AI WELL WATER PREDICTION SYSTEM

CHAPTER 1

AIM :

The goal of this project is to create a smart system that uses artificial intelligence (AI) to predict the amount of water available in wells and check whether the water quality is potable or not. This system will help people know how much water they can expect from their wells at different times. By doing so, it makes managing water resources easier and more efficient, ensuring that people have a reliable supply of water when they need it.Use AI algorithms to analyze historical data and current conditions to predict future water levels in wells with high accuracy. This helps in anticipating changes and planning accordingly.

CHAPTER 2

INTRODUCTION:

The AI Well Water Prediction System is a cutting-edge solution designed to forecast water levels in wells with high precision using artificial intelligence. This system integrates historical water level data, current environmental conditions, and predictive analytics to deliver accurate forecasts about the availability of water. By leveraging machine learning algorithms, the system processes a variety of inputs—such as rainfall, soil moisture, and previous water usage patterns—to predict future well water levels.The system features a user-friendly interface that allows users to easily input well data and receive actionable insights. It also includes an alert mechanism to notify users of potential water shortages, enabling proactive resource management. The AI Well Water Prediction System aims to enhance water management practices, reduce operational costs, and support sustainable water use by providing reliable predictions and early warnings.

CHAPTER 3

ABSTRACT:

The AI Well Water Prediction System is an advanced technological solution designed to forecast the availability and quality of groundwater resources. This system leverages artificial intelligence (AI) and machine learning techniques to analyze various data inputs, such as historical water levels, geological surveys, weather patterns, and usage trends, to provide accurate predictions about water availability and sustainability.

CHAPTER 4

EXISTING SYSTEM:

An AI Well Water Prediction System leverages machine learning and data analysis to forecast groundwater availability and quality. It integrates various data sources, such as geological, hydrological, climatic, and topographic information, to train predictive models. These models analyze historical and real-time data to estimate groundwater levels and assess water quality. The system typically features a user-friendly interface for accessing predictions and visualizing results on maps, aiding in decision-making for agricultural, urban planning, and water resource management.

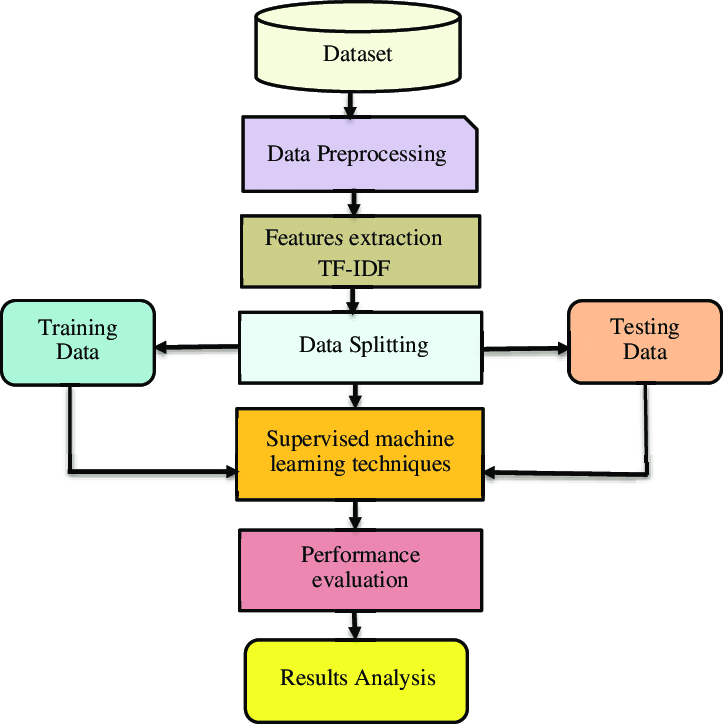
CHAPTER 5

PROPOSED SYSTEM:

The AI Well Water Prediction System is a cutting-edge solution designed to forecast groundwater levels and quality using artificial intelligence. This system integrates multiple data sources, including historical water levels, geological surveys, weather patterns, and water usage trends. By employing advanced machine learning algorithms, it processes and analyzes these datasets to generate accurate predictions of future water availability and potential issues.The system’s core components include data collection, where it gathers and preprocesses relevant information; predictive modeling, which utilizes AI techniques to forecast water levels; and anomaly detection, which identifies unusual patterns that may indicate problems like over-extraction or contamination. It also features geospatial analysis for mapping water data and scenario simulation to evaluate different conditions.

CHAPTER 6

ARCHITECTURE:



CHAPTER 7

SYSTEM MODULE:

MODULE 1:

DATA COLLECTION:

* **Historical Data:** Records of past water levels, rainfall, and groundwater usage.
* **Geological Data:** Information about soil types, rock formations, and aquifer characteristics.
* **Weather Data:** Forecasts and historical weather patterns, including precipitation and temperature.
* **Usage Data:** Patterns of water extraction and consumption.

MODULE 2:

DATA PRE PROCESSING:

Data preprocessing is a crucial step in the AI Well Water Prediction System, ensuring the accuracy and reliability of the predictions generated by the system

**Preprocessing:** Cleaning and normalizing data to ensure accuracy and consistency.

**Feature Extraction:** Identifying and selecting relevant features that impact groundwater levels.

**Training and Testing Sets:** Dividing the preprocessed data into training and testing subsets to evaluate the model’s performance. The training set is used to build the model, while the testing set assesses its accuracy and generalizability.

MODULE 3:

EDA

Exploratory Data Analysis (EDA) is a crucial step in the development of an AI-based well water prediction system. It involves examining and visualizing the dataset to understand its structure, detect patterns, spot anomalies, and gain insights that inform subsequent modeling efforts. Use these to understand the distribution of individual features such as water level and pH. It helps in identifying skewness, kurtosis, and potential outliers.Visualize trends over time for variables like water levels, precipitation, and contamination levels. Seasonal patterns or long-term trends may be evident.

MODULE 4:

MODEL IMPLEMENT

Implementing a predictive model for well water in an AI system involves several steps, from selecting the right model to training, evaluation, and deployment.

**Regression Models:** Linear Regression, Ridge/Lasso Regression, Random Forest Regressor, Gradient Boosting Machines (GBM), or Neural Networks for predicting continuous variables like water level.

**Classification Models:** Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), or Neural Networks for classifying water quality.

MODULE 5:

PREDICTION

Prediction in the context of a well water AI system involves using the trained model to make forecasts about future well water levels, quality, or other relevant metrics based on current and historical data.

If can check whether the water sample is potable or not.If it is condition is satisfied,the water sample is potable.If it is not,water sample is not potable.

**Load Model:** Load the trained model from its saved state. This could be a serialized model file (e.g., a .pkl file for scikit-learn models or a .h5 file for TensorFlow models).

**Generate Predictions:** Pass the prepared input data through the model to generate predictions. Depending on the problem, the output could be:

* **Continuous Values:** For regression models, such as predicting water level or contaminant concentration.
* **Class Labels:** For classification models, such as predicting whether the water is safe or unsafe.
* **Probabilities:** For models that output probabilities, such as the likelihood of a certain contamination level being present.

CHAPTER 8

SOURCE CODE:

index.html

<!doctype html>

<html>

<head>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static',filename='styles/stylesheet.css') }}">

<!--for bg image in html file-->

<!--background-image:url({{ url\_for('static', filename='img/img1.jpg') }});-->

<title>Water Quality</title>

</head>

<body>

<div class="name">

Water Quality

</div>

<div class="context">

{% if result %}

Given water sample is {{ result }}

{% else %}

"Access to safe drinking-water is essential to health, a basic human right and a component of effective policy for health protection. This is important as a health and development issue at a national, regional and local level. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions."

{% endif %}

</div>

<div class="project">

<form method="post" action="/" enctype = "multipart/form-data">

<h1 > Potability Checker</h1>

<input type="text" name="ph value" id="ph value" placeholder="ph value(0 to 14)" required>

<input type="text" name="Hardness" id="Hardness" placeholder="Hardness(mg/L)" required>

<input type="text" name="Solids" id="Solids" placeholder="Solids(ppm)" required>

<input type="text" name="Chloramines" id="Chloramines" placeholder="Chloramines(ppm)" required>

<input type="text" name="Sulfate" id="Sulfate" placeholder="Sulfate(mg/L)" required>

<input type="text" name="Conductivity" id="Conductivity" placeholder="Conductivity(uS/cm)" required>

<input type="text" name="Organic carbon" id="Organic carbon" placeholder="Organic carbon(ppm)" required>

<input type="text" name="Trihalomethanes" id="Trihalomethanes" placeholder="Trihalomethanes(ug/L)" required>

<input type="text" name="Turbidity" id="Turbidity" placeholder="Turbidity(NTU)" required>

<center><input class="btn" type=submit name="submit\_button" value=PREDICT></center>

</form>

</div>

</body>

</html>

app.py

import pickle

from flask import Flask, render\_template, request

import numpy as np

import os

app = Flask(\_\_name\_\_)

@app.route('/', methods=['GET', 'POST'])

def home():

if request.method == "POST":

# request all the input fields

ph = float(request.form['ph value'])

Hardness = float(request.form['Hardness'])

Solids = float(request.form['Solids'])

Chloramines = float(request.form['Chloramines'])

Sulfate = float(request.form['Sulfate'])

Conductivity = float(request.form['Conductivity'])

Organic\_carbon = float(request.form['Organic carbon'])

Trihalomethanes = float(request.form['Trihalomethanes'])

Turbidity = float(request.form['Turbidity'])

# create numpy array for all the inputs

val = np.array([ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic\_carbon, Trihalomethanes, Turbidity])

# define save model and scaler path

model\_path = os.path.join('models', 'xgboost.sav')

scaler\_path = os.path.join('models', 'scaler.sav')

# load the model and scaler

model = pickle.load(open(model\_path, 'rb'))

scc = pickle.load(open(scaler\_path, 'rb'))

# transform the input data using pre fitted standard scaler

data = scc.transform([val])

# make a prediction for the given data

res = model.predict(data)

if res == 1:

outcome = 'Potable'

else:

outcome = 'not potable'

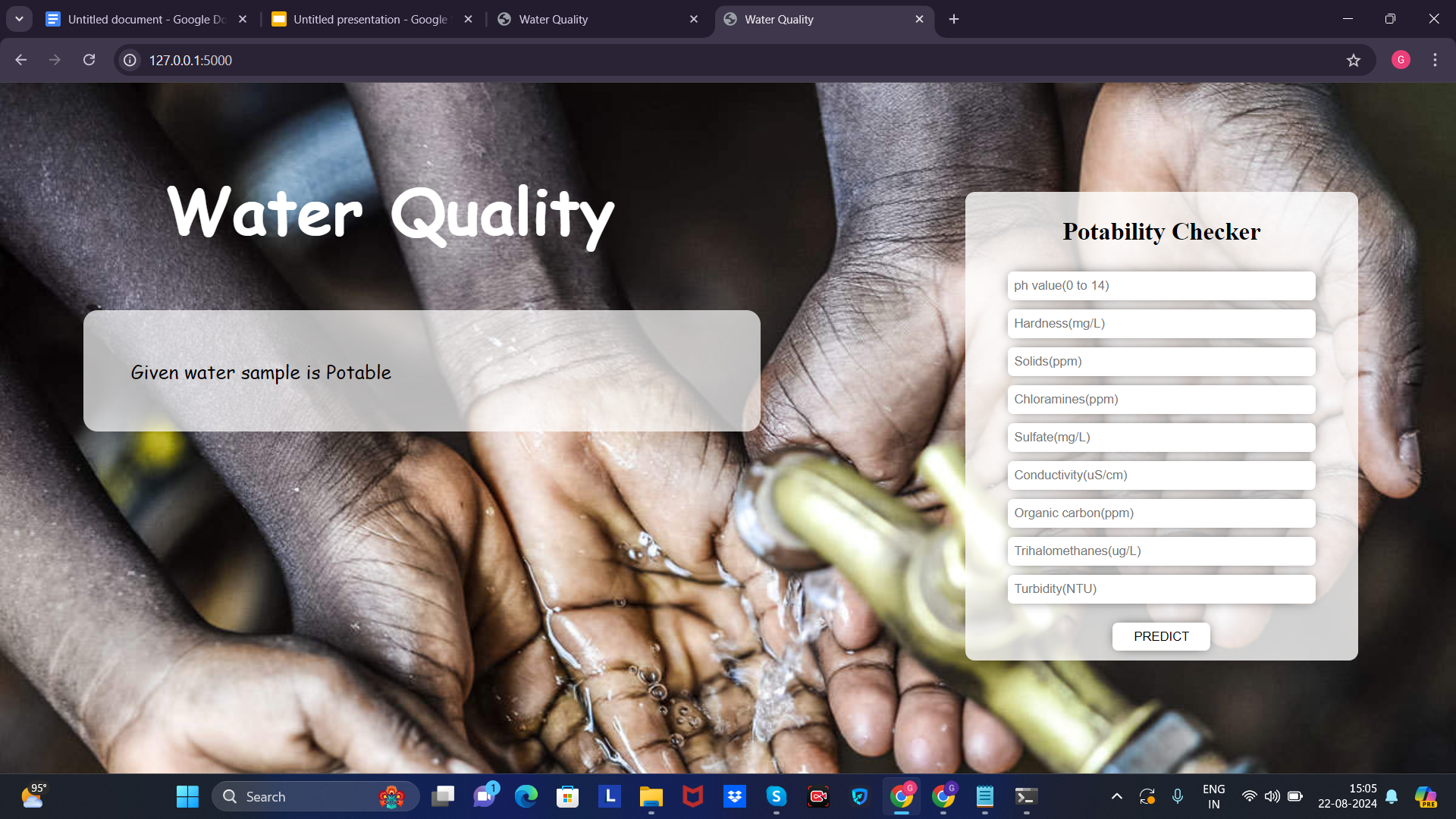
return render\_template('index.html', result=outcome)

return render\_template('index.html')

# run application

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

OUTPUT:

CHAPTER 9

CONCLUSION:

The development and implementation of an AI-based well water prediction system involve multiple stages, each critical to the overall success of the project. The process starts with thorough **Exploratory Data Analysis (EDA)** to understand the data and identify key patterns. This is followed by selecting appropriate models and training them to accurately predict outcomes such as water levels and quality metrics.

Once the model is trained, the **prediction phase** involves using the model to forecast future well water conditions based on current and historical data. This process includes preprocessing data, generating predictions, and interpreting the results in a meaningful way.

CHAPTER 10

REFERENCE:

[1] P. Zeilhofer, L. V. A. C. Zeilhofer, E. L. Hardoim, Z. M. . Lima, and C. S. Oliveira, “GIS applications for mapping and spatial modeling of urban-use water quality: a case study in District of Cuiabá, Mato Grosso, Brazil,” Cadernos de Saúde Pública, vol. 23, no. 4, pp. 875–884, 2007. [2] M. A. Kahlown, M. A. Tahir, and H. Rasheed, National Water Quality Monitoring Programme, Fifth Monitoring Report (2005–2006), Pakistan Council of Research in Water Resources Islamabad, Islamabad, Pakistan, 2007, http://www.pcrwr.gov .pk/Publications/Water%20Quality%20Reports/Water% 20Quality%20Monitoring%20Report%202005-06.pdf. [3] UN water, “Clean water for a healthy world,” Development, 2010, https://www.undp.org/content/undp/en/home/ presscenter/articles/2010/03/22/clean-water-for-a-healthyworld.html

[4] K. Farrell-Poe, W. Payne, and R. Emanuel, Water Quality & Monitoring, University of Arizona Repository, 2000, http:// hdl.handle.net/10150/146901. [5] T. Taskaya-Temizel and M. C. Casey, “A comparative study of autoregressive neural network hybrids,” Neural Networks, vol. 18, no. 5–6, pp. 781–789, 2005.