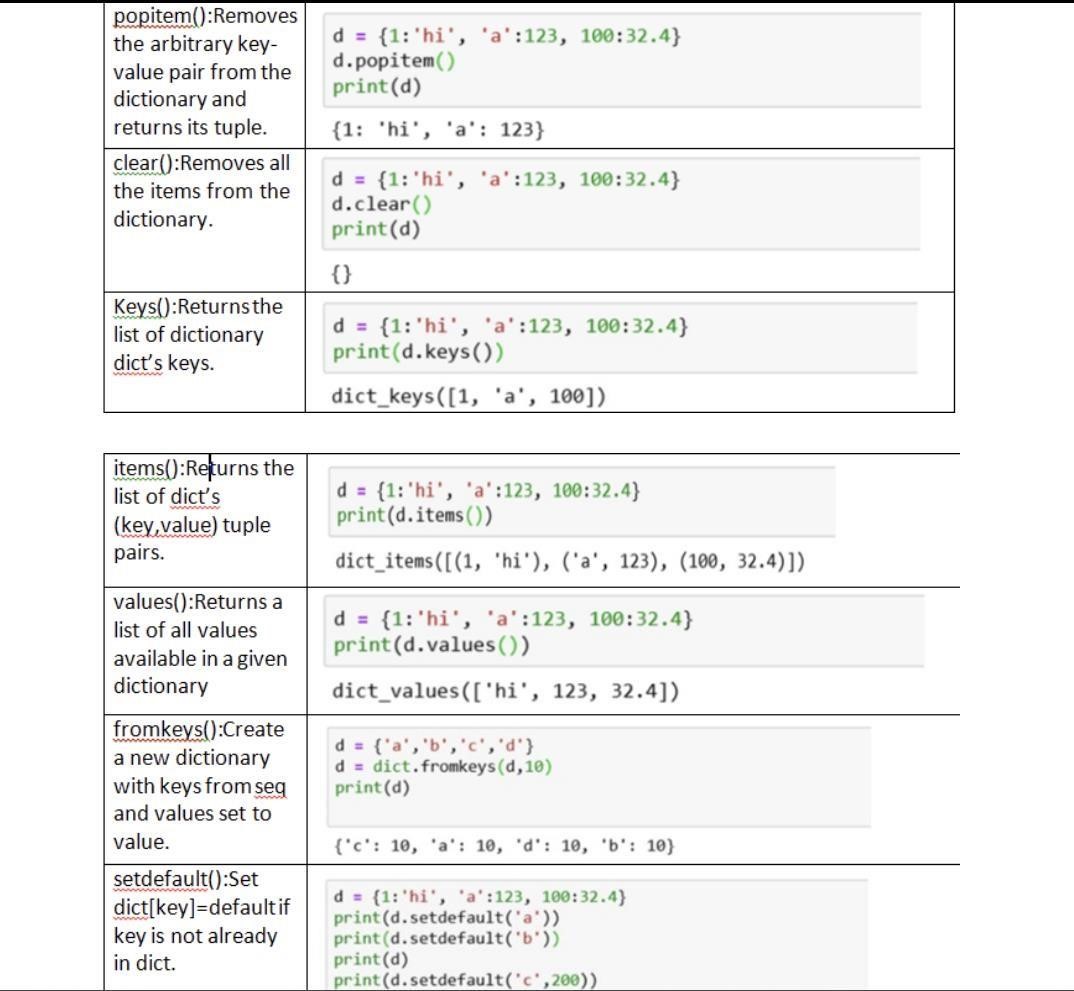
**Example:**

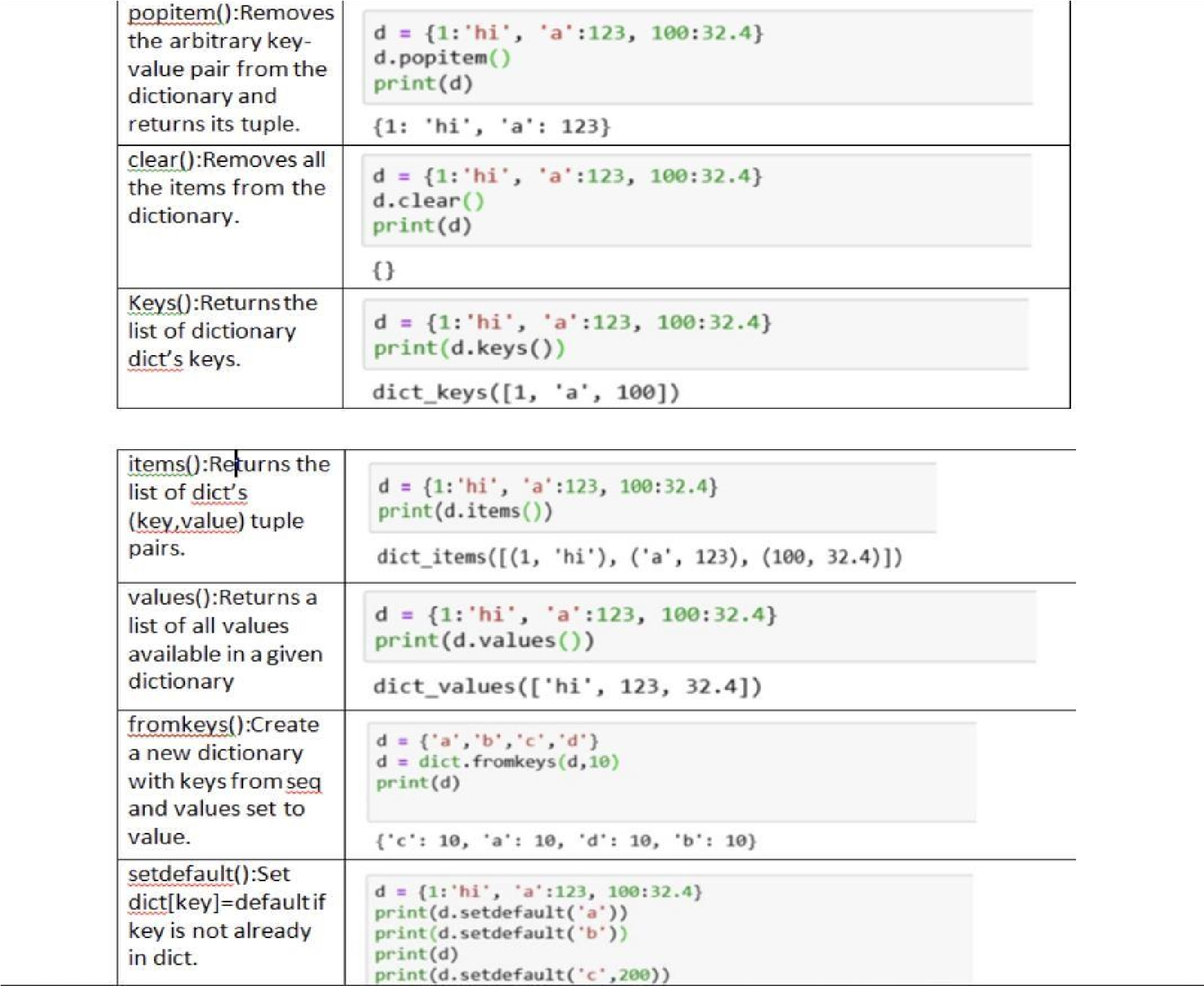
a=10 b=10

if a==b: print("Inside If") print("Outside If") The condition is a==b.

If a value, 10 is equal to b value 10 then the statement inside if statement, print("Inside If") will be executed and then the next statement will be executed, Hence the output is: Inside If Outside If >>>







**IMPORTANCE OF LISTS:**

They are many inbuilt functions and methods of Lists which helps to solve any problem statements with ease and less lines of code.

Many organizations like Cognizant, Microsoft, IBM, Accenture, Infosys Infytq ask questions on Lists.

LET'S WARM UP:

List is an ordered collection of iterns. Items can be of any data type like number, float, string. tuple, dictionary, other list. 10 Python 42.31 True (1,2,3)

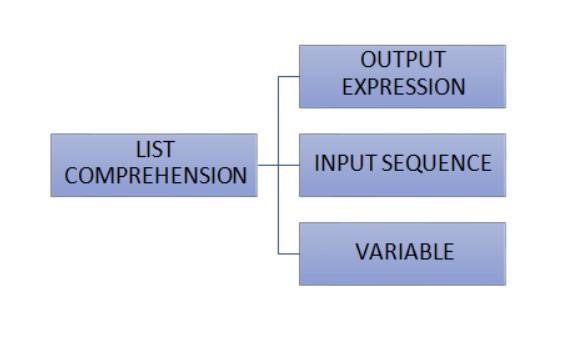
Creating a List:

List can be created using square brackets, [].

Items inside a list can be separated using commas.

>>> L [] Empty list

>>> print (L)

>>> print (type (L)) <class 'list'> 

### 4.NUMPY

OPERATIONS ON ARRAY - 1:

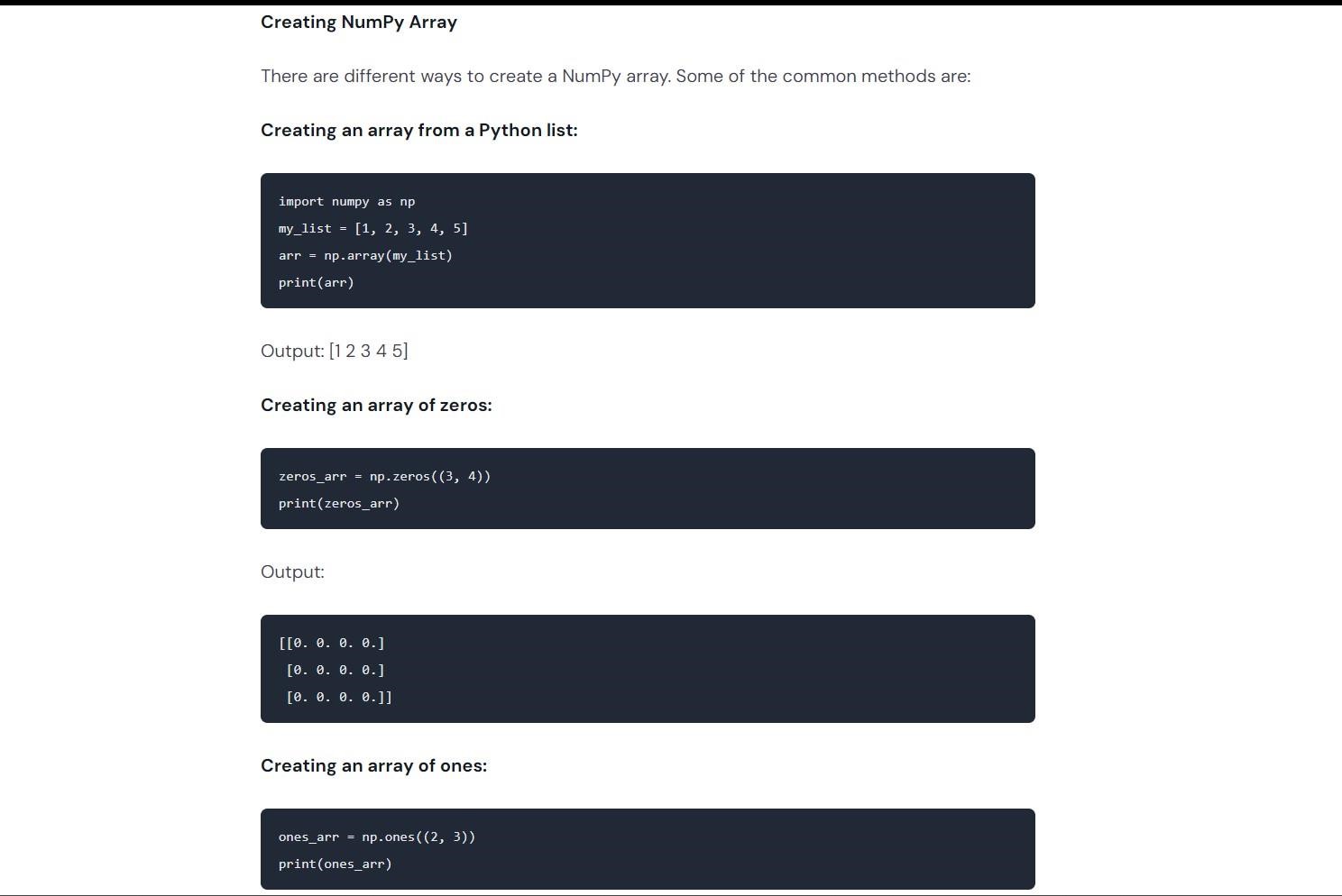
Creating an array: NumPy allows you to create arrays of a specified shape and data

type using the np. array() function.

Indexing and Slicing: You can access individual elements of an array by indexing and multiple elements of an array using slicing.

Reshaping: You can change the shape of an array using the np. reshape() function. Concatenation: Using the np. concatenate() function, you can concatenate two or more arrays.

Splitting: Using the np, you can split an array into two or more smaller arrays. split() function.



**Reshaping:**

You can reshape an array into a different shape using the np.reshape() function. Here are two examples:

Example 1: Reshaping a one-dimensional array into a two-dimensional array import numpy as np

#Create a one-dimensional array arr

np.array([1, 2, 3, 4, 5, 6])

#Reshape the array into a two-dimensional array

new\_arr = np.reshape(arr, (2, 3))

print(new\_arr) Output: [[123]

[456]]

Example 2: Reshaping a two-dimensional array into a one-dimensional array import numpy as np

#Create a two-dimensional array arr np.array([[1, 2, 3], [4, 5, 611)

#Reshape the array into a one-dimensional array

new\_arr np.reshape(arr, (,)) print(new\_arr) Output:

[123456]

**Concatenation:**

Concatenation refers to combining two or more arrays into a single array. NumPy provides the np.concatenate() function to concatenate arrays. Here are two examples:

Example 1: import numpy as np

#Create two arrays of same shape \*arr1= np.array([[1, 2, 3], [4, 5, 6]]) arr2 np.array([[7, 8, 9], [10, 11, 12]])\*

Concatenate along rows result np.concatenate([arr1, arr2], axis=0) print(result1) #Concatenate along columns result2 np.concatenate([arr1, arr2], axis=1) print(result2)

Output: [[123] [456] [789] [10 11 12]]

[[123789]

[4 5 6 10 11 12]]

#Concatenate along columns result2 = np.concatenate([arr1, arr2], axis=1) print(result2)

OPERATIONS ON ARRAY - 2:

Transposing:: You can ranspose a multi- dimensional array using the np.transpose() function.

Mathematical Operations: NumPy provides a wide range of mathematical functions that can be applied to arrays, such as addition, subtraction, multiplication, division, etc.

Broadcasting: NumPy allows for broadcasting, which means that you can perform operations on arrays of different shapes and sizes, as long as they are compatible. Reduction Operations: NumPy provides several reduction operations, such as np.sum(), np.mean(), np.max(), np.min(), etc., that allow you to compute statistics on an array.

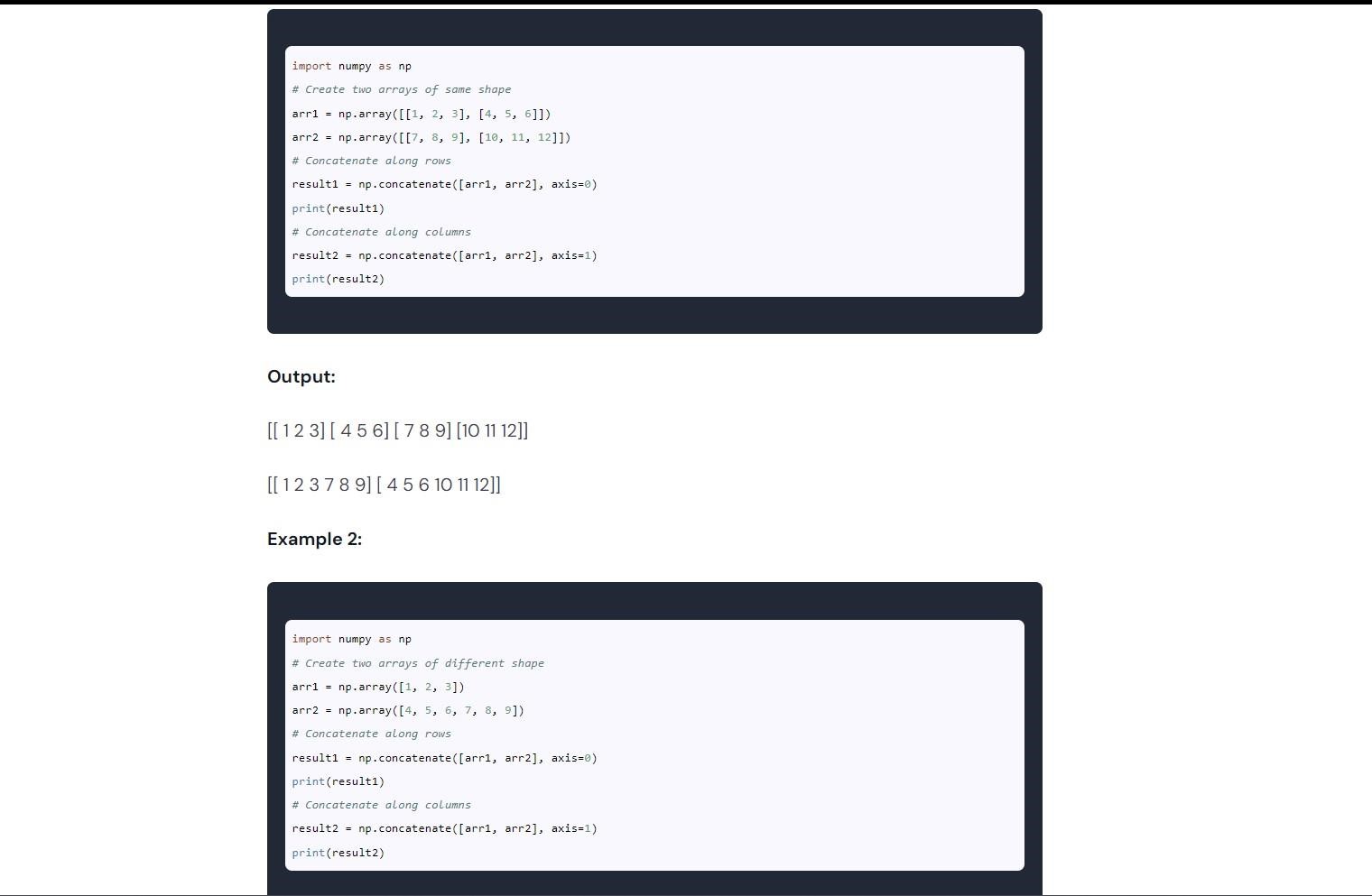
Boolean Operations: NumPy provides several boolean operations, such as np.logical\_and(), np.logical\_or(), np.logical\_not(), etc., that allow you to perform logical operations on arrays.

Transposing:

Transposing is the process of swapping the rows and columns of a matrix or array. In NumPy, you can transpose an array using the np.transpose() function. Here are two examples:

Example1: Transposing a 2D array import numpy as np #create a 20 array np.array([[1, 2, 3], [4, 5,

6]])

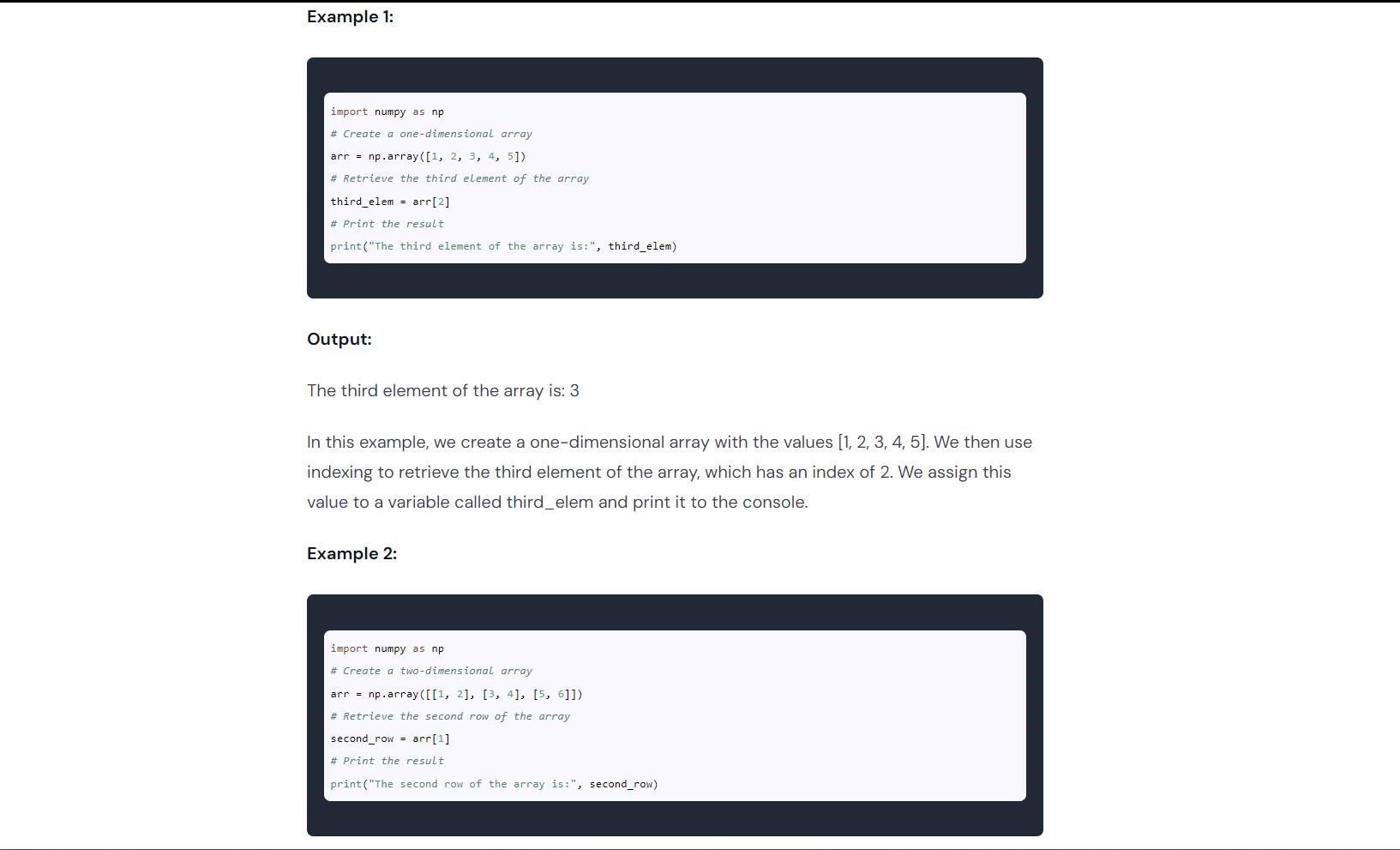


**INDEXING & SLICING**

**Indexing:**

Indexing in NumPy is used to retrieve specific elements from an array. It can be used to select individual elements or a subset of elements based on their position in the array.

Here are there two examples:



### 5.TABLEAU

Tabulae (plural of Tabula) is a Latin word that means "tables," and in the context of data analysis, it refers to a collection of information in a structured format. Tabulae are often used in data visualization and analysis to provide insights into large datasets.

Metadata is data that describes other data. It provides information about the structure, content, and context of a dataset. Metadata is essential for understanding and working with tabulae because it provides the necessary information to analyze and interpret the data accurately.

**Types of Tableau Charts:**

There are several types of tabulae charts that are commonly used in data visualization. Some of these include:

Bar Chart: A bar chart is a chart that uses rectangular bars to represent data. The height or length of each bar corresponds to the value of the data it represents.

Line Chart: A line chart is a chart that uses lines to connect data points: it is often used to show trends over time.

Pie Chart: A pie chart is a chart that uses slices of a circle to represent data. The size of each slice corresponds to the value of the data it represents

Scatter Chart: A scatter chart is a chart that uses dots to represent data. It is often used to show the relationship between two variables.

Heatmap Chart: A heatmap chart is a chart that uses color to represent data. It is often used to show patterns in large datasets.

Visual Analyst:

The visual analyst is the process of analyzing data using visual tools and techniques. It involves using charts, graphs, and other visualizations to understand and interpret data. Visual analysis is essential for understanding large datasets because it allows analysts to see patterns and trends that may not be apparent in raw data.

Visual analysis often involves using software tools to create and manipulate visualizations. Some popular tools for visual analysis include Tableau, Power BIL, and Google Data Studio. These tools provide users with a wide range of chart types and customization options, making it easier to create compelling and informative visualizations.

In conclusion, tabulae and metadata are essential components of data analysis, and using visual tools like tabulae charts and visual analysis can help analysts make sense of large datasets. By understanding these concepts and using the right tools, analysts can gain valuable insights and make better-informed decisions, here are some other concepts related to tabulae:

Data Cleaning: Before creating a tabula or using it for analysis, it is essential to ensure that the data is clean and accurate. Data cleaning involves removing or correcting any errors.

inconsistencies, or duplicates in the dataset

Data Aggregation: Data aggregation involves combining multiple data points into a single value. Aggregation is often used when working with large datasets to reduce the complexity of the data and make it easier to analyze.

Data Visualization: Data visualization is the process of using charts, graphs, and other visualizations to represent data. Visualization is an essential component of tabulae because it allows analysts to see patterns and relationships in the data.

Data Mining: Data mining is the process of extracting patterns and insights from large datasets. Data mining techniques are often used in conjunction with tabulae to uncover hidden relationships and trends in the data.

Machine Learning: Machine learning is a subset of artificial intelligence that involves training algorithms to identify patterns and make predictions based on data. Machine learning techniques are often used in conjunction with tabulae to analyze and predict trends in the data. By understanding these additional concepts related to tabulae, analysts can gain a more comprehensive understanding of the data and extract valuable insights from it.

**META DATA**

Metadata, also known as metadata, plays a crucial role in Tableau. It provides additional context and information about the data being used in a Tableau workbook. Understanding and utilizing metadata can enhance data analysis and visualization capabilities. Here are some key aspects of metadata in Tableau along with relevant examples:

Field Metadata:

Data Type: Tableau assigns appropriate data types to fields, such as String, Number, Date, Boolean, etc. This ensures proper handling and formatting of data during analysis. For example, a field called "Order Date might be assigned a Date data type.

Aggregation: Tableau applies default aggregation methods (eg, Sum, Average, count) to numeric fields. However, you can modify the aggregation settings based on your analysis requirements. For instance, a field named "Sales" could be aggregated as the sum of individual sales amounts.

Column Metadata:

Column Names: Tableau uses column names to identify and reference data fields Descriptive column names make it easier to understand and work with the data. For instance, a column named

"Product Category provides clarity on the type of products being analyzed

Column Properties: Tableau allows you to customize column properties, such as hiding or showing specific columns, sorting order, and formatting options (eg. currency percentage) These properties enhance the visual presentation and analysis of data.

Table Metadata:

Joins and Relationships: In Tableau, you can connect multiple tables using joins or establish relationships based on common fields. This metadata defines how data from different tables. is related and allows you to perform cross-table analysis. For example, joining an "Orders table with a "Customers table using a common field like "Customer ID' enables analyzing sales by customer. Hierarchies: Tableau supports creating hierarchies by grouping fields together in a hierarchical structure. This metadata enables drilling down and summarizing data at different levels of granularity. For instance, a hierarchy can be created with "Year," "Quarter," and "Month fields to analyze sales trends at various time levels.

Calculation Metadata:

Calculated Fields: Tableau allows users to create custom calculations using formula

expressions. These calculations can be based on existing fields, constants, or functions. Calculated

fields provide additional insights and flexibility in data analysis. For example, a calculated field called "Profit Margin can be created by subtracting the cost from the revenue and dividing it by the revenue.

Table Calculations: Tableau supports table calculations that perform computations on the resulting visualization. These calculations utilize the metadata of the visualization, such as the order of dimensions and measures, to generate dynamic results. For instance, a table calculation could calculate the running total of sales over different dimensions like "Region" or "Product Category" These examples illustrate how metadata in Tableau contributes to data understanding analysis, and visualization. Utilizing metadata effectively enhances the overall user experience and empowers data-driven decision-making.

Geographic Metadata:

Geocoding: Tableau has built-in geocoding capabilities that can map geographic data based on specific fields, such as addresses or coordinates. This metadata helps visualize and analyze data on maps. For example, mapping customer addresses to display their distribution across regions. Geographic Roles: Tableau allows assigning geographic roles to fields, such as Country, State/Province, City, etc. This metadata helps Tableau recognize the data as geographic and enables advanced geographic analysis. For instance, aggregating sales by state or visualizing data at different levels of geographic granularity

Parameter Metadata:

Parameters: Tableau parameters allow users to create dynamic inputs that can be adjusted to explore different scenarios or filter data. Metadata associated with parameters includes data type, allowable values, and default settings. For example, a parameter named "Sales Threshold could be used to filter data based on a minimum sales value set by the user.

Data Source Metadata:

Connection Metadata: Tableau maintains metadata about the data connection, including the type of database or file, connection settings, and credentials. This metadata ensures secure and efficient data retrieval from the data source

Extract Metadata: Tableau extracts data from the data source into a specialized format optimized for performance. Extract metadata includes details about the extract, such as refresh schedules, filters applied and calculated fields created. This metadata supports faster data access and efficient storage.

Workbook Metadata:

Captions and Annotations: Tableau allows adding captions and annotations to worksheets, dashboards, and other workbook elements. These metadata elements provide contextual information, explanations, or instructions to enhance the understanding of the analysis. For instance, adding a caption to explain the purpose of a specific visualization.

Workbook-level Filters: Tableau enables applying filters at the workbook level, which affects all worksheets and dashboards within the workbook. This metadata allows controlling the data displayed across the entire workbook based on specific criteria.

Remember that metadata in Tableau serves as a foundational layer of information that facilitates data exploration, analysis, and visualization. By leveraging and managing metadata effectively, you can gain deeper insights from your data and create impactful visualizations in Tableau.

### 6.POWER BI

Power BI is a business intelligence tool developed by Microsoft. It allows you to create interactive visualizations and reports from various data sources, including Excel spreadsheets, cloud-based and on-premise databases, and other online services. Power BI provides users with the ability to transform, analyze and visualize data with ease, making it an ideal solution for businesses of all sizes.

Let's explore some scenarios where Power BI can be useful.

Scenario 1: Sales Analysis:

Imagine you are the sales manager for a company that sells different products across different regions. You want to track the sales performance of your team to identify the areas where you can improve. By using Power Bl, you can create a dashboard that shows sales performance by product, region, and salesperson. This dashboard can be updated in real- time, allowing you to keep track of your team's progress throughout the day.

Scenario 2: Customer Behavior Analysis:

Imagine you are the marketing manager for an online retailer. You want to analyze customer behavior to determine what products are most popular and which marketing campaigns are most effective. By using Power Bl, you can create a dashboard that shows customer buying behavior, such as the number of products purchased, time spent on the site, and conversion rates. This dashboard can be used to make informed decisions about marketing campaigns, product development, and customer engagement.

Scenario 3: Financial Analysis

Imagine you are the CFO of a large corporation. You want to analyze financial data to determine the company's profitability and identify areas where costs can be reduced. By using Power BI, you can create a dashboard that shows financial metrics, such as revenue, expenses, and profit margins. This dashboard can be used to track financial performance over time and identify trends that may indicate areas for improvement.

Power BI offers several features that make it an ideal tool for data visualization and analysis.

Some of these features include:

Interactive Visualizations - Users can create interactive charts, graphs, and other visualizations that allow them to explore data in real-time.

Data Modeling - Users can transform and shape data from multiple sources into a format that is easy to analyze.

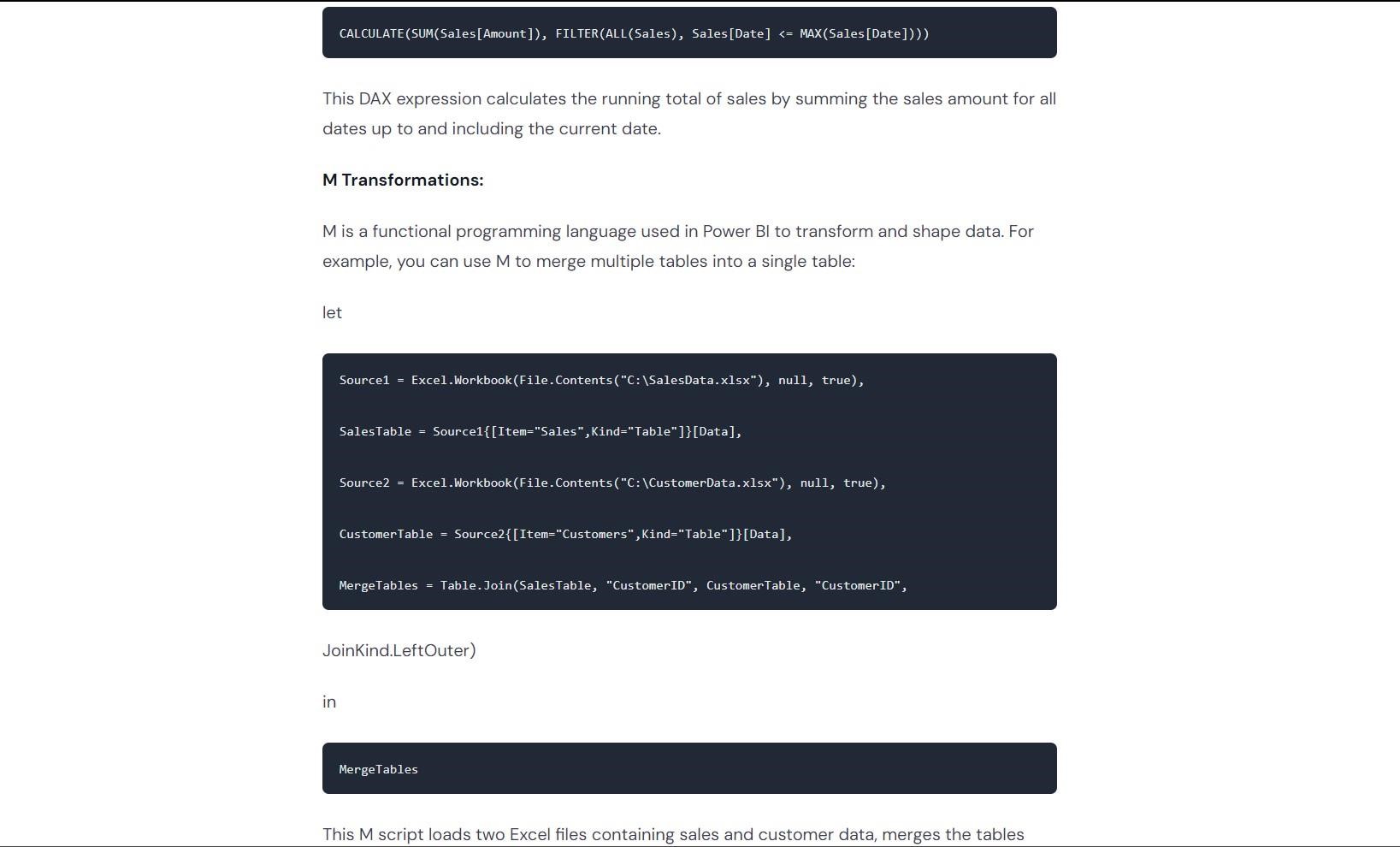
Customizable Dashboards - Users can create customized dashboards that show key metrics and data trends in a clear and concise format.

Collaboration - Users can share their dashboards and reports with others, making it easy to collaborate and make informed decisions. In summary, Power Bl is a powerful business intelligence tool that allows users to analyze and visualize data from multiple sources. Whether you are a sales manager, marketing manager, or CFO, Power BI can help you make informed decisions and improve your business operations.

Certainly! Power BI supports several programming languages, including DAX, M, and R, which can be used to execute complex calculations, create custom functions, and perform advanced data analysis. Here are a few examples of coding executions in Power Bl:

**DAX Calculations:**

**DAX (Data Analysis Expressions) is a formula language used in Power Bl to create custom calculations and measures. For example, you can use**



R Visualizations:

R is a statistical programming language used in Power BI to create custom visualizations and perform advanced data analysis. For example, you can use R to create a heatmap of customer demographics: library(ggplot2) library(dplyr)

customerData <- read.csv("CustomerData.csv")

heatmapData <- customerData %>% select(Gender, AgeGroup, IncomeGroup) %>% group\_by(Gender,

AgeGroup, Income Group) %>% summarize(count

= n()) %>% ungroup() ggplot(heatmapData, aes (x=IncomeGroup, y=AgeGroup, fill=count)) + geom\_tile() +

scale\_fill\_gradient(low="white", high="red") + facet

wrap(-Gender) +

Interface and Data Connection in Power BI

Power BI is a powerful business analytics tool developed by Microsoft that allows users to easily connect to and analyze data from various sources. With Power Bl, you can create visually stunning and interactive reports, dashboards, and data visualizations that enable you to make datadriven decisions.

As for your question on Content Interface and Data Connection in Power Bl, let me explain it in simpler terms.

Here are a few examples of Content Interface in Power BI:

Table Visualization: The table visualization is one of the simplest ways to display data in Power Bl. It displays data in rows and columns and allows you to sort and filter the data easily.

For example, you could use a table visualization to display a list of sales by product or customer.

Column Chart: A column chart is a visualization that displays data in vertical columns. It is useful for comparing data across categories, such as sales by product category or revenue by region. Line Chart: A line chart is a visualization that displays data as a series of points connected by a line. It is useful for displaying trends over time, such as sales growth or website traffic.

Map Visualization: A map visualization displays data on a map, making it easy to visualize data geographically. For example, you could use a map visualization to display sales by region or customer location.

Gauge Visualization: A gauge visualization is a useful way to display data as a percentage of a whole. For example, you could use a gauge visualization to display progress towards a sales target or customer satisfaction level.

KPI Visualizations: Key Performance Indicator (KPI) visualizations are useful for displaying important metrics at a glance. For example, you could use a KPI visualization to display the current month's sales or the number of new customers acquired this month.

Data Connection in Power BI refers to the process of connecting to one or more data sources and importing data into Power BI for analysis. Power BI supports a wide range of data sources, including Excel spreadsheets, databases such as SQL Server. Oracle, and

Power BI also provides a wide range of connectivity options to help you connect to your data sources quickly and easily.

Once you have connected to your data source, you can transform and shape your data using Power Bl's data modeling and query tools. This allows you to clean and transform your data so that it is ready for analysis.

Here are some examples of data connections that you can make in Power BI:

Excel Files: Power BI allows you to connect to Excel files stored on your local computer or in the cloud (such as OneDrive or SharePoint Online). You can import data from specific worksheets or ranges, and then use Power Query to transform the data as needed.

**PROJECT**

**on**

**SALARY PREDICTION USING POLYNOMIAL REGRESSION**

**Salary prediction using Polynomial Regression is a data analysis technique that helps us forecast salaries based on historical data. In simple terms, it's like fitting a curve to our salary data points, allowing us to make predictions about future salaries.**

**STEPS FOR SALARY PREDICTION USING POLYNOMIAL REGRESSION**

**Step 1:**

Importing Necessary Libraries We start by importing essential libraries in Python, such as NumPy, pandas, and scikit-learn. These libraries provide tools for numerical operations, data manipulation, and machine learning.

**Step 2:**

Loading and Exploring Data Load your salary data into a pandas DataFrame and explore it to understand the structure of the dataset.

**Step 3:**

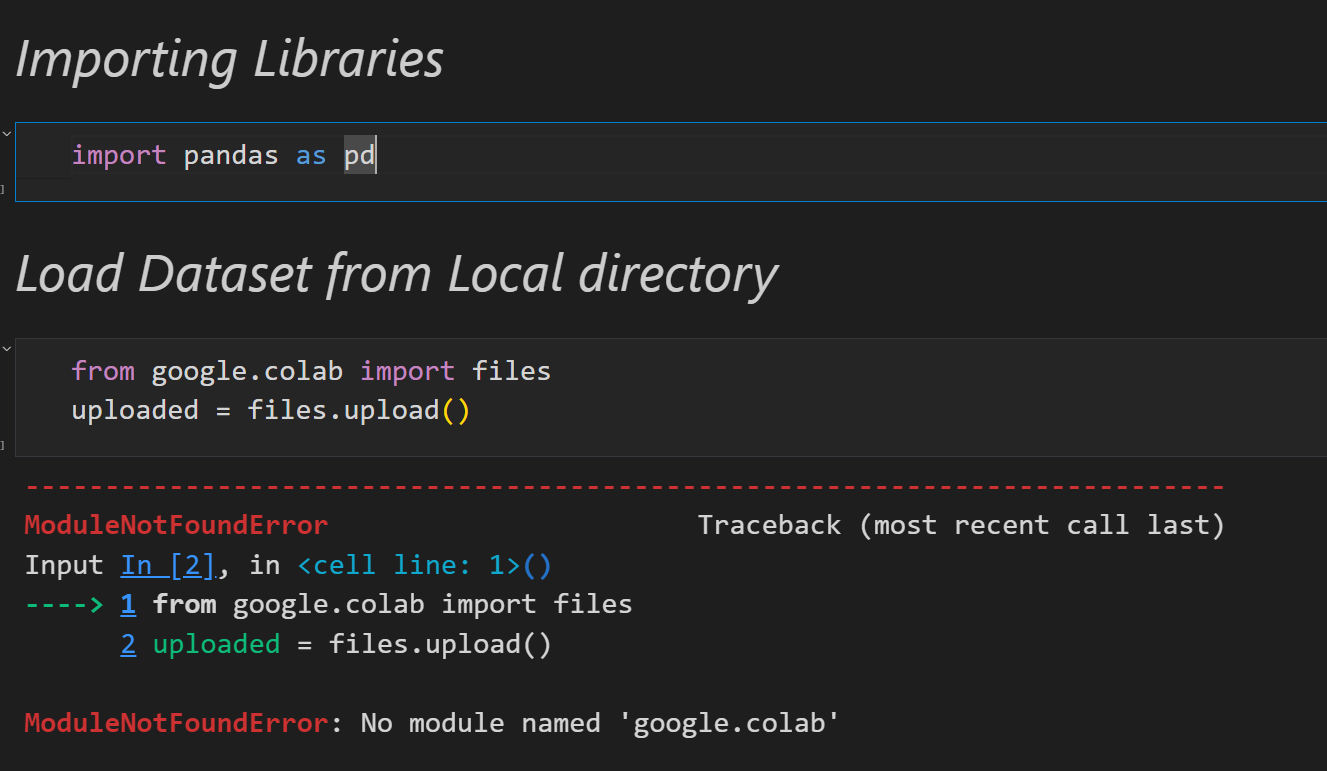
performance Splitting Data into Training and Testing Sets Divide your data into training and testing sets. The training set is used to train the model, and the testing set is used to evaluate its

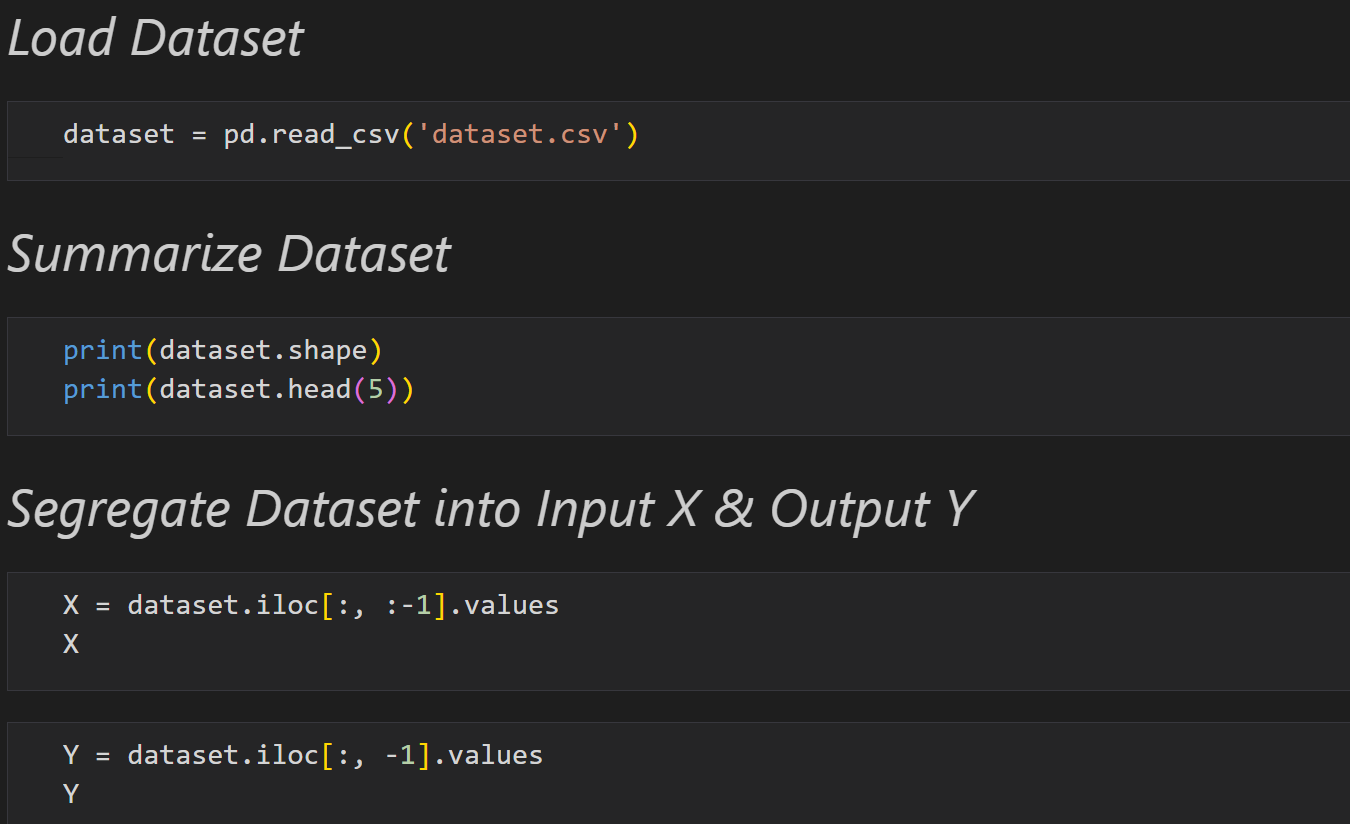
**Step 4:**

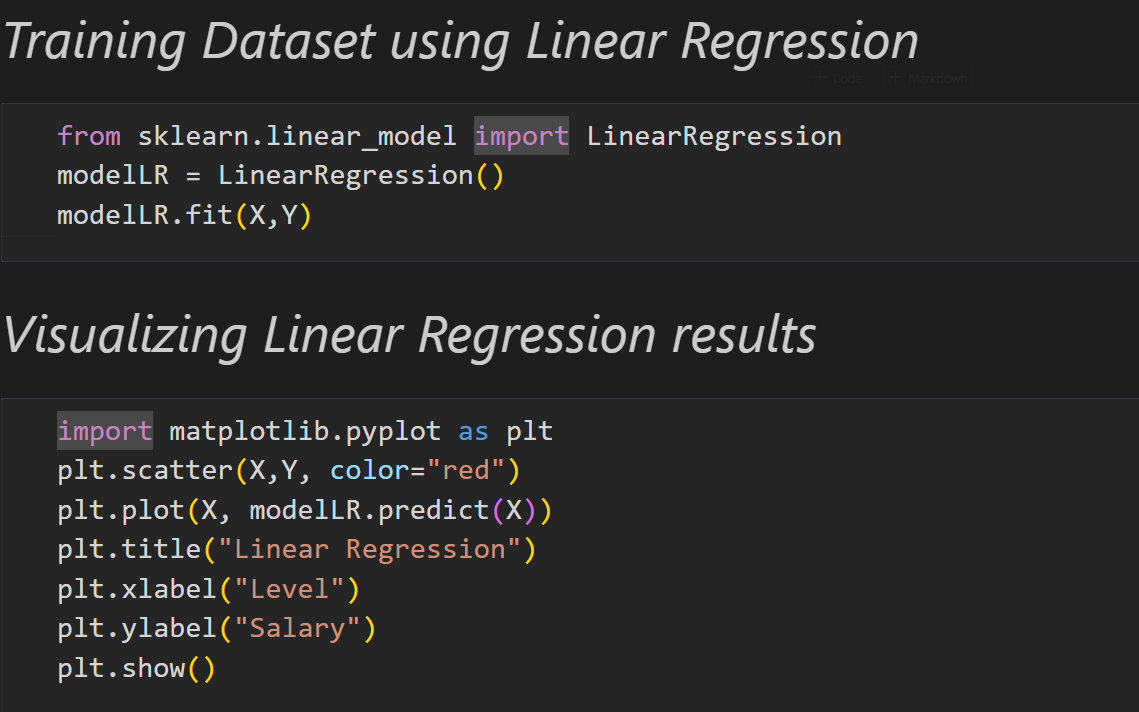
Polynomial Regression Model Apply Polynomial Regression to fit a curve to the training data. This involves transforming the features into polynomial features and using a linear regression model.

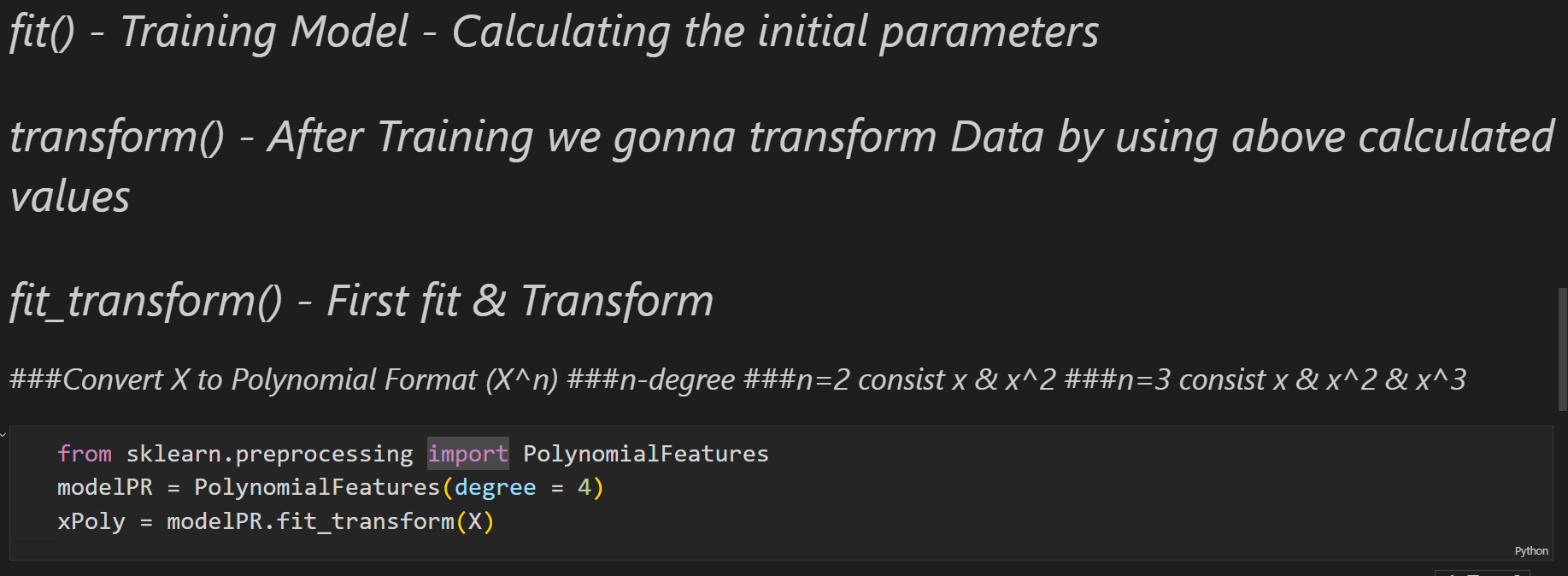
**Step 5:**

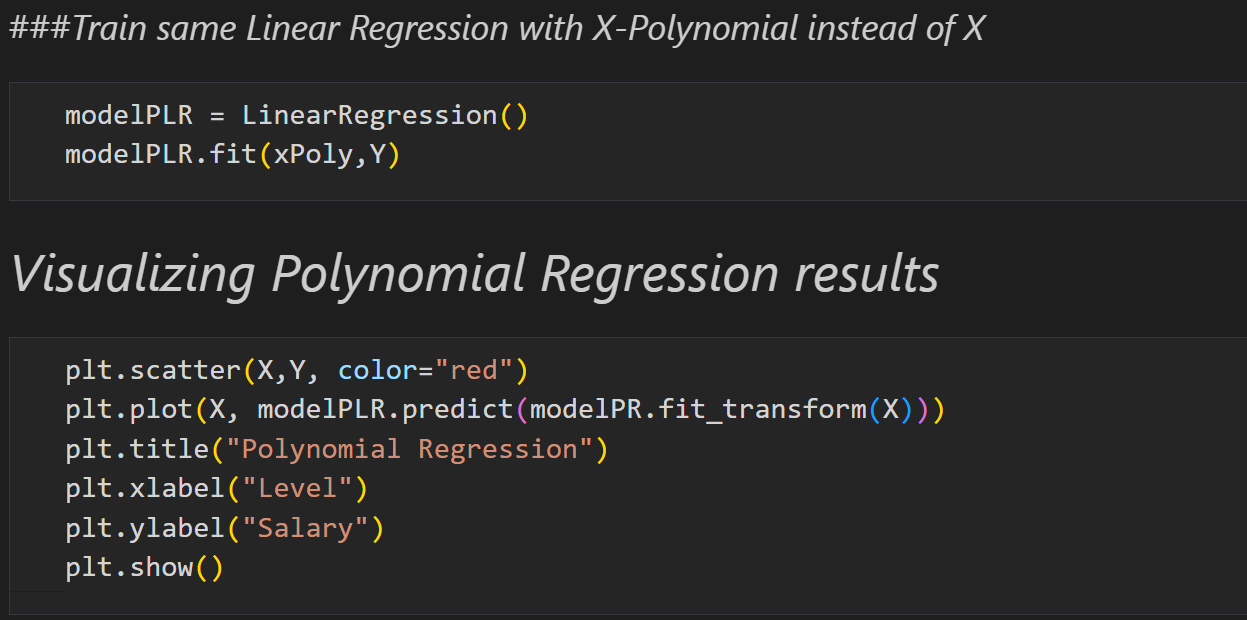
Predictions and Evaluation Make predictions using the trained model and evaluate its performance on the testing set.

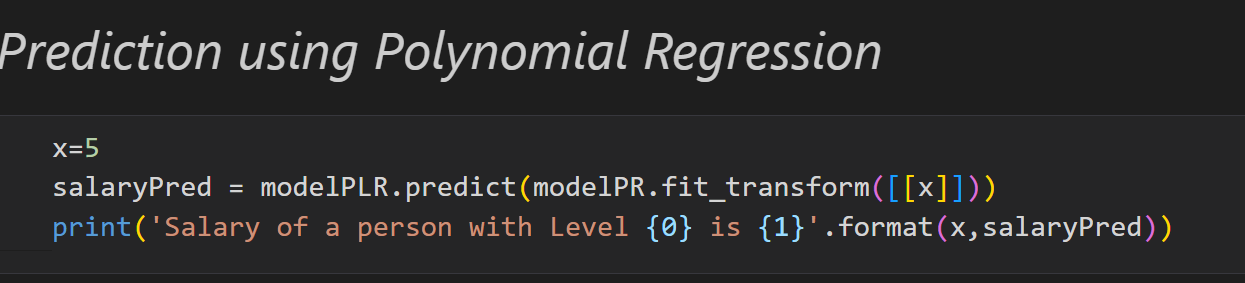
****



****

****

****

****

**CONCLUSION** :

I believe the trial has shown conclusively that it isboth possi ble and desirable to use Data Science as the principal teaching language. It is Free (as in both cost and source code).

* It can be used to teach a large number of transferable skills. It is a real-world Technology that can be and is used in academia and the commercial world.
* It appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied and most importantly, its clean syntax offers increased understanding and enjoyment for students.
* The training program having three destination was a lot more useful than staying at one place throughout the whole 5 months. In my opinion. I have gained lots of knowledge and experience needed to be successful in great engineering challenge as in my opinion, Engineering is after all a Challenge, and not a job.

**NITESH KUMAR YADAV**

**21781A05K1**