

# **Minor Project Report**

## **on**

# **Diwali sales Analysis**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE  
DIPLOMA IN COMPUTER ENGINEERING



**Guru Nanak Dev Polytechnic College, Gill Park (Ludhiana)**

**SUBMITTED TO:**

Dr. Hardeep Singh Jawanda, HOD Computer Engineering Department

**SUBMITTED BY:**

Abhishek Kumar Mahato (1201)

210179501272

**Batch: 2021-2024**

# Index

## Contents:

1. Candidate's Declaration.....	3
2. Acknowledgement.....	4
3. Preface.....	5
4. Abstract.....	6
5. Existing System Study.....	7
6. Requirements.....	8
7. Development Software's and Languages.....	9
a. PYTHON.....	9-14
b. Data Analysis.....	15-17
c. Pandas.....	18-19
d. Numpy.....	20-21
e. Matplotlib.....	22-23
f. Seaborn.....	24-25
g. Jupyter Notebook.....	26-28
8. Project Introduction.....	29-31
9. Implementation.....	32-36
10. Bibliography.....	37

### Candidate's Declaration

I am the student of **C.S.E (Computer Science Engineering) department at Guru Nanak Dev Polytechnic College Ludhiana**, hereby declare that we own full responsibility for the information, results, conclusions, etc. provided in this project report titled **"Diwali sales Data Analysis "**, submitted to **Guru Nanak Dev Polytechnic College, Ludhiana, INDIA** for the award of **C.S.E (Computer Science Engineering) diploma**. We have completely taken care in acknowledging the contribution of others in this academic work, We further declare that in case of any violation of intellectual property rights or copyrights found at any stage. We candidates will be solely responsible for the same.

---

**Abhishek Kumar Mahato**

**Certified that this project report is the benefice work of Abhishek Kumar Mahato**

---

**Internal Examiner**

---

**External Examiner**

## Acknowledgement

We express our gratitude to the **Principal Mrs. Sarbjit Kaur** for having provided me facilities to do the project successfully. We would like to express our special thanks of gratitude to our teachers **HOD Hardeep Singh Jawanda and Mrs. Prabhjot Kaur Mam** who gave us golden opportunity to do this project on (**Diwali Data sales Analysis**) , which also helped us in doing a lot of research and we came to know about many new things. We really thankful to them. Secondly, we are thankful to our parents and friends who helped me lot in finishing this project within limited time. We not make this project only for marks but also to increase knowledge. We have taken efforts in this project. However, it would not be possible without the help and kind support of many individuals and organizations and we would like to extend our sincere thanks to all of them.

We thank our God for providing us with everything that we required in completing our project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which help me in the completion of this project.

Our thanks and appreciation also to our classmates in developing the project and to the people who have willingly help us out with their abilities.

## Preface

This report reprints our opportunity of being in a professional environment as a student at **Guru Nanak Dev Polytechnic College, Ludhiana** during our minor project for the award of **Diploma in Computer Science and Engineering**. A Sincere effort has been made such that this report gives a clear picture of the interactive platform to provide basics of **PYTHON, HTML and CSS**.

Experience, no doubt, is a great teacher and a valuable asset. However, the supreme importance of sound working knowledge of principles should never be overlooked, as the mere experience cannot go a long way. The purpose of project development is to familiarize the students with the present working technology in the country and outside.

The project development provide student with an opportunity to study the latest technical trends those have been established as well as the ones that are developing. Moreover, to make students feel sure that whatever they are learning during this 5th semester would certainly help them for the future course of action and help them to develop their potential and technical skills.

### Abstract

Project is a series of activities taken by a programmer to practically apply and implement the knowledge gained during course of study. In general, one can say that a project is not a physical objective, nor is the end result.

It has something to do with the going in between, which must be same, whether we build a high technology process plant or merely hold on election, it is termed as a project. This goal or requirement of the project decides the path and mode wise that a project is to be executed.

Basically, the project is all about the **“Diwali sales DATA ANALYSIS”**. In this project we provide basics of Data Analysis concepts, and various types of libraries which are used in python data analysis program.

#### Languages are:

- PYTHON
- HTML
- CSS

### Existing System Study

Firstly, we study many websites related to “**education-website**” and grab knowledge for the project.

- <https://www.w3schools.com/>
- <https://github.com/>
- <https://www.geeksforgeeks.org/>
- <https://www.youtube.com/>

All the websites are not building as perfectly all website doesn't all types to categories and the user interface not that good enough. And the design is not good.

All this issue I am facing will collect the information from different resources keep that thing in mind make my Data analysis project.

Many websites are not responsive based that create a problem to learn all information about the site on the mobiles.

Lack of information on the site, All the website are not build as perfectly all website doesn't all types to categories and the user interface not that good enough. And the design is not that good but it will helpful.

All this issue I am facing will collect the information from different resources. And keep that thing in mind make my Data analysis project.

## Requirements

### Hardware

- **Processor:** Intel(R) Core(TM) i3-5010U CPU @ 2.10GHz 2.10 GHz
- **RAM:** 4 GB
- **Hard disk:** up to 10 GB of available space may be required. However, 10 GB free space is required in boot drive even if you are installing in other drive.

### Software

The list given below of the software's are used for Analysis the Data of Diwali sales of company.

- VS code
- Jupyter notebook
- Chrome
- Microsoft office
- MS Excel
- A Good Internet Connection to use all the services of website.

### Operating System Used Windows

- Windows 7
- Windows 8, 8.1
- Windows 10
- Linux
- Unix
- Mac



## Development Software's and Languages

### What is python?

Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python is widely used for various purposes, including web development, data analysis, artificial intelligence, scientific computing, automation, and more.

### Here are some key aspects of Python:

#### 1. Readability and Simplicity:

Python's syntax is designed to be readable and straightforward, making it easy to write and understand code. This readability reduces the cost of program maintenance and development.

#### 2. Interpreted Language:

Python is an interpreted language, which means that you can run Python code line by line without the need for compilation. This makes the development process faster and more interactive.

#### 3. Dynamically Typed:

Python is dynamically typed, meaning you don't need to declare the data type of a variable explicitly. Python interpreter automatically determines the data type during runtime, making it flexible and easy to use.

#### 4. Object-Oriented and Procedural:

Python supports both object-oriented and procedural programming paradigms. It allows developers to write modular and reusable code using classes and objects.

#### 5. Extensive Standard Library:

Python comes with a vast standard library that includes modules and packages for various tasks, such as file I/O, regular expressions, networking, databases, and more. These libraries simplify complex programming tasks and boost productivity.

#### 6. Community Support:

Python has a large and active community of developers who contribute to open-source projects, create libraries, and provide support through forums and online communities. This vibrant community ensures a wealth of resources and assistance for Python developers.

### 7. Cross-Platform Compatibility:

Python is available for different operating systems, including Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code that can run on multiple platforms without modification.

### 8. Versatility:

Python is versatile and can be used for a wide range of applications, from web development (using frameworks like Django and Flask) to scientific computing (using libraries like NumPy and SciPy) and machine learning (using libraries like TensorFlow and scikit-learn).

### 9. Easy Integration:

Python can be easily integrated with other languages like C, C++, and Java. This integration capability allows developers to use existing code and libraries from other languages within Python programs.

Python's simplicity, versatility, and strong community support make it an excellent choice for beginners and experienced developers alike. Its wide range of applications and ease of use have contributed to its popularity in the software development industry.

**Certainly! Here are a few simple Python code examples that demonstrate basic programming concepts:**

#### 1. Hello, World!

```
print("Hello, World!")
```

This is the simplest Python program that prints "Hello, World!" to the console.

#### 2. Variables and Basic Operations:

```
# Variables and basic operations
```

```
a = 5
```

```
b = 3
```

```
sum = a + b
```

```
product = a * b
```

```
print("Sum:", sum)    # Output: Sum: 8
```

```
print("Product:", product) # Output: Product: 15
```

This code snippet demonstrates variable declaration and basic arithmetic operations.

### 3. Conditional Statements:

```
# Conditional statements
```

```
num = 10
```

```
if num > 0:
```

```
    print("Positive number")
```

```
elif num == 0:
```

```
    print("Zero")
```

```
else:
```

```
    print("Negative number")
```

This code checks whether a number is positive, zero, or negative and prints the corresponding message.

### 4. Loops:

```
# Loops
```

```
for i in range(1, 6):
```

```
    print("Number:", i)
```

```
# Output: Number: 1
```

```
#    Number: 2
```

```
#    Number: 3
```

```
#    Number: 4
```

```
#    Number: 5
```

This loop prints numbers from 1 to 5 using a for loop

### 5. Lists:

# Lists

```
fruits = ["apple", "banana", "cherry"]
```

```
print(fruits[1]) # Output: banana
```

```
fruits.append("orange")
```

```
print(fruits) # Output: ["apple", "banana", "cherry", "orange"]
```

This code snippet demonstrates creating a list of fruits, accessing elements, and appending a new fruit to the list.

### 6. Functions:

# Functions

```
def greet(name):
```

```
    return "Hello, " + name + "!"
```

```
print(greet("Alice")) # Output: Hello, Alice!
```

This code defines a function called `greet` that takes a name as an argument and returns a greeting message.

These are basic examples to get you started with Python. As you progress, you can explore more advanced concepts like classes, objects, file handling, and modules to enhance your programming skills.

## DATA ANALYSIS

### What is Data Analysis?

Data analysis is the process of inspecting, cleaning, transforming, and modeling data to extract useful information, draw conclusions, and support decision-making. It involves various techniques and methods to explore patterns, relationships, and trends within datasets. Data analysis is crucial in various fields, including business, science, social research, and government.

**Here are the key components of data analysis:**

#### 1. Data Collection:

- Gathering relevant data from different sources, which can include surveys, experiments, sensors, online platforms, and more.

#### 2. Data Cleaning and Preprocessing:

- Cleaning data to handle missing values, outliers, and inconsistencies.
- Preprocessing involves tasks like normalization, transforming categorical data into numerical formats, and feature scaling.

#### 3. Exploratory Data Analysis (EDA):

- Analyzing data sets visually and statistically to discover patterns, identify relationships, and gain a deeper understanding of the data.
- EDA often involves creating charts, histograms, scatter plots, and other visualizations.

#### 4. Data Modeling:

- Applying various statistical and machine learning models to the prepared data.
- Models can include regression, classification, clustering, and more, depending on the nature of the analysis and the questions being asked.

#### 5. Data Interpretation:

- Interpreting the results obtained from the models to make informed decisions or draw conclusions.
- This step involves understanding the implications of the analysis and its significance in the context of the problem being studied.

### 6. Data Visualization:

- Presenting the findings and insights using visual elements like charts, graphs, and dashboards.
- Visualization enhances the understanding of complex data patterns and trends, making it easier for non-technical stakeholders to comprehend the results.

### 7. Decision Making and Reporting:

- Utilizing the analyzed data to make decisions, formulate strategies, or address specific research questions.

***Creating comprehensive reports or presentations summarizing the analysis process, results, and recommendations.***

### Importance of Data Analysis:

- **Informed Decision-Making:** Helps businesses and organizations make data-driven decisions, leading to more informed and strategic choices.
- **Identifying Patterns:** Allows the identification of trends, patterns, and correlations within datasets, which can be valuable for predictive analysis.
- **Problem Solving:** Assists in understanding complex problems by breaking them down into manageable components, enabling effective problem-solving strategies.
- **Optimizing Performance:** Businesses can optimize their processes, marketing strategies, and products/services based on insights derived from data analysis.
- **Research and Exploration:** Essential in scientific research for analyzing experimental data, testing hypotheses, and exploring new areas of study.

Data analysis is a multifaceted process that requires a combination of domain knowledge, statistical expertise, and technical skills to extract meaningful insights and value from data. Data analysis is a critical aspect of various fields, including business, science, healthcare, finance, and social sciences, among others. It empowers individuals and organizations to extract actionable insights from data, enabling better understanding and decision-making.

## DATA ANALYSIS IN PYTHON

In Python, data analysis refers to the process of inspecting, cleaning, transforming, and modeling data using various libraries and tools available in the Python programming language. Python has become one of the most popular programming languages for data analysis due to its simplicity, readability, and a rich ecosystem of libraries specifically designed for data manipulation, analysis, and visualization.

Here are some key libraries used for data analysis in Python:

### 1. NumPy:

- **Purpose:** NumPy provides support for arrays, which are essential for numerical computing.
- **Functionality:** It offers powerful numerical operations and mathematical functions, enabling efficient handling of large datasets.

### 2. Pandas:

- **Purpose:** Pandas is a data manipulation and analysis library.
- **Functionality:** It provides data structures like DataFrame and Series, which allow easy handling and manipulation of structured data. Pandas is widely used for data cleaning, transformation, and exploration.

### 3. Matplotlib:

- **Purpose:** Matplotlib is a 2D plotting library.
- **Functionality:** It allows you to create a wide variety of static, interactive, and animated plots and visualizations, making it useful for displaying data analysis results.

### 4. Seaborn:

- **Purpose:** Seaborn is a statistical data visualization library based on Matplotlib.
- **Functionality:** It provides a high-level interface for drawing attractive and informative statistical graphics, making it easier to create visually appealing plots.

### 5. SciPy:

- **Purpose:** SciPy is a library used for scientific and technical computing.
- **Functionality:** It builds on NumPy and provides additional functionality for optimization, integration, interpolation, eigenvalue problems, and more.

### 6. Scikit-Learn:

- **Purpose:** Scikit-Learn is a machine learning library.
- **Functionality:** It offers simple and efficient tools for data mining and data analysis, including various machine learning algorithms for classification, regression, clustering, and dimensionality reduction.

### 7. StatsModels:

- **Purpose:** StatsModels is a library for estimating and interpreting models for many different statistical techniques.
- **Functionality:** It provides classes and functions for statistical estimation, hypothesis testing, and statistical exploration.

### 8. Jupyter Notebooks:

- **Purpose:** Jupyter Notebooks are interactive computing environments.
- **Functionality:** They allow you to create and share documents that contain live code, equations, visualizations, and narrative text. Jupyter Notebooks are widely used for data analysis and exploration tasks

## Example Workflow for Data Analysis in Python:

### 1. Import Libraries:

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns
```

### 2. Load Data:

```
data = pd.read_csv('data.csv')
```

Data Exploration:

```
print(data.head()) # Display the first few rows of the dataset  
print(data.describe()) # Summary statistics
```

### 3. Data Cleaning:

```
data.dropna(inplace=True) # Remove rows with missing values
```



### 4. Data Visualization:

```
sns.scatterplot(x='column1', y='column2', data=data)
plt.title('Scatter Plot')
plt.show()
```

### 5. Data Analysis:

```
average_value = data['column3'].mean() # Calculate the mean of a specific column
correlation_matrix = data.corr() # Compute the correlation matrix
)
```

# Pandas

## What are pandas?

Pandas are a powerful open-source data analysis and data manipulation library for Python. It provides data structures like Series and DataFrame, which are designed to efficiently manipulate large datasets and perform complex data analysis tasks. A panda is widely used in the fields of data science, finance, research, and analytics due to its ease of use and flexibility.

## Key Features of Pandas:

### 1. Data Structures:

- **Series:** A one-dimensional array-like object that can hold any data type.
- **DataFrame:** A two-dimensional table with rows and columns, similar to a spreadsheet or SQL table.

### 2. Data Manipulation:

- **Data Cleaning:** Pandas provides functions to handle missing data, remove duplicates, and fill gaps in data.

### 3. Data Transformation:

- Perform operations like filtering, sorting, and aggregating data.
- Merging and Joining: Merge multiple DataFrames based on common columns or indices.

### 4. Data Input/Output:

- Pandas supports reading data from various file formats including CSV, Excel, SQL databases, and JSON.
- Data can be exported to different formats after processing.

### 5. Powerful Indexing:

- Pandas uses customizable row and column indices, allowing for easy and efficient data retrieval and manipulation.

### 6. Handling Time Series Data:

- Pandas has built-in support for time series data, making it easier to work with time-based data.

### 7. Data Visualization:

- Pandas integrates well with visualization libraries like Matplotlib and Seaborn, enabling the creation of meaningful plots and charts.

### 8. Integration with NumPy:

- Pandas is built on top of NumPy, allowing seamless integration between the two library.

#### Example Usage of Pandas:

import pandas as pd

#### # Creating a DataFrame from a dictionary

```
data = {'Name': ['Alice', 'Bob', 'Charlie'],  
        'Age': [25, 30, 35],  
        'City': ['New York', 'London', 'Paris']}  
df = pd.DataFrame(data)
```

#### # Reading data from a CSV file

```
# df = pd.read_csv('data.csv')
```

#### # Performing operations on the DataFrame

```
average_age = df['Age'].mean()  
filtered_data = df[df['Age'] > 30]
```

#### # Displaying the DataFrame

```
print(df)  
print("Average Age:", average_age)  
print("Filtered Data:")  
print(filtered_data)
```

In this example, Pandas is used to create a DataFrame from a dictionary, calculate the average age, and filter the data based on age. The resulting DataFrame is displayed and manipulated using Pandas functions.

## NUMPY

### What is numpy?

NumPy is a powerful and popular open-source library in Python for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a variety of mathematical functions to operate on these arrays. NumPy is a fundamental package for scientific computing with Python, enabling tasks related to linear algebra, statistics, Fourier analysis, and more.

### Key Features of NumPy:

- 1. Multidimensional Arrays:** NumPy provides a powerful array object that can hold elements of the same data type in multiple dimensions. These arrays can be one-dimensional, two-dimensional, or even higher dimensional.
- 2. Element-wise Operations:** NumPy allows performing element-wise operations on arrays. Operations like addition, subtraction, multiplication, and division can be performed on entire arrays without the need for explicit loops.
- 3. Broadcasting:** Broadcasting is a powerful feature in NumPy that allows performing operations on arrays of different shapes and sizes. NumPy automatically handles these operations, making code concise and efficient.
- 4. Linear Algebra Operations:** NumPy provides a wide range of functions for linear algebra, including matrix multiplication, decomposition, eigenvalue calculations, and solving linear equations.
- 5. Random Number Generation:** NumPy includes functions for generating random numbers from various probability distributions, which is essential for simulations and statistical applications.
- 6. Integration with C and Fortran Code:** NumPy arrays can be used to interface with C and Fortran code, providing a bridge between high-level Python and low-level, efficient code written in these languages.
- 7. Efficient and Fast Operations:** NumPy operations are implemented in C and optimized for performance, making them significantly faster than equivalent Python operations performed using loops.

### Example Usage of NumPy:

```
import numpy as np
```

#### # Creating a 1D NumPy array

```
array_1d = np.array([1, 2, 3, 4, 5])
```

#### # Creating a 2D NumPy array

```
array_2d = np.array([[1, 2, 3], [4, 5, 6]])
```

#### # Performing element-wise operations

```
result = array_1d * 2
```

#### # Matrix multiplication

```
matrix_1 = np.array([[1, 2], [3, 4]])
```

```
matrix_2 = np.array([[5, 6], [7, 8]])
```

```
result_matrix = np.dot(matrix_1, matrix_2)
```

```
print("1D Array:", array_1d)
```

```
print("2D Array:")
```

```
print(array_2d)
```

```
print("Element-wise Operation Result:", result)
```

```
print("Matrix Multiplication Result:")
```

```
print(result_matrix)
```

In this example, NumPy is used to create 1D and 2D arrays, perform element-wise operations, and demonstrate matrix multiplication. NumPy's concise and efficient syntax makes it a preferred choice for numerical computations in Python.

## Matplotlib

### What is matplotlib?

Matplotlib is a popular data visualization library in Python used for creating static, interactive, and animated plots and charts. It provides an object-oriented API for embedding plots into applications that use general-purpose GUI toolkits, such as Tkinter, wxPython, Qt, or GTK. Matplotlib is widely used in fields like data science, engineering, finance, and scientific research for visualizing data and conveying insights.

### Key Features of Matplotlib:

#### 1. Versatile Plots:

Matplotlib supports a wide variety of plots, including line charts, scatter plots, bar charts, histograms, 3D plots, and more.

#### 2. Customization:

Users can customize almost every aspect of a plot, including colors, labels, markers, line styles, and annotations, allowing for highly tailored visualizations.

#### 3. Publication-Quality Graphics:

Matplotlib produces high-quality publication-ready graphics. The appearance of plots can be finely tuned to meet the requirements of scientific publications and presentations.

#### 4. Multiple Output Formats:

Matplotlib can export plots to various file formats, including PNG, PDF, SVG, and EPS, making it easy to incorporate plots into documents and presentations.

#### 5. Interactive Plotting:

Matplotlib can be used in interactive mode in IPython and Jupyter Notebook, allowing users to manipulate plots dynamically.

#### 6. Integration with NumPy:

Matplotlib seamlessly integrates with NumPy, making it easy to visualize data stored in NumPy arrays.

#### 7. Extensibility:

Users can create custom plots and visualization types, allowing for the extension of Matplotlib's capabilities.

### Example Usage of Matplotlib:

```
import matplotlib.pyplot as plt
import numpy as np

# Generate data
x = np.linspace(0, 2*np.pi, 100) # Generate 100 points between 0 and 2*pi
y = np.sin(x) # Compute sine values for each x

# Create a line plot
plt.figure(figsize=(8, 6)) # Set the figure size
plt.plot(x, y, label='Sine Function', color='blue', linestyle='-', linewidth=2) # Plot the data
plt.title('Sine Function') # Set the title of the plot
plt.xlabel('x') # Set the x-axis label
plt.ylabel('sin(x)') # Set the y-axis label
plt.legend() # Display legend
plt.grid(True) # Add gridlines
plt.show() # Show the plot
```

In this example, Matplotlib is used to create a line plot of the sine function. The `plt.plot()` function is used to plot the data, and various customization options are applied, such as setting labels, title, legend, and gridlines. Finally, `plt.show()` is called to display the plot. Matplotlib's flexibility and ease of use make it a go-to library for creating a wide range of static and interactive visualizations in Python.

## Seaborn

### What is seaborn?

Seaborn is a statistical data visualization library based on Matplotlib in Python. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn is particularly useful for visualizing complex datasets with multiple variables. It comes with several built-in themes and color palettes to make it easy to create aesthetically pleasing and easily interpretable visualizations.

### Key Features of Seaborn:

#### 1. Statistical Plots:

Seaborn simplifies the process of creating statistical plots such as scatter plots, box plots, violin plots, pair plots, and regression plots. These plots allow you to visualize relationships and distributions in your data.

#### 2. Integration with Pandas:

Seaborn seamlessly integrates with Pandas DataFrames, making it easy to plot data directly from Pandas structures without extensive data manipulation.

#### 3. Automatic Estimation and Aggregation:

Seaborn can automatically estimate and aggregate data for summary visualizations. For instance, it can compute and plot the mean and confidence interval of a dataset.

#### 4. Color Palettes and Themes:

Seaborn offers a variety of color palettes and themes, allowing you to customize the visual style of your plots. This makes it easy to create professional-looking plots with minimal effort.

#### 5. Matrix Plots:

Seaborn provides functions to create matrix plots, including heatmaps and cluster maps, to visualize matrices or high-dimensional datasets.

#### 6. Categorical Plots:

Seaborn supports various categorical plots like bar plots, count plots, and point plots, making it convenient to work with categorical data.



### 7. Time Series Visualization:

Seaborn can be used to create visually appealing time series plots, allowing you to explore temporal patterns in your data.

#### Example Usage of Seaborn:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load a sample dataset from Seaborn
iris = sns.load_dataset("iris")

# Create a pair plot to visualize relationships between variables
sns.pairplot(iris, hue="species", markers=["o", "s", "D"])

# Add a title to the plot
plt.title("Pair Plot of Iris Dataset")

# Show the plot
plt.show()
```

In this example, Seaborn is used to load the famous Iris dataset and create a pair plot. The `sns.pairplot()` function automatically creates scatter plots for all numerical variables in the dataset and colors the points based on the species of the iris flowers. This compact visualization allows for a quick exploration of relationships between variables. Seaborn's simplicity and integration with Pandas make it a popular choice for data visualization, especially in the context of statistical analysis and exploration of complex datasets.

## Jupyter Notebook

### What is Jupyter notebook?

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, Julia, and Scala. Jupyter Notebook is widely used in data analysis, scientific research, machine learning, and education due to its interactive and flexible nature.

### Key Features of Jupyter Notebook:

#### 1. Interactive Computing:

Jupyter Notebook provides an interactive computing environment where you can write and execute code in a step-by-step manner. This interactivity is ideal for exploring data, experimenting with algorithms, and visualizing results.

#### 2. Support for Multiple Languages:

Jupyter supports various programming languages through kernels. Each kernel enables Jupyter to interact with a specific programming language, allowing users to work with multiple languages in the same notebook.

#### 3. Rich Output:

Notebooks can display rich outputs, including tables, charts, images, videos, and interactive widgets. This feature allows you to create visually appealing and informative documents.

#### 4. Combination of Code and Text:

Jupyter notebooks combine executable code cells with text cells written in Markdown. This combination allows you to include explanations, documentation, and mathematical equations alongside your code.

#### 5. Visualization Capabilities:

Jupyter supports popular Python libraries for data visualization, such as Matplotlib, Seaborn, and Plotly. You can create interactive and static visualizations directly within your notebooks.

#### 6. Notebook Sharing and Collaboration:

Notebooks can be easily shared with others. Platforms like GitHub, JupyterHub, and Google Colab allow collaborative editing and sharing of notebooks with colleagues and the wider community.

### 7. Educational Tool:

Jupyter Notebook is widely used in educational settings to teach programming, data analysis, and scientific computing. It provides an interactive and engaging environment for learning.

#### How Jupyter Notebook Works:

##### 1. Web-Based Interface:

Jupyter Notebook runs in your web browser. You access the interface by running a local server or using online platforms like Google Colab.

##### 2. Cells:

Notebooks consist of cells, which can contain code, text, equations, or visualizations. Code cells are executed one at a time, and their outputs are displayed below the cell.

##### 3. Kernel:

Each notebook is associated with a specific kernel, which is responsible for executing code in the notebook. Kernels can be restarted without affecting the notebook's text or output cells.

##### 4. Saving and Exporting:

Notebooks can be saved in the .ipynb format, allowing you to preserve the code, text, and output. Notebooks can also be exported to other formats such as HTML, PDF, and slides.

Jupyter Notebook provides an interactive and dynamic way to work with code and data, making it a valuable tool for researchers, data scientists, educators, and developers.

#### Why Jupyter Notebook is Essential:

##### 1. Interactive Computing:

Jupyter Notebook provides an interactive computing environment. You can write and execute code in a step-by-step manner, making it ideal for data exploration, algorithm testing, and prototyping.

##### 2. Mix of Code and Text:

Notebooks combine executable code cells with text cells written in Markdown. This combination allows you to provide explanations, document your thought process, and include mathematical equations alongside your code. It's an excellent tool for creating interactive narratives around your data and code.

##### 3. Data Visualization:

Jupyter supports popular data visualization libraries like Matplotlib, Seaborn, and Plotly. You can create static and interactive charts directly within your notebook, aiding in data exploration and presentation.

### **4. Education and Learning:**

Jupyter is widely used in educational settings. It provides an interactive environment for teaching programming, data science, and scientific computing. Students can experiment with code, visualize results, and learn interactively.

### **5. Experimentation and Prototyping:**

For data scientists and researchers, Jupyter Notebook is invaluable. It allows for quick experimentation with algorithms, immediate visualization of results, and easy collaboration with colleagues. The ability to document each step ensures that experiments are reproducible.

### **6. Data Analysis and Exploration:**

Jupyter Notebooks are excellent for data analysis. You can import datasets, clean and preprocess data, perform exploratory data analysis, and visualize the results—all within a single document. This interactive approach significantly speeds up the data analysis process.

### **7. Reproducibility:**

Notebooks capture the code, the output, and the explanations in a single document. This makes it easier to share your work and ensure that others can reproduce your results, fostering transparency and collaboration.

### **8. Collaboration and Sharing:**

Notebooks can be shared with others, making it an excellent collaboration tool. Platforms like GitHub, JupyterHub, and Google Colab allow collaborative editing and sharing of notebooks, enhancing teamwork and knowledge sharing.

### **9. Flexible Environment:**

Jupyter Notebook is flexible and can be extended. There are extensions available that enhance its functionality, allowing users to customize their environment according to their needs.

In summary, Jupyter Notebook is a powerful tool that facilitates interactive computing, data analysis, and collaborative work. Its ability to seamlessly integrate code, visualizations, and explanations makes it a must-have tool for anyone working with data, whether you're a beginner learning programming or an experienced data scientist conducting advanced analyses.

## Project Introduction

### Project Name: Diwali Sales Analysis

#### Introduction

Diwali, the festival of lights, is one of the most significant festivals celebrated in India. This project aims to analyze Diwali sales data to gain insights into customer behavior, popular products, and overall market trends during this festive season. Python programming language and its data analysis libraries have been used for this purpose.

#### Objectives

- Analyze Diwali sales data to identify patterns and trends.
- Determine the best-selling products and categories.
- Understand customer preferences and buying behavior.
- Provide data-driven recommendations for businesses to optimize their Diwali sales strategies.

#### Methodology

##### 1. Data Collection

Sales data for the Diwali season was obtained from various e-commerce platforms and retail stores. The dataset includes information such as product ID, category, price, quantity sold, and customer demographics.

##### 2. Data Preprocessing

- **Data cleaning:** Handling missing values, removing duplicates, and correcting inconsistencies.
- **Data transformation:** Converting categorical variables into numerical format if necessary.
- **Feature engineering:** Creating new features like total sales, average transaction value, etc.

##### 3. Data Analysis Tools

- **Programming Language:** Python 3.x
- **Libraries:** Pandas, NumPy, Matplotlib, Seaborn for data manipulation and visualization.
- **IDE:** Jupyter Notebook for interactive data analysis.

### Analysis

#### 1. Overall Sales Trends

Insert visualizations showing overall sales trends during the Diwali season, including total sales, daily sales fluctuations, and popular buying times.

#### 2. Best-Selling Products and Categories

Show visualizations and statistics indicating the top-selling products and categories. Include insights on why these products were popular, such as discounts or marketing strategies.

#### 3. Customer Behavior Analysis

Analyze customer demographics, including age groups, locations, and purchasing power. Include visualizations and patterns observed.

### Results

- Diwali sales peaked during specific days, indicating the most lucrative period for businesses.
- Electronics and clothing or foods were the top-selling categories, with Smartphone's and traditional clothing being the best-selling products.
- Young adults (18-35 years) contributed significantly to the sales, especially in urban areas.
- Discounts and promotional offers greatly influenced customer purchasing decisions.

### Conclusion

The analysis of Diwali sales data provides valuable insights for businesses looking to enhance their marketing and sales strategies. Understanding customer preferences and market trends during festive seasons is crucial for maximizing revenue.

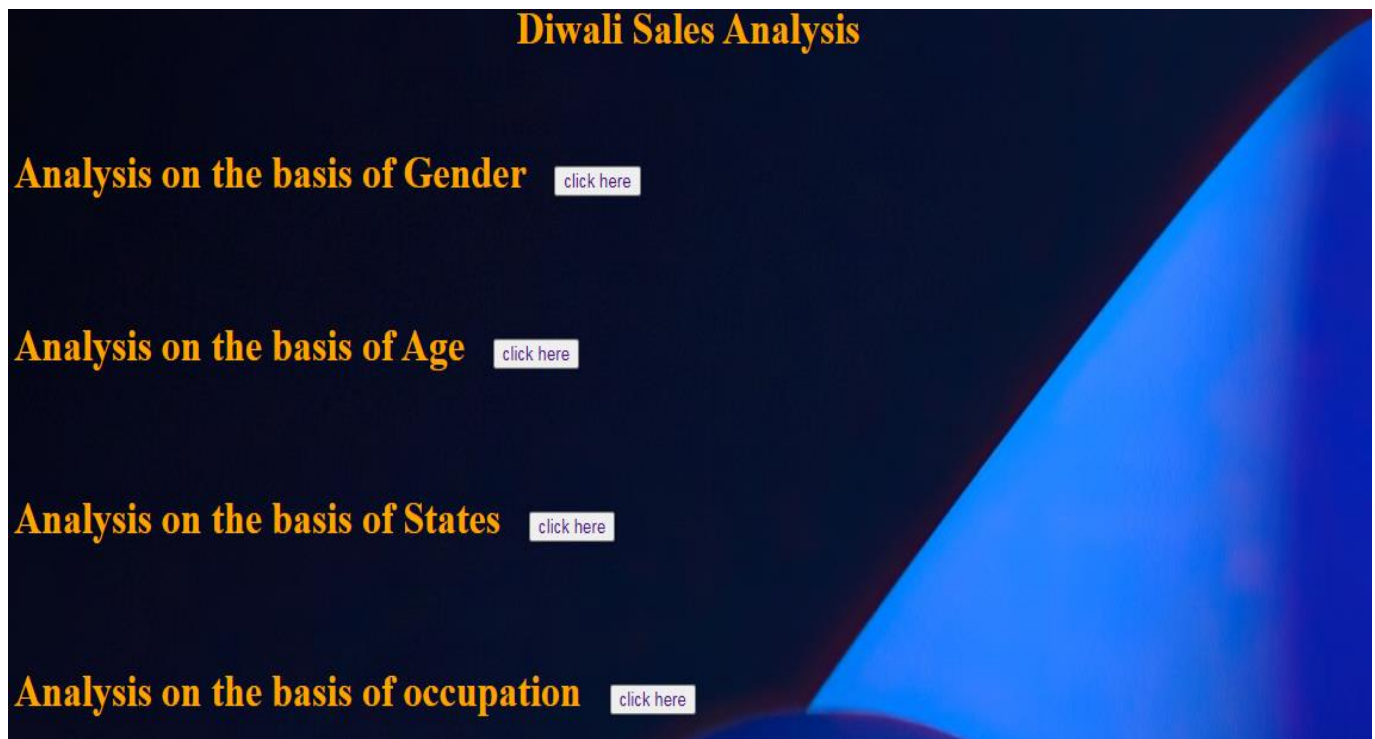
### Future Work

- 1. Predictive Analysis:** Implement machine learning algorithms to predict future sales trends based on historical data.
- 2. Customer Segmentation:** Perform advanced clustering techniques to identify different customer segments and tailor marketing strategies accordingly.
- 3. Sentiment Analysis:** Integrate sentiment analysis to understand customer feedback and reviews, enabling businesses to improve product offerings and customer satisfaction.

**Aim:** This project report summarizes the data analysis process, findings, and recommendations based on the Diwali sales dataset. It serves as a valuable resource for businesses aiming to optimize their sales strategies during festive seasons.

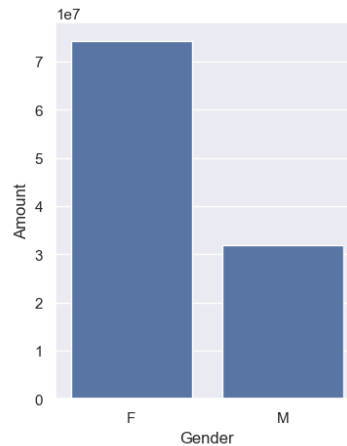
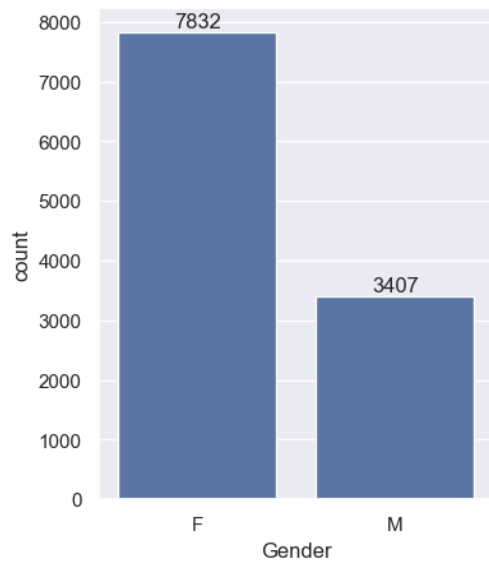
In summary, Jupyter Notebook is a powerful tool that facilitates interactive computing, data analysis, and collaborative work. Its ability to seamlessly integrate code, visualizations, and explanations makes it a must-have tool for anyone working with data, whether you're a beginner learning programming or an experienced data scientist conducting advanced analyses.

## Implementation





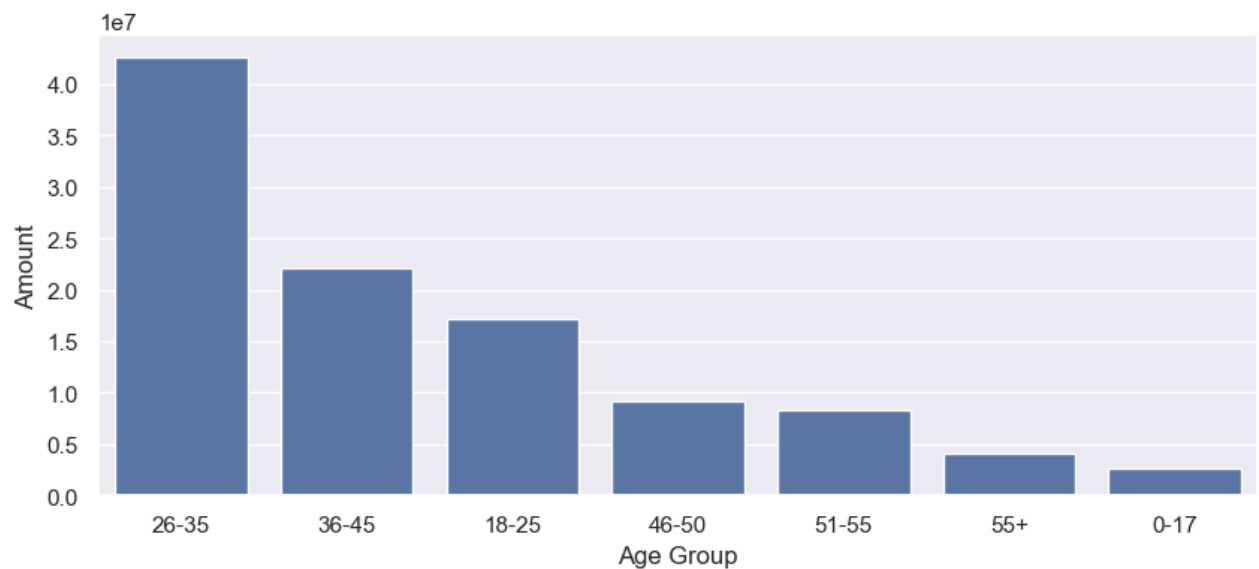
### Analysis on the basis of Gender:

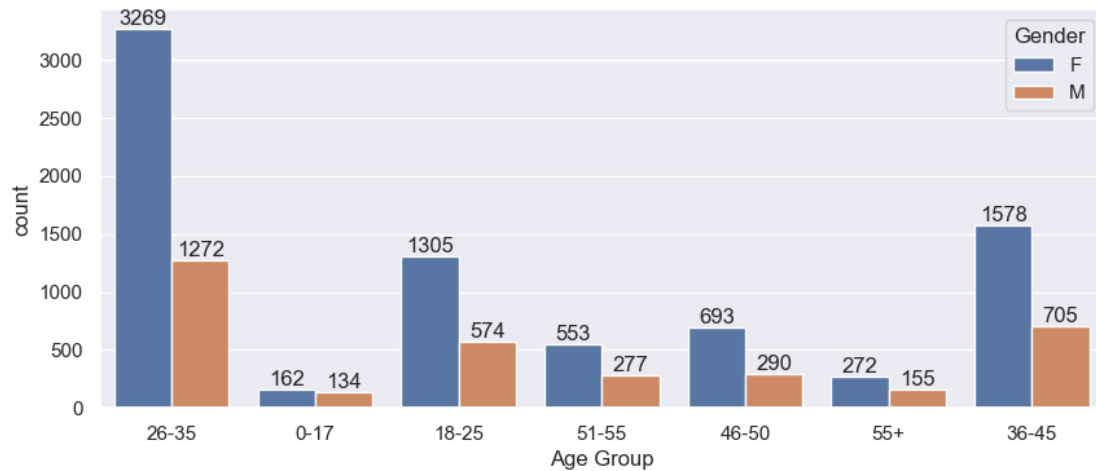


From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

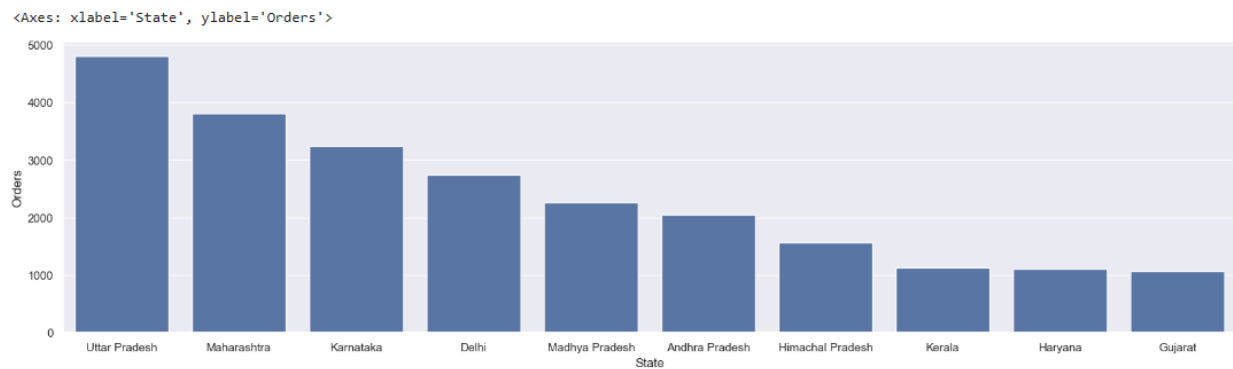
### Analysis on the basis of AGE GROUP

<Axes: xlabel='Age Group', ylabel='Amount'>

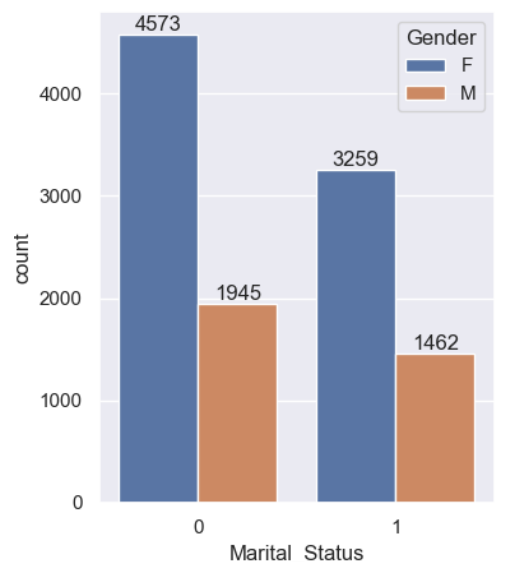




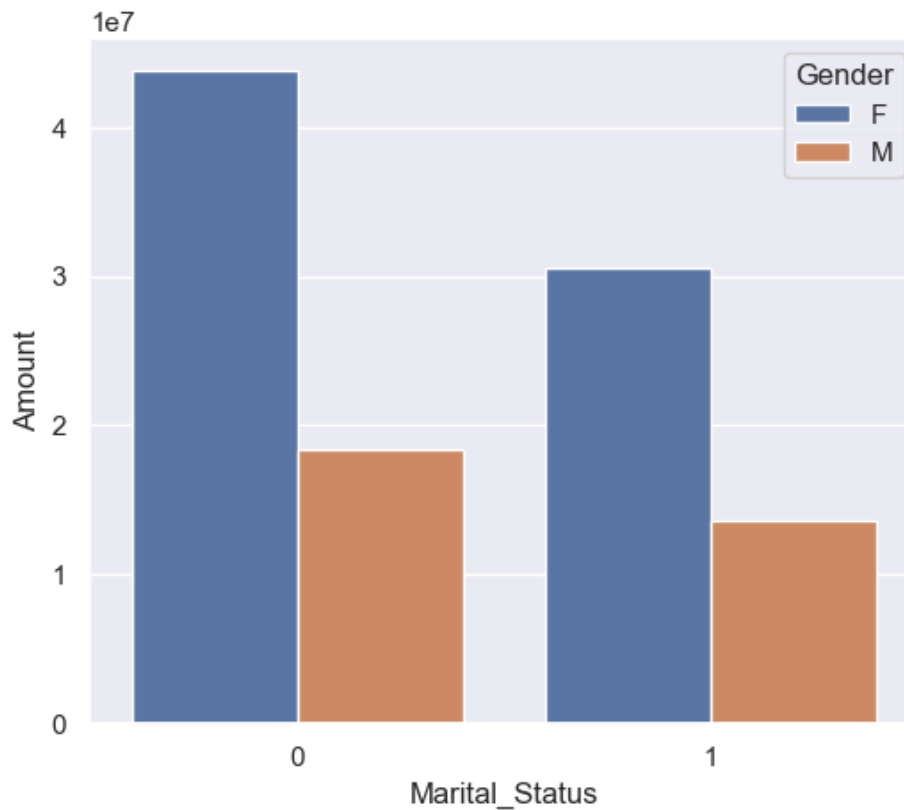
### Analysis on the basis of State



### Analysis on the basis of Marital Status



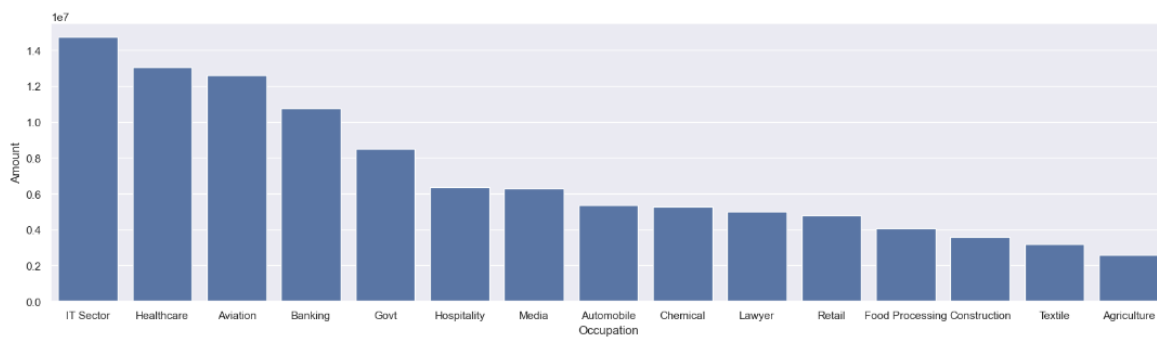
<Axes: xlabel='Marital\_Status', ylabel='Amount'>



*From above graphs we can see that most of the buyers are married (women) and they have high purchasing power*

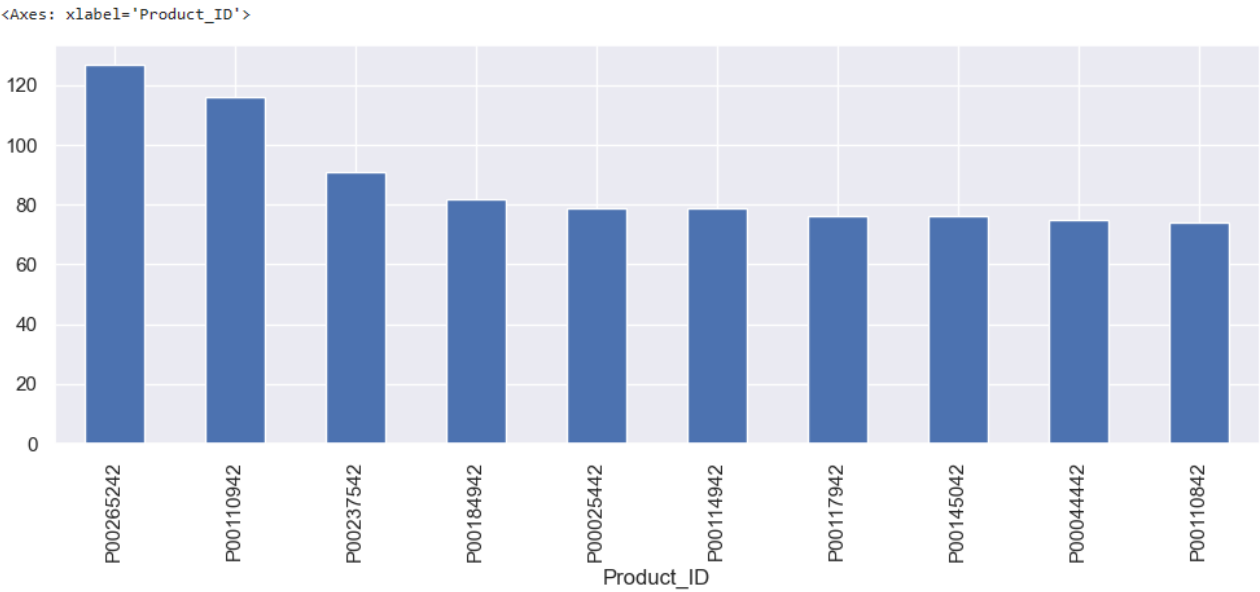
### Analysis on the basis of Occupation

<Axes: xlabel='Occupation', ylabel='Amount'>



*From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector*

Analysis on the basis of Product\_ID



## Bibliography

- <https://www.w3schools.com/> :- Learn Python
- <https://github.com/> :- Get the CSV file
- <https://www.youtube.com/> :-Data Analysis Basic concepts