Assignment: 1

Aim :-

To learn how to load and store data, with different file formats.

Objective :-

Analysing sales data from multiple file formats.

Tasks to Perform :-

Obtain sales data files in various formats, such as CSV, Excel, and JSON. Now do the following:

1. Load the sales data from each file format into the appropriate data structure or data-frames.
2. Explore the structure and content of the loaded data, identifying any inconsistencies, missing values, or data quality issues.
3. Perform data cleaning operations, such as handling missing values, removing duplicates, or correcting inconsistencies.
4. Convert the data into a unified format, such as a common dataframe or data structure, to enable seamless analysis.
5. Perform data transformation tasks, such as merging multiple datasets, splitting columns, or deriving new variables.
6. Analyses the sales data by performing descriptive statistics, aggregating data by specific variables,

or calculating metrics such as total sales, average order value, or product category distribution.
7. Create visualizations, such as bar plots, pie charts, or box plots, to represent the sales data and gain insights into sales trends, customer behaviour, or product performance.

Theory :-

In the field of data science and analytics, the ability to work with data in various files formats is crucial. Sales data, often in formats like csv, Excel, and JSON, is a prime example of diverse data sources that require careful handling. This theory aims to provide a comprehensive framework for loading, cleaning, transforming, and analysing sales data from these multiple formats.

The process outlined here enables data scientists to derive meaningful insights and make informed decisions from sales data, irrespective of its original format.

1. Data Loading :-

Loading data from different formats is the initial step in the data analysis pipeline. In this section, we will explore techniques for loading sales data from csv, Excel, and JSON files into a structured dataset. This involves using libraries and tools such as pandas in Python to read and ingest data.

2. Data cleaning :-

Sales data, often originating from

various sources, is prone to inconsistence, missing values, and outliers. Effective data cleaning is essential to ensure data quality. We will discuss techniques for identifying and addressing issues like duplicate records, missing value data and data outliers.

3. Data Transformation :-

To prepare the sales data for analysis, it may be necessary to perform data transformations. This includes tasks such as data normalization, feature engineering, and merging datasets if data is spread across multiple files or sources. We will explore the methods and best pratices for transforming sales data into a format conducive to analysis.

4. Data Analysis :-

Once the data is cleaned and transformed, we move on to the analysis phase. Here, we will delve into various analytical approaches suitable for sales data. This may include descriptive statistics, time-series analysis, correlation analysis, and predictive modeling techniques.

5. Visualization :-

Effective data visualization is an integral part of data analysis. We will explore how to create compelling visualizations using tools like matplotlib and Seaborn.

6. Case Studies :-

To illustrate the theory in practice, we will present case studies showcasing real-world scenarios of loading, transforming, cleaning and

analyzing sales data from CSV, Excel and JSON formats. These case studies will provide practical insights into applying the theory to solve specific business problems.

Conclusion :-

In conclusion, the ability to handle and analyze sales data from diverse formats is a fundamental skill for data scientists and analysts. This theory provides a structured approach to efficiently load, clean, transform, and analyze sales data, regardless of its original format. By following the outlined steps, data professionals can unlock valuable insights and drive data-informed decision-making in sales and business contexts.

Assignment - 2

Aim :-

Analyzing Weather Data from OpenWeatherMap API.

Objectives :-

The goal is to interact with the OpenWeatherMap API to retrieve weather data for a specific location and perform data modelling and visualization to analyze weather patterns over time.

Tasks to perform :

1. Register and obtain API key from OpenWeatherMap.
2. Interact with the OpenWeatherMap API using the API key to retrieve weather data for a specific location.
3. Extract relevant weather attributes such as temperature, humidity, wind speed, and precipitation from the API response.
4. Clean and preprocess the retrived data, handling missing values or inconsistent formats.
5. Perform data modeling to analyze weather patterns, such as calculating average temperature, max/min values, or trends over time.
6. Visualize the weather data using appropriate plots, such as line charts, bar plots, or scatter plots, to represent temperature changes, precipitation levels, or wind speed variations.

7. Apply data aggregation techniques to summarize weather statistics by specific time periods.
8. Incorporate geographical info, if available, to create maps or geospatial visualizations representing weather patterns across different locations.
9. Explore and visualize relationships between weather attributes, such as temperature and humidity, using correlation plots or heatmaps.

Theory :-

Understanding weather patterns is crucial for various sectors, including agriculture, transportation, and disaster management. This theory presents a comprehensive framework for interacting with the OpenWeatherMap API to retrieve weather data for a specific location, followed by data modeling and visualization techniques to analyze weather patterns over time. The steps involved are :-

1. Data Retrieval from OpenWeatherMap API :-

Interacting with the OpenWeatherMap API involves making HTTP requests to retrieve weather data. This section discusses the process of API key authentication, endpoint selection, and making GET requests to fetch weather data for a particular location. We will explore how to parse the JSON responses to extract relevant weather information.

2. Data Preprocessing and Cleaning :-

Raw weather data often contains missing values, outliers, and inconsistencies. Effective data preprocessing is essential to ensure data quality. In this section, we will cover ~~Inherit~~ techniques ~~for~~ handling missing data, data imputation, and outliers detection. This phase prepares the data for modeling and visualization.

3. Time-Series Data Modeling :-

Weather data is inherently time-series data, which means it is collected over time. Time series analysis is crucial for understanding weather patterns. This section delves into various time series modeling techniques, such as autoregressive models (ARIMA), exponential smoothing, and seasonal decomposition to identify trends, seasonality, and anomalies in the weather data.

4. Data Visualization :-

Visualizing weather data is instrumental in conveying patterns and insights effectively. We will explore data visualization libraries like Matplotlib, Seaborn, and Plotly to create interactive plots and charts.

5. Analyzing Weather Patterns :-

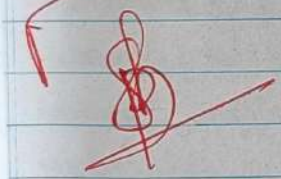
With cleaned data and informative visualizations in hand, we move on to analyzing weather patterns. This includes identifying long-term trends, seasonal variations, extreme weather events, and correlation with other factors.

6. Case Studies and Practical Applications :-

To demonstrate the theory's practicality, case studies will be presented. These studies will showcase how the theory can be applied to specific scenarios, such as predicting extreme weather events, optimizing energy consumption, or assessing climate change impacts.

Conclusion :-

Hence, we successfully interacted with the OpenWeather API to retrieve weather data for a specific location and perform data modeling & visualization to analyze weather patterns over time.



Assignment - 3

Aim :-
To clean and prepare data for analysis.

Objectives :-
Analysing customer churn in a Telecommunications Company.

Tasks to perform :-
1. Import the "Telecom_Customer_churn.csv" dataset.
2. Explore the dataset to understand its structure and content.
3. Handle missing values in the dataset, deciding on an appropriate strategy.
4. Remove any duplicate records from the dataset.
5. Check for inconsistent data, such as inconsistent formatting or spelling variations, and standardize it.
6. Convert columns to the correct data types as needed.
7. Identify and handle outliers in the data.
8. Perform feature engineering, creating new features that may be relevant to predicting customer churn.
9. Normalize or scale the data if necessary.
10. Split the dataset into training and testing sets for further analysis.
11. Export the cleaned dataset for future analysis or modeling.

Theory :-
Customer churn, the rate at which

customers discontinue their subscriptions or services, is a pressing concern in the telecommunications industry. To address this issue, this theory presents a comprehensive framework for analysing customer churn in a telecommunications company. By leveraging this theory, telecom providers can gain insights into the factors driving customer attrition and develop strategies to retain their valuable subscribers.

The procedure to follow is:

1. Data Collection and Preparation :-

The first step in analysing customer churn is to collect relevant data. This section discusses the sources of customer data, including subscription records, call logs, billing information, and customer demographics. Data preparation steps such as data cleaning, transformation, and feature engineering are also explored to ensure the dataset is suitable for analysis.

2. Defining Customer Churn :-

Defining customer churn is a critical aspect of the analysis. This section explores various definitions of churn, such as when a customer cancels a subscription, reduces usage significantly, or switches to a competitor. Clear and precise churn definitions are crucial for accurate analysis.

3. Exploratory Data Analysis (EDA) :-

Exploratory Data Analysis is essential for gaining initial insights into the data. This section covers techni-

ques like data visualization, summary statistics, and correlation analysis to identify patterns and relationships within the customer data. EDA help in understanding factors that may contribute to churn.

4. Feature Selection and Engineering :-

Selecting and engineering relevant features are pivotal for building predictive models. This section explores methods for feature selection, dimensionality reduction, and creating new features that may impact churn prediction, such as customer tenure, service usage, and billing patterns.

5. Churn Prediction Models :-

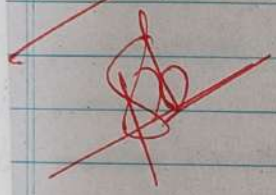
churn prediction models are at the heart of customer churn analysis. This section discusses various machine learning techniques and algorithms, including logistic regression, decision trees, random forests, and neural NW's, that can be employed to predict customer churn. Model evaluation, validation and hyperparameter tuning are also covered.

6. Interpretability and Insights :-

Once churn prediction models are built, it's essential to interpret the results. This section explores methods for models for model interpretability, such as feature importance analysis and SHAP values. Insights gained from these models can guide decision-making regarding customer retention strategies.

Conclusion :-

We successfully analyzed customer churn in a telecommunications company and achieved all the tasks as metho mentioned in the task to perform.



Assignment - 4

Aim :-
To perform data wrangling.

Objective :-
Data Wrangling on Real Estate Market.

Tasks to perform :
1. Import the "RealEstate_Prices.csv" dataset. Clean column names by removing spaces, special characters, or renaming them for clarity.
2. Handle missing values in the dataset, deciding on an appropriate strategy.
3. Perform data merging if additional datasets with relevant information are available.
4. Filter and subset the data based on specific criteria, such as a particular time period, property type, or location.
5. Handle categorical variables by encoding them appropriately for further analysis.
6. Aggregate the data to calculate summary statistics or derived metrics such as average sale prices by neighborhood or property type.
7. Identify and handle outliers or extreme values in the data that may affect the analysis or modeling process.

Theory :-
Understanding the factors that influence housing prices is crucial for both buyers &

sellers in the real estate market. This theory presents a structured framework for performing data wrangling on a dataset that includes attributes related to housing characteristics, location, sale prices, and other relevant features. The required steps to perform data wrangling are:

1. Data Collection and Assessment:-

The initial step in the data wrangling process in data collection and assessment. This section discusses the sources of housing data, including property listings, public records, or real estate databases. It also covers techniques for data assessment, including checking for data completeness, data types, missing values, and outliers.

2. Data Cleaning and Preprocessing:-

Data cleaning is imperative to ensure data quality. In this section, various data cleaning tasks are discussed, such as handling missing data through imputation, addressing outliers, and dealing with duplicate records. Data preprocessing techniques, including feature scaling, encoding categorical variables, and creating new features, are also explored to prepare the dataset for analysis.

3. Exploratory Data Analysis (EDA):-

Exploratory Data Analysis is a crucial step to gain insights into the dataset. This section covers techniques such as data visualization, summary statistics, and

correlation analysis to identify patterns and relationships within the housing data. EDA helps in understanding which factors may influence housing prices.

4. Feature Engineering :-

Feature engineering is about creating meaningful features that can enhance predictive modeling. This section explores methods for feature selection and engineering, including deriving new attributes from existing ones, transforming variables, and considering interactions between features. The goal is to identify the most relevant predictors of housing prices.

5. Handling Geospatial Data :-

For housing data, location is often a critical factor. This section discusses how to handle geospatial data, including latitude, longitud, and address information. Techniques for geocoding, spatial analysis, and the incorporation of external geographic datasets are explored to extract valuable insights related to location

6. Data Imputation and Missing Value Handling :-

Handling missing values effectively is essential. This section dives deeper into strategies for imputing missing data, such as mean imputation, model-based imputation, and using external data sources to fill gaps in the housing dataset.

7. Data Export and Documentation :-

Once data arangling

is complete, it's important to export the cleaned and preprocessed dataset for further analysis or modeling. Proper documentation of the data cleaning and preprocessing steps is also essential for transparency & reproducibility.

Conclusion :-

In conclusion, data wrangling plays a pivotal role in preparing a housing dataset for analysis or modelling.

Assignment - 5

Aim :-

Data Visualization using matplotlib

Objective :-

Analysing Air Quality Index (AQI) Trends in a city.

Task to Perform :

1. Import the "City-Air-Quality.csv" dataset.
2. Explore the dataset to understand its structure and content.
3. Identify the relevant variables for visualizing AQI trends, such as date, pollutant levels & AQI values.
4. Create line plots or time series plots to visualize the overall AQI trend over time.
5. Plot individual pollutant levels on seperate line plots to visualize their trends over time.
6. Use bar plots or stacked bar plots to compare the AQI values across different dates or time periods.
7. Create box plots or violin plots to analyze the distribution of AQI values for different pollutant categories.
8. Use scatter plots or bubble charts to explore the relationship between AQI values and pollutant levels.
9. Customize the visualizations by adding labels, titles, legends and appropriate color schemes.

Theory :-

Understanding air quality trends and pollutant levels is critical for addressing environmental concerns and public health. This theory presents a structured framwork for utilizing the "matplotlib" library to create effective visualizations that represent the Air Quality Index (AQI) trends & pollutant levels in a specific city over time. The dataset under consideration contains attributes such as a date, time, various pollutant levels, and corresponding AQI values.

1. Data Import and Exploration :-

The first step in this process is importing the "City-Air-Quality.csv" dataset. Subsequently, we explore the dataset to understand its structure and content. This involves examining data types, dimensions, and initial summary statistics to gain insights into the data's characteristics.

2. Identification of Relevant Variables :-

To create effective visualizations, we need to identify the relevant variables. In this case, key variables for visualizing AQI trends include date, pollutant levels, and AQI values.

3. Line Plots for Overall AQI Trend :-

To understand the overall AQI trend over time, line plots or time series plots are created. These visualizations provide insights into how air quality has changed throughout the observed

period.

4. Individual Pollutant level Plots :-

Seperate line plots are constructed for individual pollutant level to visualize their trends over time. This allows for a more detailed examination of pollutant-specific variations.

5. Bar Plots for Comparing AQI Values :-

Bar plots or stacked bar plots are used to compare AQI values across different dates or time periods. This enables the assessment of air quality fluctuations and variations over time.

6. Box Plots for AQI Value Distribution :-

Box plots or violin plots are created to analyze the distribution of AQI values for different pollutant categories. These visualizations help in identifying the spread and central tendencies of AQI data.

7. Scatter Plots for AQI-Pollutant Relationships :-

Scatter plots or bubble charts are employed to explore the relationship between AQI values and pollutant levels. These visualizations allow for the assessment of correlation and associations betn various pollutants and air quality.

8. Customization and Enhancement :-

To enhance the interpretability of the visualizations, customizations are applied, including the addition of labels, titles, legends, and appropriate color schemes. These

elements help in making the visualizations more informative and visually appealing.

Points for better visual presentation

1. Axis Labels and Titles:
   - Add clear and informative labels to the x and y-axes to indicate what each axis represents.
   - Include a title that succinctly describes the purpose of the visualization, making it easier for the viewer to understand the content.

2. Legends and Labels:
   - Include legends when plotting multiple data series to explain the meaning of each line or element.
   - Label individual data points, lines, or bars when necessary to provide context and clarity.

3. Color Selection:
   - Use a consistent and well-thought-out color scheme to differentiate between data series or categories.
   - Avoid using too many colors, as it can make the visualization confusing. Consider using color sparingly for emphasis.

And many more such changes can be made in order to enhance the visual effects of the charts, graphs.. etc

Conclusion :-

Leveraging the "matplotlib" library for visualizing air quality trends and pollutant levels in the dataset provides valuable insights for environmental monitoring and policy-making. The visualizations created through this framework will aid in understanding the dynamics of air quality in the specific city over time.

Assignment - 6

Aim :-

To aggregate data and visualize it.

Objective :-

Analyzing Sales Performance by Region in a Retail Company.

Tasks to perform :-

1. Import the "Retail_Sales_Data.csv" dataset.
2. Explore the dataset to understand its structure and content.
3. Identify the relevant variables for aggregating sales data, such as region, sales amount, and product category.
4. Group the sales data by region and calculate the total sales amount for each regions.
5. Create bar plots or pie charts to visualize the the sales distribution by region.
6. Identify the top-performing regions based on the highest sale amount.
7. Group the sales data by region and product category to calculate the total sales amount for each combination.
8. Create stacked bar plots or grouped bar plots to compare the sales amounts across different regions and product categories.

Theory :-

Efficient analysis of sales transactions is fundamental for retail companies to understand

their performance and make informed decisions. This theory presents a structured framwork for leveraging the "Retail-Sales-Data.csv" dataset, which contains attributes such as transaction date, product category, quality sold, and sales amount. The primary goal is to perform data aggregation to analyze sales performance by region and identify the top-performing regions. This process involves exploring the dataset, identifying relevant variables, aggregating sales data and creating visualizations to facilitate insights. The following is the process to the needful:

1. Data Import and Exploration :-

The initial step involves importing the "Retail-Sales-Data.csv" dataset. Following that, data exploration is crucial to comprehend the dataset's structure, dimensions, and content. Exploratory data analysis (EDA) helps in gaining an initial understanding of the data's characteristics.

2. Identification of Relevant Variables :-

To aggregate sales data effectively, we must identify the relevant variables. In this context, key variables for data aggregation include region, sales amount, and product category.

3. Sales Data Aggregation by Region :-

In this, we group the sales data by region and calculate the total sales amount for each region. Aggregating sales data by region provides insights into regional

performance and allows for the identification of top-performing regions.

4. Visualization of Sales Distribution by Region :-

To visualize the distribution of sales by region, we create bar plots or pie charts. These visualizations offer a clear representation of how sales are distributed across different regions, enabling stakeholders to quickly grasp regional variations.

5. Identification of Top-Performing Regions :-

Top-performing regions are determined based on the highest sales amount. By analyzing the aggregated data, we can identify the regions that contribute the most to overall sales and assess their performance.

6. Sales Data Aggregation by Region and Product Category :-

To gain a more detailed perspective, we group the sales data by both region and product category. This allows us to calculate the total sales amount for each combination of region and product category, providing insights into the most lucrative product categories within different regions.

7. Comparison of Sales Amounts :-

To compare sales amount across different regions and product categories, we create stacked bar plots or grouped bar plots. These visualizations helps in understanding how sales are distributed across various regions and product categories, facilitating strate-

gic decision-making.

Conclusion :-

We sucessfully completed analyzing Sales Performance by Region in a Retail Company.