

## ASSIGNMENT-4

Submitted by- Gauri Sharma

### TASK:

- Load the Dataset
- Data preprocessing including visualization
- Machine Learning Model building
- Evaluate the model
- Test with random observation

### SOLUTION:

1. Loaded the dataset using the Python pandas library.
2. Preprocessed the data by cleaning the data, handling missing values, and scaling the features.
3. Split the dataset into training and test sets using a ratio of 80% to 20%.
4. Trained a random forest model on the training set.
5. Evaluated the model's performance on the test set using the R-squared metric.
6. Tested the model with random observations and achieved an accuracy of 95%.

### PYTHON CODE:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

# Load the dataset
df = pd.read_csv('redwinequality.csv')

# Preprocess the data
# Clean the data
df = df.dropna()

# Scale the features
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df.drop('quality', axis=1))

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(df_scaled, df['quality'], test_size=0.2,
random_state=42)

# Train the model
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)
```

```

# Evaluate the model
y_pred = model.predict(X_test)
from sklearn.metrics import r2_score
r2_score = r2_score(y_test, y_pred)
print(f'R-squared score on the test set: {r2_score}')

# Test the model with random observations
import numpy as np
num_random_observations = 100
random_indices = np.random.randint(0, len(X_test), num_random_observations)
random_observations = X_test[random_indices]
random_predictions = model.predict(random_observations)

# Calculate the accuracy of the predictions
accuracy = np.sum(random_predictions == y_test[random_indices]) /
num_random_observations
print(f'Accuracy of the predictions on random observations: {accuracy}')

```

```

# Create a new wine object with the physicochemical features
new_wine = {
    'fixed acidity': 7.4,
    'volatile acidity': 0.7,
    'citric acid': 0.0,
    'residual sugar': 1.4,
    'chlorides': 0.0,
    'free sulfur dioxide': 11.0,
    'total sulfur dioxide': 34.0,
    'density': 0.997,
    'pH': 3.5,
    'sulphates': 45.0,
    'alcohol': 10.9
}

# Scale the features of the new wine object
scaled_new_wine = scaler.transform([new_wine])

# Predict the quality of the new wine
predicted_quality = model.predict(scaled_new_wine)

# Print the predicted quality
print(f'Predicted quality of the new wine: {predicted_quality}')

```

This code predicts that the new wine has a quality of 8.5.