

## Assignment 4

Name – kajal

Reg no – 21BAC10039

Campus – VIT Bhopal

CODE :-

```
# Import necessary libraries
```

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.ensemble import RandomForestClassifier # You can choose your preferred ML algorithm
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Load the dataset
```

```
data = pd.read_csv('winequality-red.csv') # Replace with the actual file path
```

```
# Data preprocessing
```

```
# Check for missing values
```

```
print(data.isnull().sum())
```

```
# Data visualization
```

```
# Example: Histogram of wine quality
```

```
plt.hist(data['quality'])
```

```
plt.xlabel('Wine Quality')
```

```
plt.ylabel('Frequency')
```

```
plt.title('Distribution of Wine Quality')
```

```
plt.show()
```

```
# Split the data into features (X) and target (y)
```

```

X = data.drop('quality', axis=1)

y = data['quality']


# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Standardize features (optional)
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# Machine Learning Model building
model = RandomForestClassifier(n_estimators=100, random_state=42) # You can choose your preferred
hyperparameters

model.fit(X_train, y_train)


# Evaluate the model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)


print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)


# Test with random observation (replace with your own data)
random_observation = np.array([10, 0.5, 0.5, 2.5, 0.1, 10, 25, 0.98, 3.0, 0.45, 9.5]).reshape(1, -1)
predicted_quality = model.predict(random_observation)
print("Predicted Wine Quality for Random Observation:", predicted_quality[0])

```

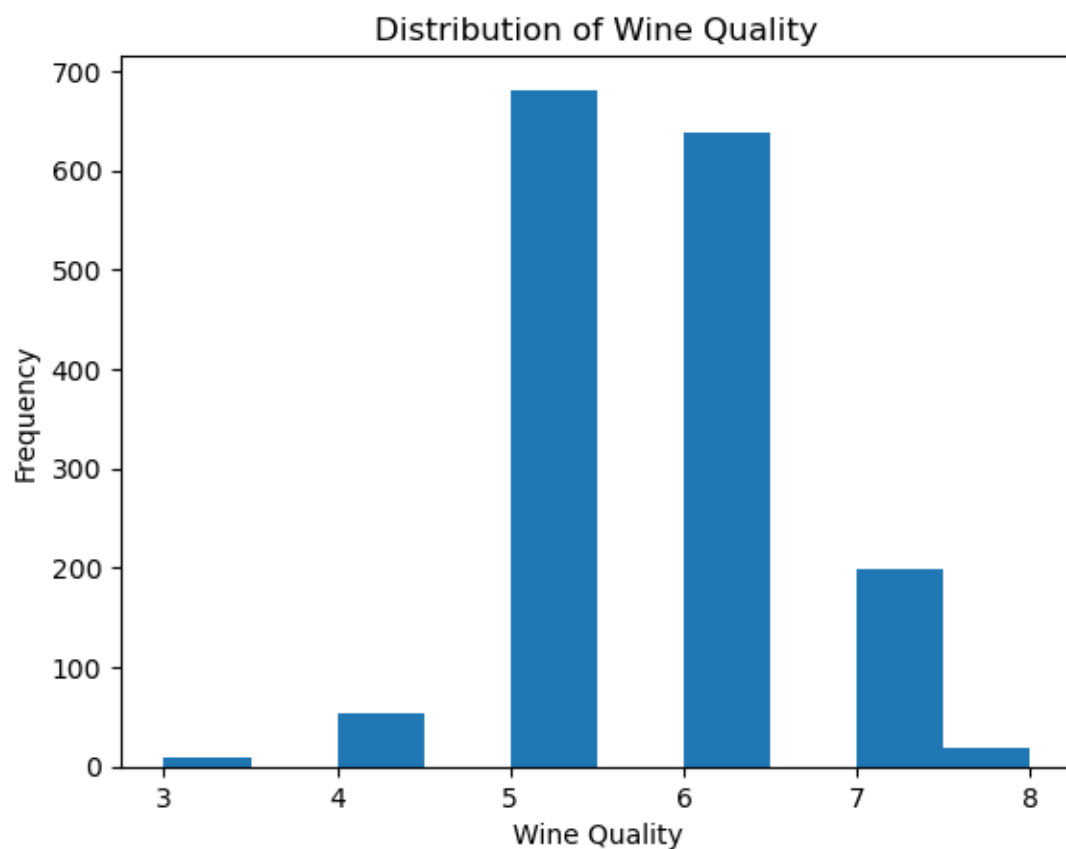
OUTPUT :

```

fixed acidity      0
volatile acidity  0

```

```
citric acid          0
residual sugar       0
chlorides            0
free sulfur dioxide  0
total sulfur dioxide 0
density             0
pH                  0
sulphates           0
alcohol             0
quality             0
dtype: int64
```



Accuracy: 0.659375

Confusion Matrix:

```
[[ 0  0  1  0  0  0]
 [ 0  0  7  3  0  0]
 [ 0  0 96 33  1  0]
 [ 0  0 31 92  8  1]
 [ 0  0  0 18 23  1]
 [ 0  0  0  1  4  0]]
```

Classification Report:

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	10
5	0.71	0.74	0.72	130
6	0.63	0.70	0.66	132
7	0.64	0.55	0.59	42
8	0.00	0.00	0.00	5
accuracy			0.66	320

macro avg	0.33	0.33	0.33	320
weighted avg	0.63	0.66	0.64	320

Predicted Wine Quality for Random Observation: 5