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.Dataset

The quality of a concrete dataset is crucial for accurate predictions in machine learning applications. A high-quality dataset should include a diverse range of concrete mix designs with detailed attributes such as cement composition, water-cement ratio, aggregate types, curing time, and compressive strength. It should be free from missing values, outliers, and inconsistencies to ensure reliable model training. Proper data preprocessing, normalization, and feature selection enhance predictive performance.

4.Exploratory Data Analysis (EDA) – Code

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


# Load the new dataset

df = pd.read_csv("/content/sample_data/Quality_Concrete.csv")


# Basic info

df.info()
df.describe()


# Check for missing values
print(df.isnull().sum())


# Confusion matrix for visualization
sns.heatmap(confusion_matrix(y_test, pred_rf)/np.sum(confusion_matrix(y_test, pred_rf)), annot=True, fmt='0.2%', cmap='Reds')
plt.show()
```

5.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.

6. 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))
```

7. Metrics for Model Evaluation

The calculated metric terms are

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

8. 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.

9. 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.