AIR Q ASSESMENT TN

Phase 2: innovation

Project Definition:

- The project involves analyzing air quality data to assess the suitability of air for specific purposes, such as breathing.
- The objective is to identify potential issues or deviations from regulatory standards and determine air probability based on various parameters.
- This project involves analysis objectives , collecting air quality data, designing relevant visualizations, building a predictive model.

Dataset: https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014

Data Collection and Preprocessing:

- Using air quality dataset. This dataset may include information about air pollutants like PM2.5, PM10, CO, SO2, NO2, temperature, humidity, etc.
- Preprocessing the data:

Handling missing values, outliers, and ensure it's in a suitable format machine learning.

Example:

Checking for missing values using isnull() and notnull()

Filling for missing values using fillnull() and replace()

Features Engineering:

Create relevant features of air quality or transform existing ones.

For Example: we might calculate daily average or identify seasonal trends.

```
import numpy as np


# Sample air quality data (replace with your own data)
air_quality_data = = pd.read_csv("c:\users\ELCOT\Downloads)
```

```
# Calculate the average
average_air_quality = np.mean(air_quality_data)
# Print the result
print("Average Air Quality:", average_air_quality)
```

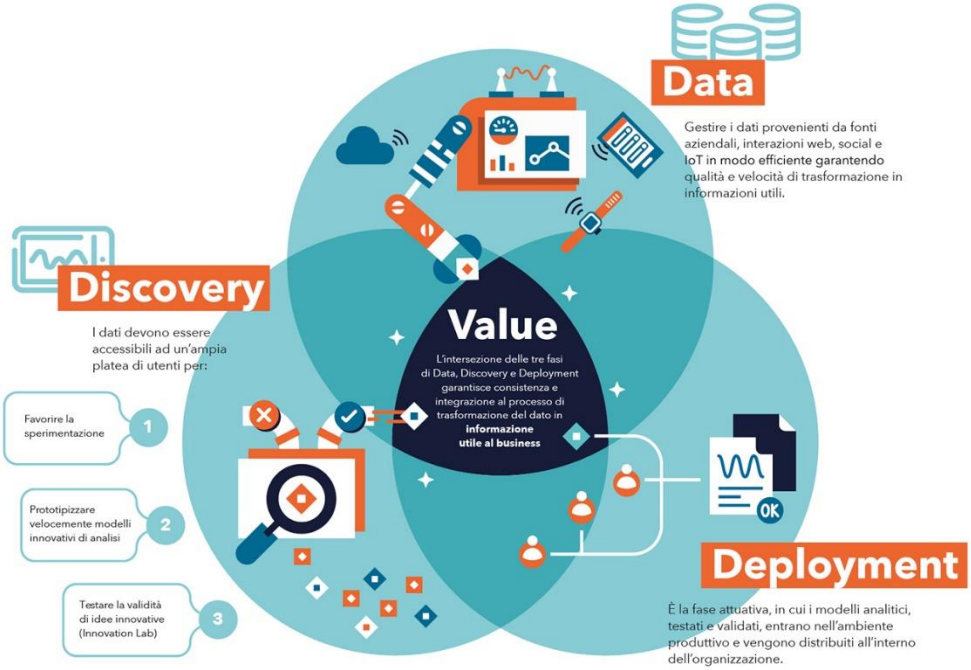
Analytics in Action:

This data is often structured and organized to support data analysis and decision-making. Analytical data in Cognos can come from various sources and be transformed and modeled to meet the specific needs of business users. Here are key aspects related to analytical data in Cognos

- Data Sources:
  Analytical data can originate from various sources, including databases, data warehouses, spreadsheets, and external data feeds. Cognos can connect to these sources to access and analyze data.
- Data Modeling:
  Data modeling in Cognos involves defining the structure and relationships within the data. It helps users understand how different data elements are related and organized, making it easier to query and analyze the data.
- ETL (Extract, Transform, Load):
  Data extraction, transformation, and loading are crucial processes to prepare data for analysis. Cognos provides tools and capabilities to extract data from source systems, transform it to suit analytical needs, and load it into its data model.
- Data Exploration:
  Cognos offers features for data exploration, allowing users to navigate, filter, and interact with the data to identify patterns, trends, and anomalies.
- Data Visualization:
  Cognos enables users to create various data visualizations, such as charts, graphs, and dashboards, to represent data in a visually meaningful way.
- Reporting:
  Cognos allows users to create reports based on the analytical data. These reports can be customized, scheduled, and distributed to relevant stakeholders.
- Ad-Hoc Analysis:

Business users can perform ad-hoc analysis by creating their own queries and reports, enabling them to explore data and generate insights without relying on IT or data analysts.

- Data Security and Governance:
  Cognos provides tools for managing data security and governance. This ensures that only authorized users have access to sensitive data, and data remains compliant with organizational policies and regulations.



- Data Integration:
  Analytical data in Cognos may involve the integration of data from various sources, making it possible to analyze and report on a comprehensive set of data.
- Data Performance:
  Cognos optimizes data retrieval and analysis performance, ensuring that analytical processes run efficiently, even with large datasets.

Visualization:

Data visualization is the graphical representation of information and data. By using visual elements like chart, graph and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion. In the world of Big Data, data

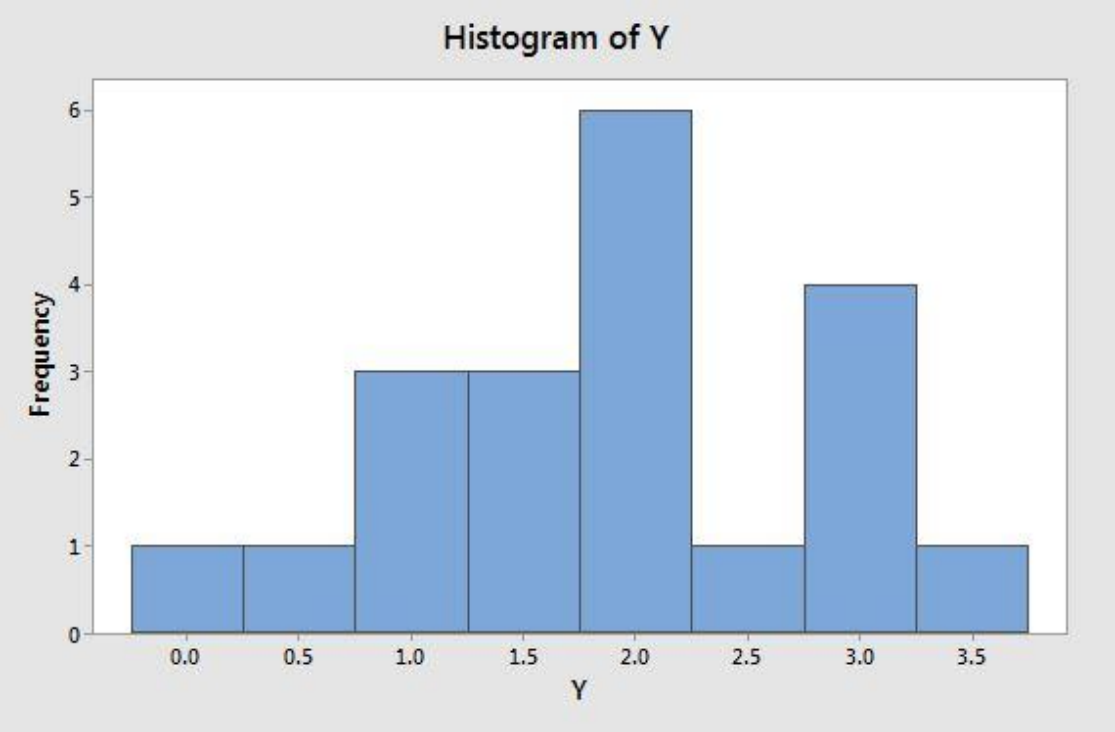
visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions

Histogram:

A histogram can be defined as a set of rectangles with bases along with the intervals between class boundaries. Each rectangle bar depicts some sort of data and all the rectangles are adjacent. The heights of rectangles are proportional to corresponding frequencies of similar as well as for different classes.

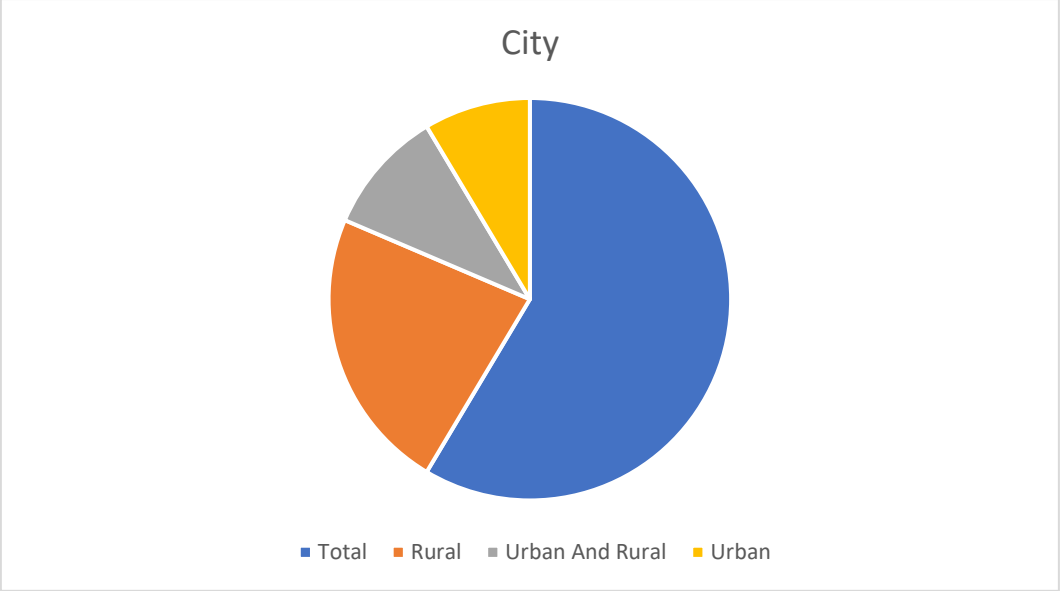
```
import matplotlib.pyplot as plt
import numpy as np


# Sample air quality data (replace with your own data)
air_quality_data = pd.read_csv(“c:\users\ELCOT\Downloads)
# Create a histogram
plt.hist(air_quality_data, bins=10, edgecolor='black', alpha=0.7)
plt.title("Histogram of  Y")
plt.xlabel("Y")
plt.ylabel("Frequency")
plt.grid(True)
# Display the histogram
plt.show()
```



Pie Charts:

Pie graphs are used to show the distribution of qualitative (categorical) data.It shows the **frequency** or **relative frequency** of values in the data .Frequency is the amount of times that value appeared in the data. Relative frequency is the percentage of the total. Each category is represented with a slice in the 'pie' (circle). The size of each slice represents the frequency of values from that category in the data.



Analyzing and Handling Data set:

- Load the Dataset:
  - Import the necessary libraries (e.g., Pandas, NumPy) in Python.
  - Load your dataset into a data structure (e.g., a Pandas DataFrame)

    ```
    import pandas as pd
    data = pd.read_csv('your_dataset.csv')
    ```

- Missing Values:
  Identify and handle missing values in your dataset. You can drop rows with missing values, impute values, or use other strategies depending on the context.

  ```
  print(data.head())
  print(data.info())
  print(data.describe())
  ```

- Duplicate Data:
  Check for and remove duplicate rows if necessary.

  ```
  data.dropna()  # To remove rows with missing values
  data.fillna(value)  # To fill missing values with a specific value
  ```

- Outliers:
  Identify and handle outliers in your data. You can use statistical methods or visualization techniques to detect outliers.
  ```
  # Example using Z-score for outlier detection
  from scipy import stats
  z_scores = stats.zscore(data)
  data_no_outliers = data[(z_scores < 3).all(axis=1)]
  ```

- Data Transformation:
  Transform data if needed, such as converting data types, encoding categorical variables, or scaling numerical features.
  ```
  # Example: Encoding categorical variables
  data = pd.get_dummies(data, columns=['categorical_column'])
  ```

- Feature Engineering:
  Create new features from existing ones to improve model performance.
  ```
  data['new_feature'] = data['feature1'] + data['feature2']
  ```

- Data Splitting:
  Split your dataset into training, validation, and test sets for model development and evaluation.
  ```
  from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X, y,
  test_size=0.2, random_state=42)
  ```

- Scaling and Normalization:
  Scale or normalize numerical features to ensure that they have a similar scale and distribution.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

- Data Analysis:
  Perform the specific analysis you need, such as building machine learning models, statistical analysis, or hypothesis testing.

Conclusion:

In this phase, we have defined the problem, objectives, and outlined the design thinking process for the project. The ultimate goal is to provide meaningful insights that can inform policies and initiatives aimed at improving the lives of Air Q Assessment in Tamil Nadu. Document your entire analysis process, including data sources, methods, and assumptions. This documentation is crucial for transparency and reproducibility.