Using machine learning to rainfall prediction and flood risk forecast

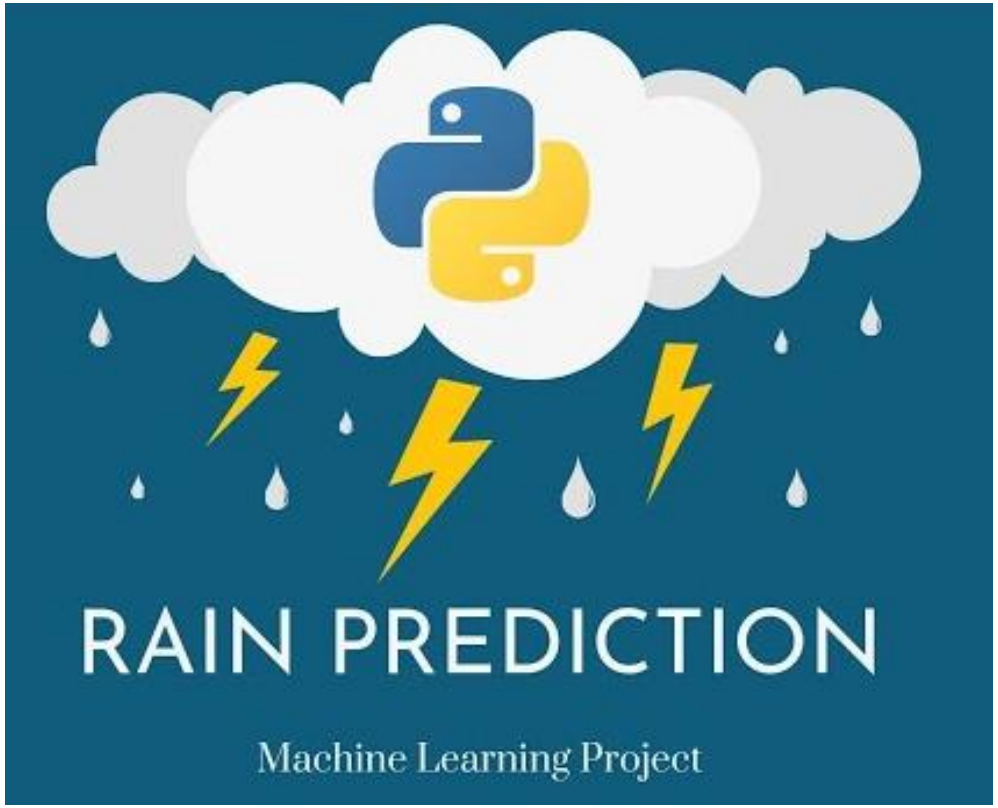
Kalybek Aruzhan

Mathematical foundations of intelligent systems

Course Work

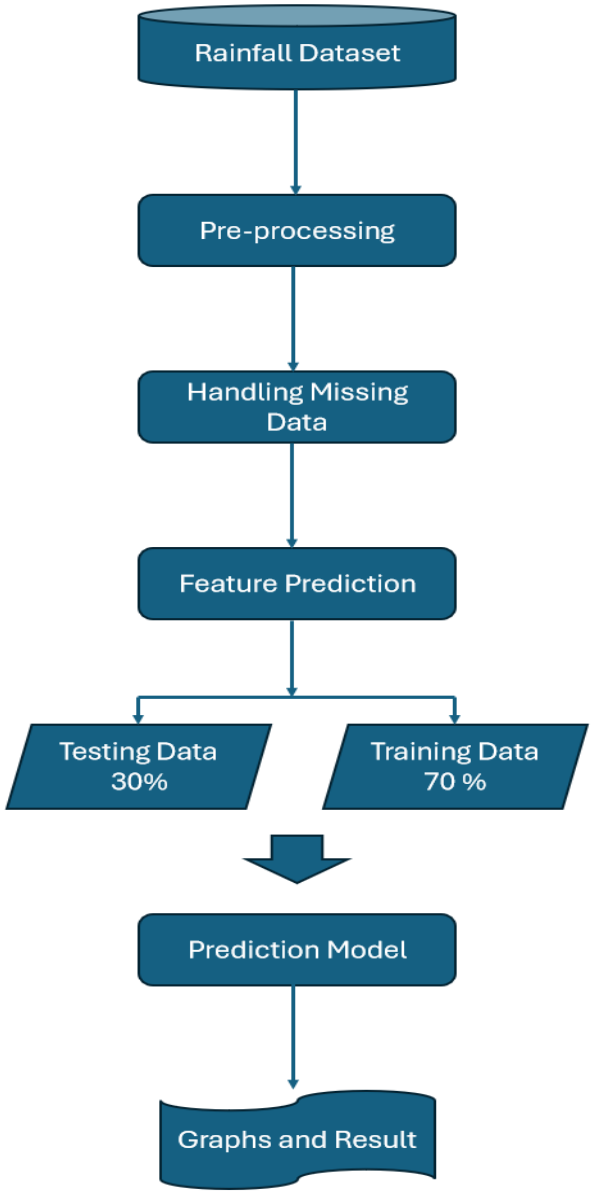
Introduction

This project leverages machine learning to analyze meteorological data, predict rainfall, and assess flood risks. By integrating data like rainfall measurements and historical floods, it builds a robust model using regression analysis, gradient descent, and performance metrics like accuracy and mean squared error. The work supports climate resilience, environmental management, and disaster preparedness, offering practical solutions for sustainable development through predictive analytics and user-friendly tools.



Machine Learning model

1.The process begins with a dataset of historical rainfall data, including time-series and meteorological parameters.
2.During pre-processing, the data is cleaned, normalized, and outliers are removed to improve quality.
3.Missing values are handled using techniques like mean imputation, interpolation, or removing incomplete rows.
4.Principal Component Analysis reduces the number of features while preserving critical information.
5.The dataset is split into training (70%) and testing (30%) subsets for model development and evaluation.
6.Machine learning algorithms are applied to predict rainfall volumes based on the input data.
7.Performance is assessed through graphs and statistical metrics to evaluate prediction accuracy and efficiency.



Dataset

The dataset contains daily meteorological data from 20 major U.S. cities for 2024–2025, offering a rich foundation for predictive modeling and weather trend analysis. Its diverse features enable various applications, such as training machine learning algorithms to predict rain, identifying weather trends across cities, and exploring relationships between variables like humidity, temperature, and precipitation.

Date	Location	Temperature	Humidity	Wind speed	Precipitation	Cloud Cover	Pressure	Water Level In Rivers	Rain tomorrow	Flood Risk
01.01.2024	New York	87.5	75.6	28.3	0.0	69.6	1026.03	4,3	0	0
02.01.2024	New York	83.2	28.7	12.4	0.5	41.6	995.96	5,5	0	0
03.01.2024	New York	80.9	64.7	14.1	0.9	77.3	980.7	7,5	1	1
...	...	...	...	...	...	...	...	...	...	...
20.06.2025	Chicago	49.9	23.1	1.5	0.5	94.2	987.3	2,6	0	0

Data Description

Feature	Description
Temperature	The average daily temperature, which is crucial for analyzing seasonal and regional climatic conditions.
Humidity	The measurement of moisture content in the air, vital for evaluating precipitation probabilities and atmospheric comfort levels.
Wind Speed	A parameter that influences weather formation and can indicate extreme conditions.
Precipitation	The amount of rainfall recorded for a given day, a fundamental feature for analyzing rainfall patterns and intensities.
Cloud Cover	The extent of cloudiness, an important factor in forecasting sunlight exposure and predicting weather phenomena.
Pressure	A key indicator often used to detect weather changes, such as the likelihood of rain or storms.
Water Level In Rivers	A hydrological metric critical for assessing flood risks and monitoring water resource conditions.

Statistical Metrics

Statistical metrics

Features	Mean	Median	Std Dev	Min	Max	IQR
Temperature	65.36	64.95	19.94	30.01	99.96	34.36
Humidity	59.98	60.23	23.33	20.06	99.97	39.90
Precipitation	0.38	0.19	0.47	0	2.63	0.65
Cloud Cover	54.4	53.48	25.74	10.03	99.86	44.07
Pressure	1005.53	1005.41	20.23	970	1039.98	35.12
Wind Speed	14.58	14.22	8.66	0.01	29.99	15.34
Water Level In Rivers	5.09	4.81	2.57	1.06	9.9	4.35

During the data analysis, basic statistical metrics for each variable were obtained, including mean, median, standard deviation, minimum and maximum values, and interquartile range. These measurements help you understand the distribution of data and detect potential outliers. Temperature, humidity, precipitation, clouds, pressure, wind speed, and river water level are all statistically significant factors. Temperature and humidity have a minimal standard deviation, indicating their stability. Simultaneously, precipitation and river water levels exhibit significant fluctuation, as seen by a large standard deviation and interquartile range. This might be owing to their inherent unpredictability.

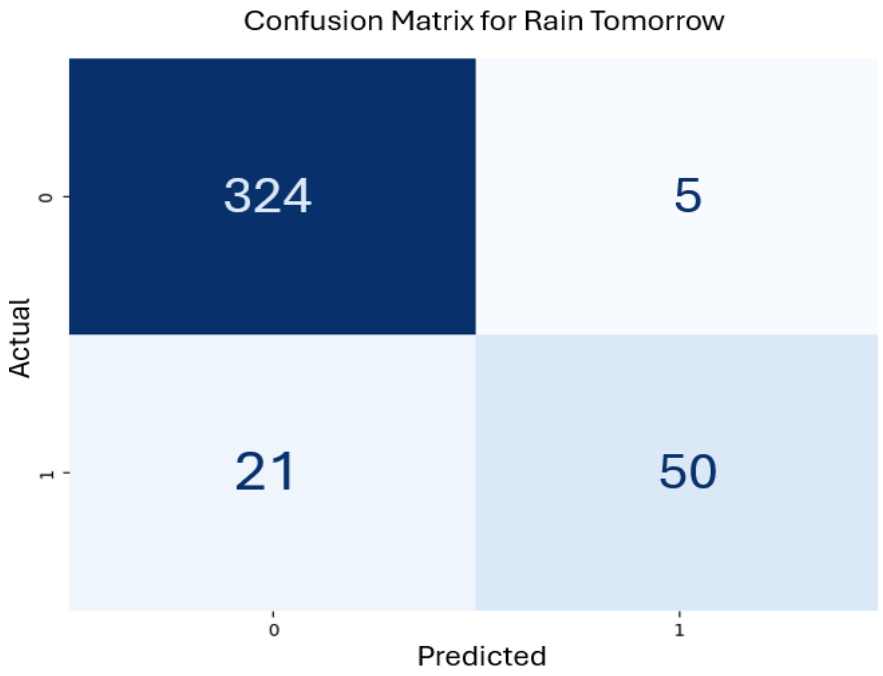
Training and Testing data

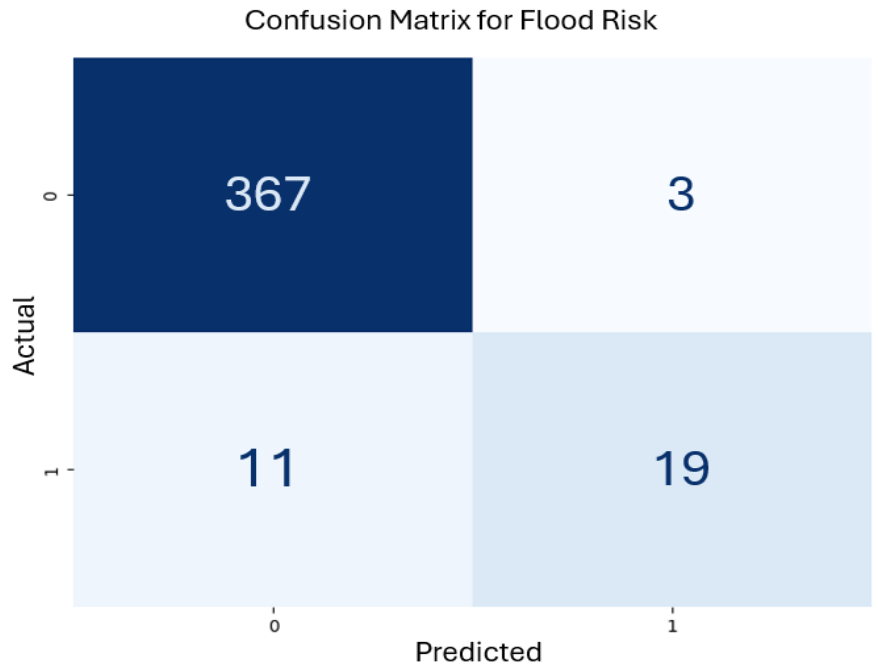
```
from sklearn.model_selection import train_test_split
X = data[['Temperature', 'Humidity', 'Precipitation', 'Cloud Cover', 'Pressure',
'Wind Speed', 'Water Level In Rivers ']]
y_rain = data['Rain Tomorrow']
y_flood = data['Flood Risk']
X_train, X_test, y_train_rain, y_test_rain = train_test_split(X, y_rain,
test_size=0.2, random_state=42)
X_train, X_test, y_train_flood, y_test_flood = train_test_split(X, y_flood,
test_size=0.2, random_state=42)
```

The code snippet splits the dataset into training and testing sets for predicting Rain Tomorrow (y_rain) and Flood Risk (y_flood). Features include temperature, humidity, precipitation, cloud cover, pressure, wind speed, and river water levels. Using train_test_split from sklearn.model_selection, 80% of the data is allocated for training and 20% for testing, with random_state=42 ensuring reproducibility.This process produces training and testing sets for features (X_train, X_test) and targets (y_train_rain, y_test_rain for rain; y_train_flood, y_test_flood for floods). It helps prevent overfitting and provides an accurate evaluation of the model’s performance.

Logistic Regression

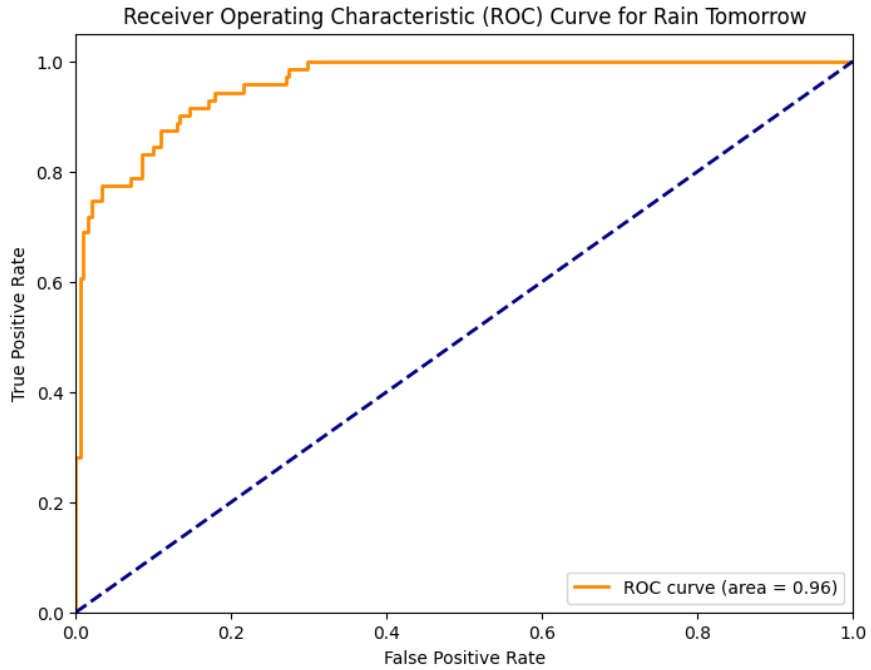
Logistic regression produces probabilities that are limited between 0 and 1, which makes it ideal for binary outcomes like forecasting the likelihood of rain or flooding, in contrast to linear regression, which produces continuous values. Higher accuracy and fewer misclassifications show that the logistic regression model outperforms rain tomorrow in forecasting flood danger. The model's dependability is demonstrated by the low rates of false positives and false negatives seen in both confusion matrices.

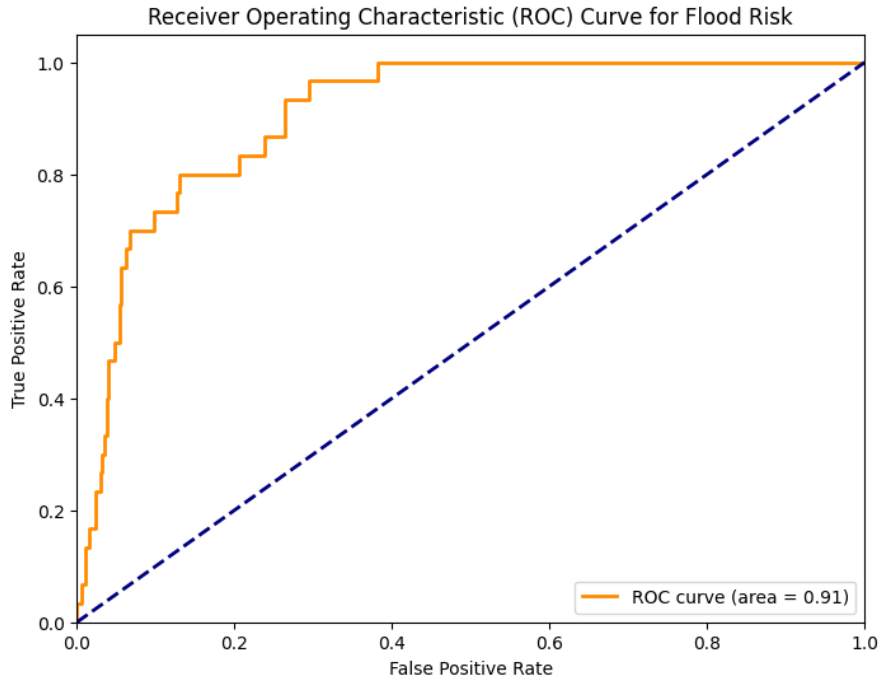




ROC curve

The ROC (Receiver Operating Characteristic Curve) curve is a graph that shows how well a classification model distinguishes between two classes at various thresholds. It lets you visualize the model's performance at all possible thresholds, not just a single one. If the curve is closer to the top-left corner, that's great.





Database

	username character varying (60)	password character varying (60)	address character varying (60)	phone character varying (60)
1	comwsty	8520	Astana	36038

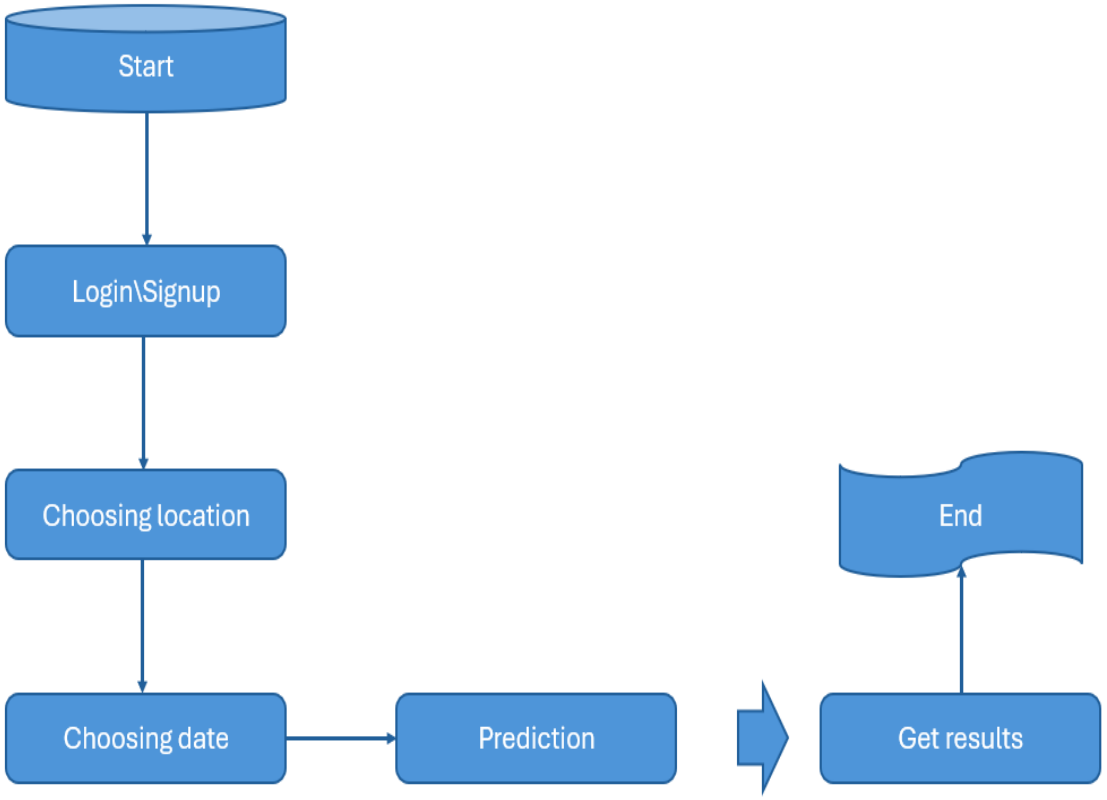
```
create table users (
        username text not null,
        password text not null,
        address text not null
        phone text not null
);
```

The query to create the users table that stores user information.

```
db_name = 'postgres'
db_user = 'postgres'
db_password = '5658'
db_host = 'localhost'
db_port = '5432'
```

The code to connect database with python application.

Application architecture



The application is intended to anticipate precipitation and flood threats based on meteorological data. At launch, the user inputs the city and date, and the program gets the necessary meteorological data from the dataset and filters it based on the criteria entered. Next, using trained machine learning models, the program forecasts if it will rain and whether there is a risk of flooding in the selected city on the stated day. The forecast is shown as a series of meteorological characteristics, including temperature, humidity, wind speed, and others. Finally, the program shows information regarding probable precipitation and flood danger, telling the user about the present meteorological conditions and hazards.

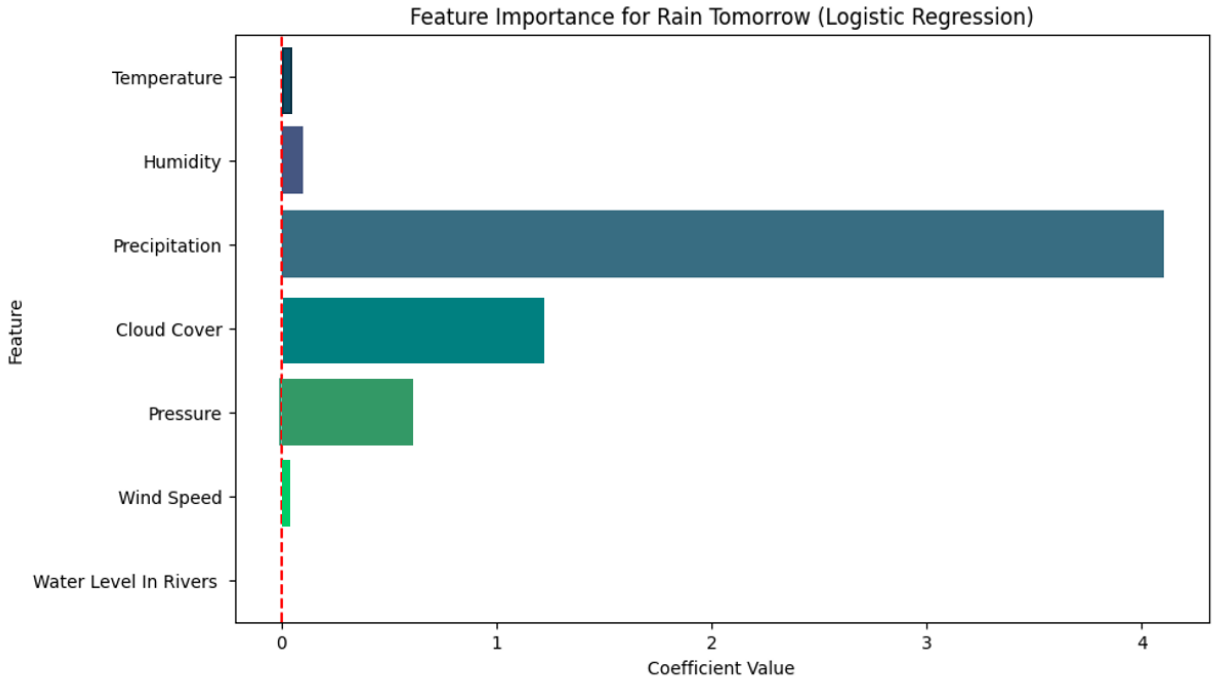
Python Libraries

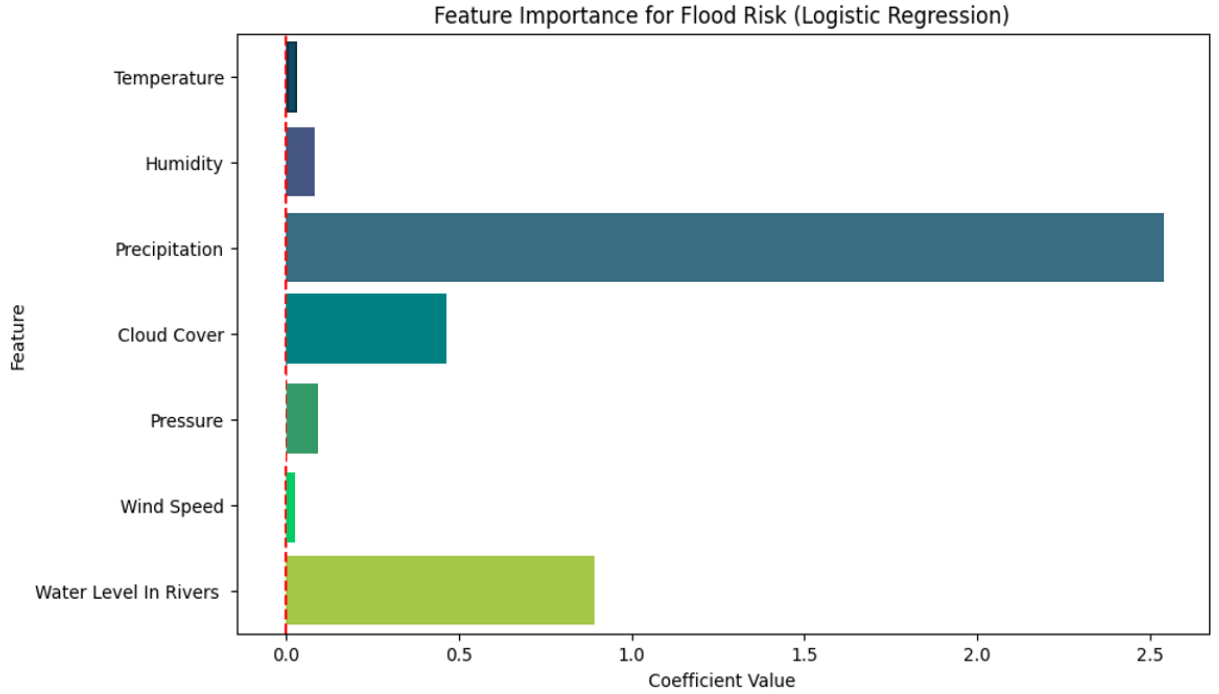
- *Tkinter* is a library for creating graphical user interfaces in Python. This library is used to create the main window, Label for text labels, and other components like messagebox to display messages.
- *psycopg2* is a library for interacting with PostgreSQL. It allows application to connect to a database, retrieve, and update data.
- *pandas* is a library for data manipulation and analysis. It's used to load and manipulate data, such as reading from Excel or CSV files and preparing the data for model training.
- *sklearn (scikit-learn)* is a machine learning library. In the case, it's used:
  - ✓ *LogisticRegression* is an algorithm for classification tasks (e.g., predicting rainfall or flood risks).
  - ✓ *train_test_split* is a function for splitting the data into training and testing sets.
- *StandardScaler* is a tool for data standardization (normalization), which is important for efficient model training.

Results

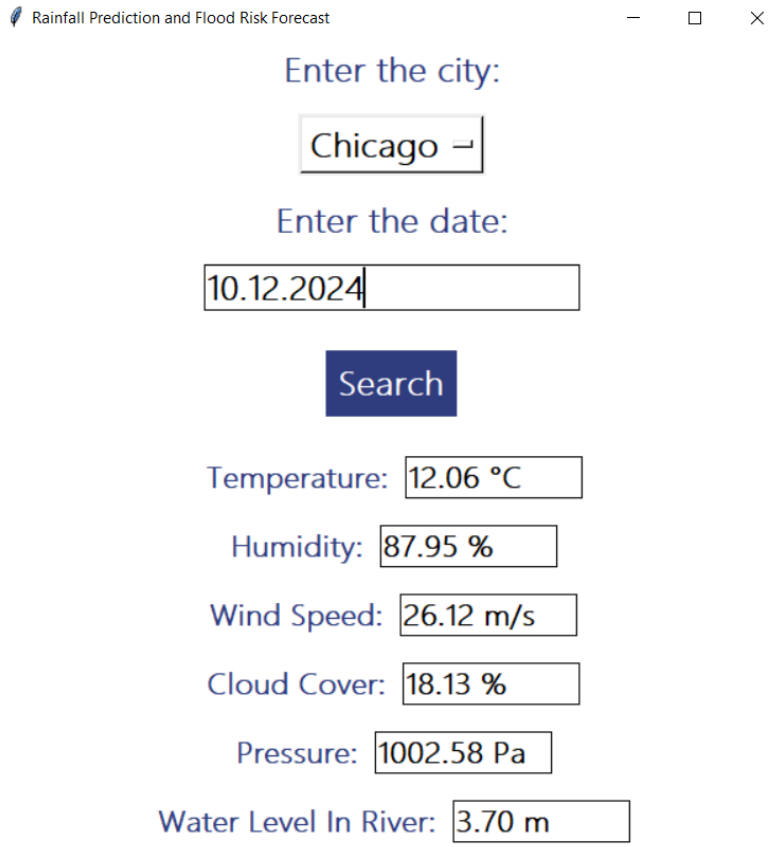
First chart shows that “Precipitation” is the most important factor in predicting rain tomorrow, followed by “Temperature.” “Humidity” and “Cloud Cover” have moderate influence, while “Pressure,” “Wind Speed,” and “Water Level in Rivers” contribute minimally.

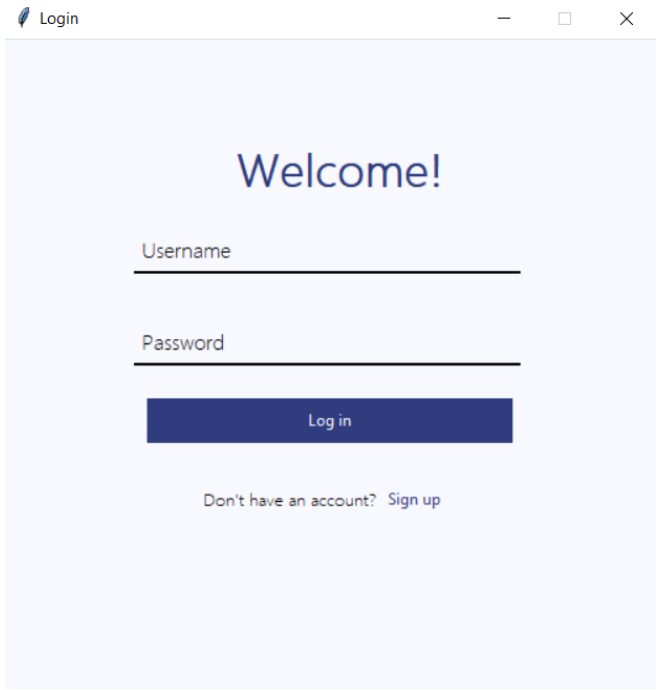
First chart highlights “Precipitation” and “Water Level in Rivers” as key factors in flood risk prediction. “Temperature” and “Humidity” have minor roles, while “Cloud Cover,” “Pressure,” and “Wind Speed” have little impact.

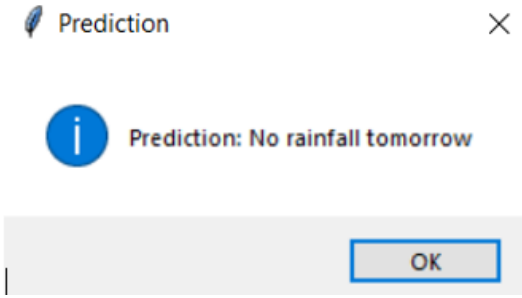


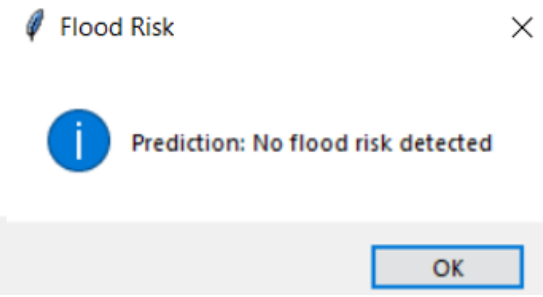


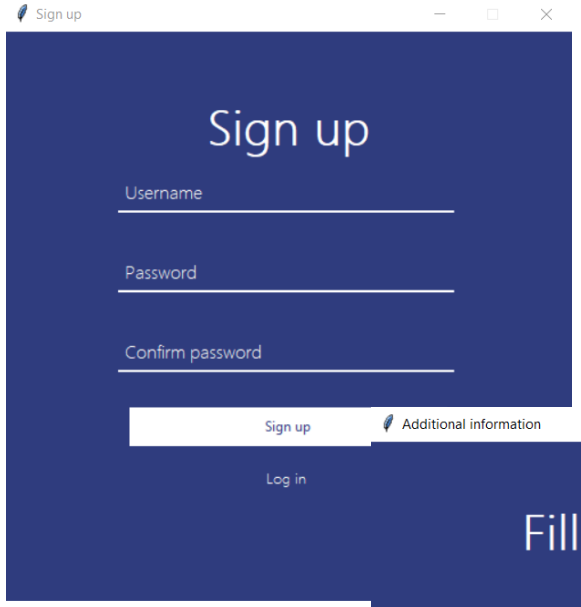
Results

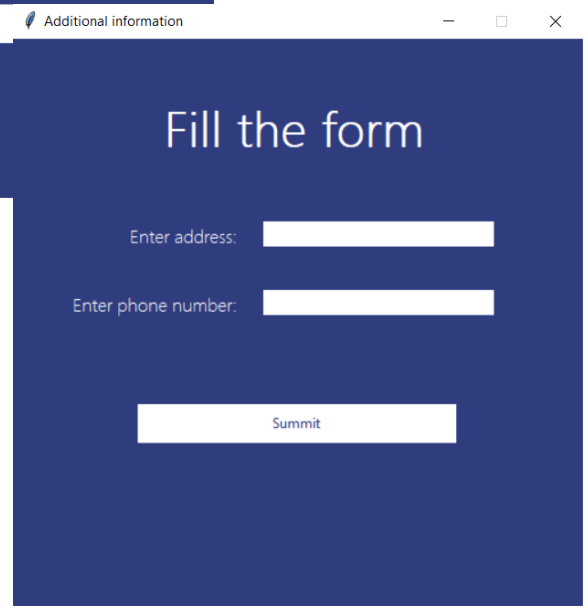

Rainfall Prediction and Flood Risk Forecast
Enter the city:
Chicago
Enter the date:
10.12.2024
Search
Temperature: 12.06 °C
Humidity: 87.95 %
Wind Speed: 26.12 m/s
Cloud Cover: 18.13 %
Pressure: 1002.58 Pa
Water Level In River: 3.70 m


Login
Welcome!
Username
Password
Log in
Don't have an account? Sign up

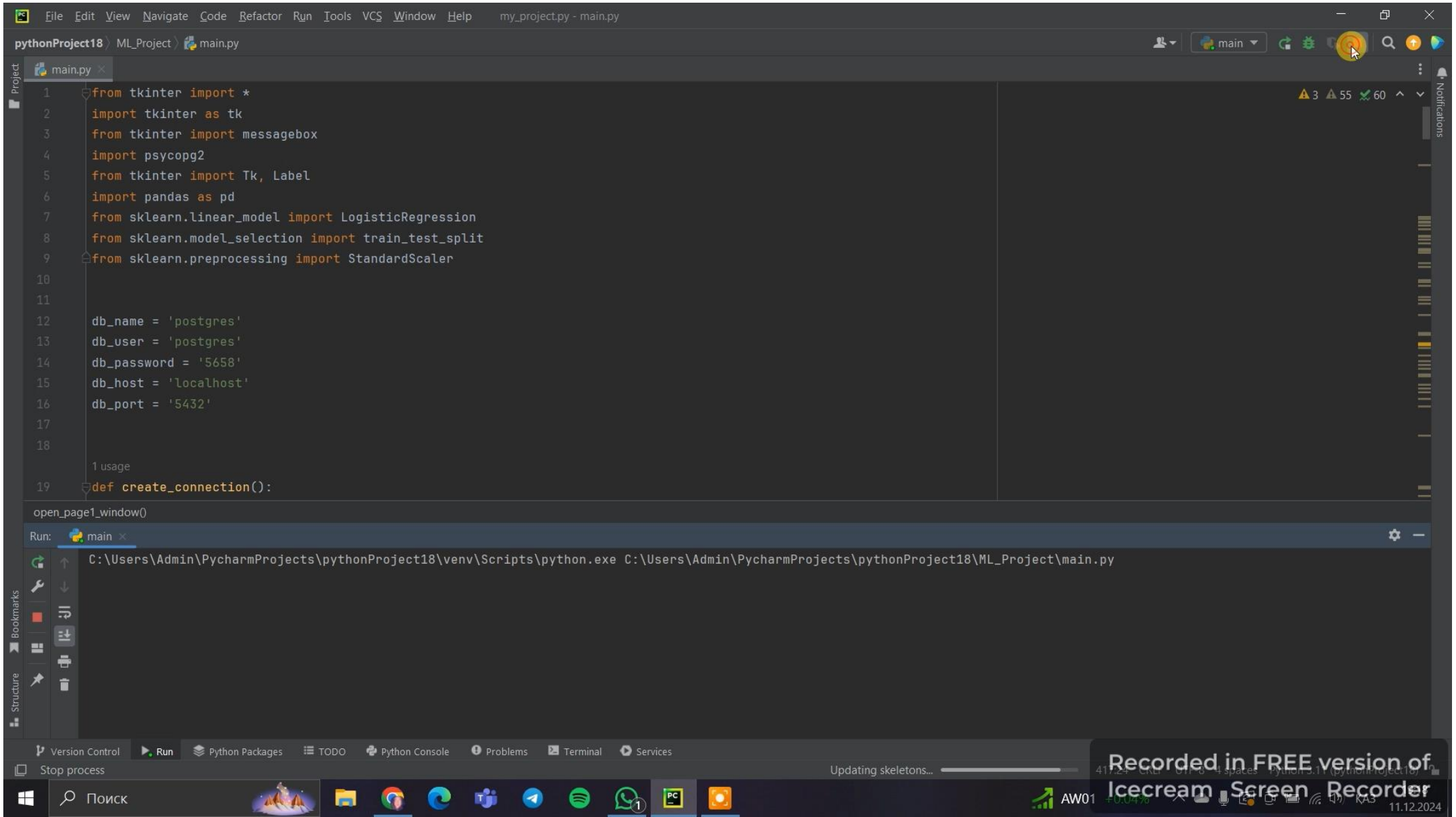

Prediction
Prediction: No rainfall tomorrow
OK


Flood Risk
Prediction: No flood risk detected
OK


Sign up
Sign up
Username
Password
Confirm password
Sign up
Log in


Additional information
Fill the form
Enter address:
Enter phone number:
Summit

Teaser


File Edit View Navigate Code Refactor Run Tools VCS Window Help
my_project.py - main.py
pythonProject18 ML_Project main.py
main.py
from tkinter import *
import tkinter as tk
from tkinter import messagebox
import psycopg2
from tkinter import Tk, Label
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
db_name = 'postgres'
db_user = 'postgres'
db_password = '5658'
db_host = 'localhost'
db_port = '5432'
1 usage
def create_connection():
open_page1_window()
Run: main
C:\Users\Admin\PycharmProjects\pythonProject18\venv\Scripts\python.exe C:\Users\Admin\PycharmProjects\pythonProject18\ML_Project\main.py
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
Stop process
Updating skeletons...
Recorded in FREE version of Icecream Screen Recorder
Поиск
AW01
11.12.2024

Conclusion

This project applies machine learning to predict rainfall and assess flood risks, tackling a critical issue amplified by climate change. Using meteorological data, geographical features, and historical flood records, it delivers a robust tool for disaster preparedness and resource management. The system combines advanced machine learning, mathematical models, and performance metrics for accuracy and usability. It also features an SQL database for user data management and a Python-based interface for seamless interaction, ensuring practicality and accessibility. This work demonstrates the potential of AI in addressing climate challenges and promoting sustainable development. Future enhancements could include incorporating additional data sources, advanced algorithms, and real-time forecasting to improve accuracy and adaptability, further supporting global flood risk management.