**Report on**

**Weather data analysis**

**by**

Priyanshu Asthana 202410116100154

Sakshi Khandelwal 202410116100175

Sanya Dwivedi 202410116100184

**Session:2024-2025 (II Semester)**

Under the supervision of

**Ms. Komal Salgotra**

### KIET Group of Institutions, Delhi-NCR, Ghaziabad



### Department Of Computer Applications

**KIET GROUP OF INSTITUTIONS, DELHI-NCR GHAZIABAD-201206**

**Abstract**

Weather data analysis plays a crucial role in understanding climate patterns, predicting weather conditions, and making data-driven decisions across various sectors including agriculture, aviation, transportation, and disaster management. This project focuses on analyzing a comprehensive weather dataset using Python to explore trends, patterns, and correlations in meteorological data. The dataset includes temperature, humidity, wind speed, visibility, pressure, and other parameters collected on an hourly basis throughout the year 2012. The aim is to preprocess the data, handle anomalies, visualize relationships, and derive statistical insights. Through thorough data exploration and analysis, we aim to enhance our understanding of atmospheric behavior and support future predictive modeling.

**Introduction**

In today's data-driven world, the role of data analysis has become increasingly significant across all fields, including meteorology. Weather data analysis is an essential process that helps us interpret and understand the vast amount of meteorological information collected from various sources such as satellites, weather stations, and radar systems. The primary purpose of weather data analysis is to uncover hidden patterns, understand trends, identify anomalies, and derive insights that can be applied to real-world scenarios like agriculture planning, disaster management, infrastructure development, and climate research.

The discipline of meteorology has evolved significantly with the advent of digital technologies and high-volume data collection systems. Modern meteorology not only involves forecasting future weather events but also includes analyzing past and present data to understand long-term climate behavior. In this regard, data analysis serves as the foundation upon which accurate weather predictions and climate models are built. The process of analyzing weather data involves several steps, including data collection, cleaning, processing, visualization, and interpretation. This systematic approach helps meteorologists and researchers derive meaningful insights from raw datasets that may otherwise appear complex and unstructured.

This project is centered around the detailed analysis of a time-series weather dataset collected hourly throughout the year 2012. This dataset contains critical atmospheric variables such as temperature, humidity, wind speed, visibility, and pressure, among others. These variables are not only interrelated but also influenced by numerous external factors, making their analysis a complex yet insightful task. By using Python, one of the most popular programming languages for data science, we can utilize powerful libraries and tools such as Pandas, NumPy, Matplotlib, and Seaborn to manipulate, analyze, and visualize the data effectively.

The importance of understanding weather patterns and trends cannot be overstated. Accurate weather data analysis helps in predicting natural disasters like hurricanes, floods, and droughts, which can save lives and reduce economic losses. In agriculture, understanding seasonal weather trends aids in deciding the right time for sowing, irrigation, and harvesting, thereby improving crop yield and food security. In aviation, weather conditions play a pivotal role in ensuring flight safety and scheduling. In urban planning and infrastructure development, weather analysis helps engineers design resilient systems that can withstand extreme climatic conditions.

**Literature Review/Methodology**

1. **Dataset Overview:** The dataset used in this project is a weather time-series dataset from 2012, obtained from a reliable source. It contains hourly records of various atmospheric parameters such as temperature, humidity, wind speed, visibility, and pressure. The data provides a detailed view of weather patterns across different times of the day and seasons. The structure of the dataset allows for time-series analysis and correlation study among the variables, which is essential for extracting meaningful insights.
2. **Data Preprocessing:** Data preprocessing is a crucial step to prepare raw data for meaningful analysis. This involves several sub-tasks:
   * Handling Missing Values: Missing or null entries are handled using appropriate strategies such as imputation (mean/median/mode), interpolation, or deletion, depending on the nature and extent of the missing data.
   * Date-Time Conversion: The 'date' or 'timestamp' column is converted into a datetime format, enabling time-based indexing and feature extraction.
   * Feature Engineering: Additional columns like 'Hour', 'Day of Week', 'Month', 'Season', and 'Weekday/Weekend' are derived from the datetime column. These features help in temporal pattern analysis.
   * Outlier Detection and Removal: Outliers in temperature, humidity, or other continuous variables are detected using techniques like Z-score or IQR method and are either removed or corrected based on domain knowledge.
   * Normalization/Standardization: For certain analyses, especially if applying machine learning in future phases, numerical values may be normalized or standardized to improve performance.
3. **Exploratory Data Analysis (EDA):** EDA is the heart of any data analysis project. It helps in understanding the underlying patterns and anomalies in the dataset:
   * Summary Statistics: Measures like mean, median, mode, standard deviation, minimum, and maximum values are calculated for each variable.
   * Univariate Analysis: Distributions of individual features are visualized using histograms, bar plots, and box plots.
   * Bivariate and Multivariate Analysis: Relationships between pairs or groups of variables are analyzed using scatter plots, line plots, pair plots, and correlation matrices.
   * Time Series Decomposition: The temperature and humidity data are decomposed into trend, seasonality, and residuals to observe underlying components of the series.
   * Rolling Statistics: Moving averages are computed to smoothen short-term fluctuations and highlight long-term trends.
4. **Statistical Analysis:** Statistical techniques provide a deeper quantitative understanding of the data:
   * Trend Analysis: Observing whether certain variables are increasing, decreasing, or stable over time.
   * Correlation and Covariance: These are computed between all pairs of continuous variables to identify the strength and direction of linear relationships.
   * Seasonality Detection: Seasonal patterns, such as temperature peaking during summer months or humidity increasing during monsoon, are identified.
   * Anomaly Detection: Sudden spikes or drops in values, such as extreme temperature or high wind speeds, are flagged as anomalies.
   * Hypothesis Testing: In some cases, statistical tests like t-tests or ANOVA may be applied to compare weather parameters across seasons or months.
5. **Tools and Libraries Used:** The analysis is performed using Python due to its simplicity, wide community support, and the rich ecosystem of libraries:
   * Pandas: Used for data loading, cleaning, manipulation, and summarization.
   * NumPy: Provides efficient numerical operations and supports array-based computations.
   * Matplotlib and Seaborn: Utilized for generating high-quality, informative visualizations.
   * Jupyter Notebook: Serves as the main environment for running and documenting the analysis with code and narrative combined.
   * SciPy and Statsmodels: Used for applying statistical functions and time-series decomposition.

**Project Objective**

The primary objective of this project is to conduct a comprehensive data analysis on a historical weather dataset using Python, with the goal of uncovering meaningful patterns, trends, and insights that are hidden within the data. This involves cleaning, transforming, visualizing, and interpreting various weather-related variables to develop a deeper understanding of atmospheric behavior over time.

More specifically, the objectives of the project are:

1. **To Understand Weather Trends**: Analyze the seasonal and temporal variations in weather parameters such as temperature, humidity, wind speed, and visibility to understand how these factors fluctuate throughout the year.
2. **To Explore Inter-variable Relationships**: Determine how various atmospheric parameters are correlated with one another. For example, analyze whether there is a strong relationship between temperature and humidity, or between wind speed and visibility.
3. **To Perform Data Preprocessing**: Handle missing values, outliers, and incorrect data types effectively to prepare a clean and consistent dataset suitable for analysis.
4. **To Extract Meaningful Features**: Use feature engineering techniques to derive additional insights such as day/night patterns, weekend effects, or seasonal patterns from the existing data.
5. **To Visualize the Data Effectively**: Use visual tools and plots to identify patterns, detect anomalies, and present findings in a visually appealing and easy-to-understand format.
6. **To Support Climate Awareness and Forecasting**: Provide insights that could help in long-term climate monitoring or form the basis for predictive weather modeling in future studies.
7. **To Enhance Analytical Skills**: Apply real-world data science techniques in Python, thus improving technical and analytical skills for academic and professional development.
8. **To Build a Foundation for Machine Learning Applications**: Although this project focuses on data analysis, it lays the groundwork for applying predictive models and machine learning algorithms in future extensions of the project.
9. **To Contribute to Decision-Making Processes**: Generate insights that may assist stakeholders in fields like agriculture, transport, and public safety by helping them understand environmental patterns.

**Project Flow**

1. The project flow refers to the sequential and logical progression of steps followed throughout the weather data analysis process. A structured workflow ensures clarity, reproducibility, and efficient execution of tasks. Below is a detailed overview of the project flow used in this weather data analysis project:
2. **Problem Definition**: The first step is to clearly define the goal of the project. In this case, the problem statement involves analyzing historical weather data to identify patterns, trends, and relationships between different atmospheric parameters.
3. **Dataset Acquisition**: The dataset is obtained from a publicly available source, such as Kaggle or an official meteorological database. It contains hourly weather data recorded throughout the year 2012. The data includes variables such as temperature, humidity, visibility, pressure, and wind speed.
4. **Setting Up the Environment**: Python is used as the programming language due to its versatility in data analysis. Jupyter Notebook or Kaggle Notebook is selected as the development environment. Necessary libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Statsmodels are imported.
5. **Data Loading and Initial Exploration**: The dataset is loaded into a Pandas Data Frame. A preliminary examination is performed using functions like .head(), .info(), .describe(), and .shape() to understand the structure and contents of the data.
6. **Data Cleaning and Preprocessing**:
7. Missing values are handled using interpolation or imputation methods.
8. Date and time fields are converted to datetime objects for easier time-based analysis.
9. Outliers are identified and addressed through visual inspection or statistical methods.
10. Unnecessary columns, if any, are removed.
11. **Feature Engineering**: New features are derived from existing ones to enhance analytical power:
12. Extracting day, month, year, hour, and day of week from the datetime column.
13. Creating new categorical columns like "Season" or "IsWeekend" to analyze data contextually.
14. **Exploratory Data Analysis (EDA)**:
15. Visualizations such as line plots, histograms, box plots, and heatmaps are created.
16. Statistical summaries and correlation matrices are generated to explore relationships.
17. Temporal trends and seasonal variations are analyzed.
18. **Insights Generation**: Based on visual and statistical analysis, key insights are derived. For instance:
19. Average temperature trend over months.
20. Peak humidity periods.
21. Correlation between temperature and wind speed.
22. **Documentation and Reporting**: All findings, graphs, and interpretations are documented in a structured report or notebook format. Each section includes code, outputs, and explanations for clarity.
23. **Conclusion and Future Scope**: The project concludes with a summary of insights gained and how they could be useful in real-world scenarios. Suggestions for future work, such as applying machine learning models for prediction or expanding the analysis to multiple years, are provided.
24. **Presentation and Sharing**: The final report or notebook is prepared for sharing with peers, instructors, or stakeholders. The documentation is designed to be understandable to both technical and non-technical audiences.

**CODE:**

import pandas as pd

#Reading the csv file

df=pd.read\_csv('1. Weather Data.csv')

df

print(df.shape)

#Q1: To check if there are null values and drop them

print(df.isnull().sum())

df.dropna() #drop any row with NaN

df.dropna(axis=1) #drop any column with NaN

Date/Time 0

Temp\_C 0

Dew Point Temp\_C 0

Rel Hum\_% 0

Wind Speed\_km/h 0

Visibility\_km 0

Press\_kPa 0

Weather 0

dtype: int64

#Q2: To find unique instances of weather types

weather=df['Weather'].value\_counts()

dfweather=pd.DataFrame(weather)

dfweather = dfweather.reset\_index()

dfweather.columns = ['Weather', 'Frequency'] # change column names

dfweather

#Q3: To rename column named 'Weather' to Weather Condition'

df.rename(columns = {'Weather' : 'Weather\_Condition'}, inplace=False)

#Q4: To find all records from data of when the weather was exactly clear

df[df['Weather\_Condition'] == 'Clear']

#Q5: To find the mean temperature, wind speed and visibility

print(df['Temp\_C'].mean())

print(df['Wind Speed\_km/h'].mean())

print(df['Visibility\_km'].mean())

8.79814435336981

14.94546903460838

27.66444672131151

#Q6: To find the variance of pressure

print(df['Press\_kPa'].var())

0.7123440111838408

#Q7: To find the days on which wind speed was less than or equal to 30 km/hr and temperature was greater than 0 C

windspeed=df[(df['Wind Speed\_km/h'] <=30) & (df['Temp\_C']>0)]

print(windspeed['Date/Time'])

print(windspeed.shape)

13 1/1/2012 13:00

14 1/1/2012 14:00

15 1/1/2012 15:00

16 1/1/2012 16:00

17 1/1/2012 17:00

...

8545 12/22/2012 1:00

8546 12/22/2012 2:00

8547 12/22/2012 3:00

8779 12/31/2012 19:00

8780 12/31/2012 20:00

Name: Date/Time, Length: 6302, dtype: object

(6302, 8)

#Q8: To find the date and temperatures for all instances when snow was recorded

snow=df[df['Weather\_Condition'] == 'Snow']

snow.loc[:, ['Date/Time', 'Temp\_C']]

import matplotlib.pyplot as plt

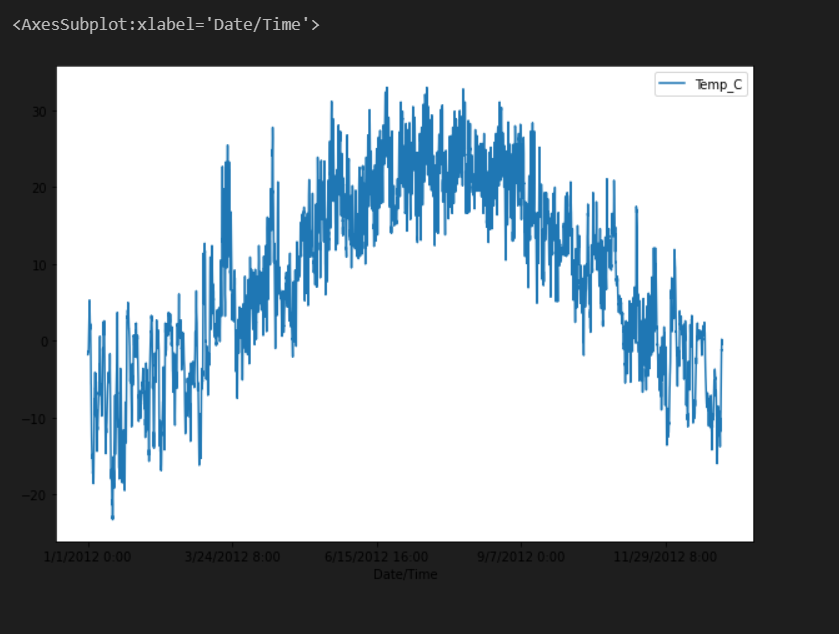
import numpy as np

%matplotlib inline

#Q9: To display a graph of variation of temperature with respect to time

graph=df[['Temp\_C', 'Date/Time']]

graph.plot(x='Date/Time', y='Temp\_C',figsize=(10,7) )



#Q10: To display a pie chart of percentages of weather conditions

#pie chart

keep=dfweather[dfweather['Frequency']>20]

print(keep)

keep.plot.pie(y='Frequency',autopct='%1.1f%%', shadow=False, figsize=(12,9))

plt.title("Weather Conditions")

plt.show()

Weather Frequency

0 Mainly Clear 2106

1 Mostly Cloudy 2069

2 Cloudy 1728

3 Clear 1326

4 Snow 390

5 Rain 306

6 Rain Showers 188

7 Fog 150

8 Rain,Fog 116

9 Drizzle,Fog 80

10 Snow Showers 60

11 Drizzle 41

12 Snow,Fog 37

