Water Quality Test Prediction for Concrete Mixing

**1.Aim**

To develop a machine learning model for real-time prediction of water quality for concrete mixing, ensuring compliance with quality standards and reducing structural risks.

**2.Motivation**

* Poor water quality leads to weak and less durable concrete, increasing maintenance costs.
* Manual testing is time-consuming and prone to errors.
* ML can automate water quality assessment, improving efficiency and reliability.

**3.Exploratory Data Analysis (EDA) – Code**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

df = pd.read\_csv("water\_quality\_dataset.csv")

# Basic info

print(df.info())

print(df.describe())

# Check for missing values

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

# Correlation heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(df.corr(), annot=True, cmap="coolwarm")

plt.show()

# Class distribution

sns.countplot(x=df['Quality']) # Assuming "Quality" is the target column

plt.show()

**4.ML Model Justification**

* **Logistic Regression**: Simple, interpretable, good for binary classification.
* **Random Forest**: Handles non-linearity, robust against missing data.
* **SVM**: Effective in high-dimensional spaces.
* **XGBoost**: Best for feature-rich datasets, highly accurate.

5. ML Model Code (Example with Random Forest)

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# Data preprocessing

X = df.drop(columns=['Quality']) # Assuming "Quality" is the target

y = df['Quality']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Model training

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Predictions

y\_pred = model.predict(X\_test)

# Evaluation

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

**6. Metrics for Model Evaluation**

**The calculated metric terms are**

* **Confusion Matrix**
* **Precision = 0.76**
* **Recall = 0.85**
* **F1-score = 0.86**
* **Live prediction**

**7. Self Inference**

* **The model effectively classifies water quality, reducing dependency on manual testing.**
* **Random Forest/XGBoost yields higher accuracy than simpler models like Logistic Regression.**
* **Turbidity and Sulphate levels significantly impact classification accuracy.**

**8. Scope for Enhancement**

* **Use real-time IoT sensors to feed live data into the model.**
* **Improve accuracy with deep learning models like LSTMs or CNNs for time-series analysis.**
* **Optimize feature selection using PCA or feature importance analysis.**
* **Deploy as a web-based tool for field engineers to assess water quality instantly.**