Data Science Tools Report

Phase 1

Prepared by

KATAIEF TEAM

Tools and Technologies

Programming Language

- Python – main language for data processing, modeling, and GUI development

---

Libraries for Data Extraction and Cleaning

- `requests` – API calls
- `pandas` – data manipulation
- `re` – regular expressions for cleaning text and formatting dates

---

Data Analysis and Modeling

- `numpy` – numerical operations
- `xgboost` – regression modeling
- `statsmodels` – ARIMA/SARIMA for time series
- `prophet` – long-term forecasting

Clustering and Graph Modeling

- `networkx` – graph-based analysis and visualization of weather relationships
- `scikit-learn` – classification (KNN), clustering (KMeans/DBSCAN)

Visualization

- `plotly` – main tool for all interactive and animated plots
    - Animated line plots, bar charts, and maps
    - Interactive heatmaps and time series charts
- `matplotlib`, `seaborn` – for quick static charts or cross-checks
- `plotly` will also be used inside the GUI to display visualizations dynamically

---

GUI Development

- `Flask` – for creating the main application interface supported by front-end
- `Plotly` + `Flask` – embedded plots using Plotly inside the GUI to allow:
  - Real-time chart updates
  - Animated visualizations
  - Interactive filtering and tooltips

Database

- `MongoDB` – to store cleaned and processed weather data
- Flexible and supports fast access to region-based queries

Development Environment

- **Jupyter** Notebook – for data exploration and model testing
- VS Code – PyCharm Ultimate Edition – for full project development, app integration, and database connection

---

Planned Workflow

Step 1: Data Extraction

- Extract historical and forecasted weather data from Open-Meteo API.
- Collect temperature, rainfall, humidity, wind speed, and sea level data.
- Target multiple Egyptian cities to enable regional analysis.

Step 2: Data Cleaning and Preprocessing

- Clean and structure raw data using pandas and regex.
- Normalize timestamps, remove duplicates, and handle missing values.
- Prepare the dataset for modeling, analysis, and visualization

---

Step 3: Feature Engineering

- Create new features:Time-based features (month, day, season)
- Location labels (e.g., Coastal, Desert, Delta)
- Calculated features like daily temperature range
- These features will enhance the performance of both prediction and clustering models.

Step 4: Clustering and Network Analysis

- Apply clustering algorithms (e.g., K-Means, DBSCAN) to group cities with similar weather behavior.
- Use `NetworkX` to analyze relationships between variables as a graph structure.
- This helps discover underlying patterns and visualize connections between temperature, rainfall, humidity, and sea level.

Step 5: Forecasting Models

- We will use time series forecasting models including:
- Prophet – for long-term seasonal trends
- `ARIMA` / `SARIMA` – for short-term accurate predictions
- These models will be used to forecast:
- Temperature trends
- Rainfall levels
- Sea level changes

Step 6: Classification Models

- We'll also use classification models such as K-Nearest Neighbors (KNN).
- Example: classify days as "**Safe**", "**High Risk**", or "**Extreme**" based on weather conditions.
- This helps simulate alerts or climate-based decisions.

Step 7: Time Series Visualization

- We will create time series plots for:
- Temperature over time
- Rainfall trends over **5 years**
- These will help show seasonal patterns, climate shifts, and weather behavior.

Step 8: Animated Visualizations (Plotly)

- Animated line plot: Temperature changes over time across cities
- Animated bar chart: Yearly rainfall comparison per city
- Animated map: Geographical weather evolution (e.g., temperature shift across Egypt)
  This will make data insights interactive and easy to understand.

Step 9: GUI Development

We will build a graphical desktop application using:

- **Flask** for the main Back-end and Front-end for interface
- **Plotly** for all embedded charts and visual components

Features?

- Choose city and date
- Display forecasted temperature and rainfall
- View time series graphs, animated charts, and maps
- All visuals will be interactive Plotly elements, shown in-app

---

Step 10: Data Storage

- Store final datasets in **MongoDB**
- Organize by location and date for future querying or updates
- Enables reuse of processed data across sessions and modules

---

Data to Be Extracted

From Open-Meteo, we will collect:

- Daily temperature (min, max, average)
- Rainfall and precipitation levels
- Humidity levels
- Wind speed and direction
- Sea level
- Location coordinates
- Time series data **(5+ years)**

Project Overview

We're building a project that collects and analyzes weather data from different cities in Egypt.

Our main goal is to predict temperature and understand how climate variables are affecting our environment. This project includes data extraction, cleaning, statistical analysis, machine learning, and visualization.

We're especially focused on revealing patterns in temperature, rainfall, sea levels, and their possible environmental impacts.

Website Description

We're using Open-Meteo.com, a free and powerful weather API. It provides structured, location-based weather data including:

- Temperature
- Humidity
- Wind speed
- Precipitation
- Sea level
- Forecast and historical data

The site allows city-level access across Egypt, making it ideal for region-specific analysis.

Main Project Ideas

Idea 1: Temperature Forecasting in Egypt

We'll build a **predictive model** that estimates temperature for any Egyptian city based on date and location.

We'll also compare temperature changes between different cities and historical periods to track long-term climate trends.

Idea 2: Weather and Soil Fertility

We'll study how temperature, humidity, and wind patterns relate to soil health.

The aim is to identify possible indicators of desertification and how certain weather shifts may lead Egypt toward more arid zones.

Idea 3: Climate Change and Sea Level Rise

We'll analyze the relationship between rising temperatures, other weather factors, and sea level changes.

The goal is to estimate flooding risk for coastal cities like Alexandria and track when this risk might become critical.

Thanks for All My Team

Kataief_Team