```
pwd
```

```
/kaggle/working/Machinelearning/Wine_quality_data
```

```
ls /kaggle/working/Machinelearning/Wine_quality_data/
```

```
winequality-red.csv winequality-white.csv
```

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_regression import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor

# Configure plot styles
plt.rcParams['figure.figure.dpi'] = 100
plt.rcParams['figure.figure.figsize'] = (10, 6)
sns.set_palette('viridis')

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')

# Display all columns
pd.pd.set_option('display.max_columns', None)

# Load the data
df = pd.read_csv('/kaggle/input/winequality-red.csv')

# Display first 5 rows
df.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
df.tail()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

```
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex:1599 entries,0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Nullcount  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-Null   float64
 1   volatile acidity      1599 non-Null   float64
 2   citric acid           1599 non-Null   float64
 3   residual sugar        1599 non-Null   float64
 4   chlorides             1599 non-Null   float64
 5   free sulfur dioxide   1599 non-Null   float64
 6   total sulfur dioxide  1599 non-Null   float64
 7   density               1599 non-Null   float64
 8   pH                    1599 non-Null   float64
 9   sulphates             1599 non-Null   float64
 10  alcohol               1599 non-Null   float64
 11  quality               1599 non-Null   int64
dtypes:float64(11),int64(1)
memory usage:150.0+KB
```

```
df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

```
# Check for missing values
print("Missing values:")
print(df.isnull().sum())

# Check for duplicate rows
print("\nDuplicate rows:", df.duplicated().sum())
```

```
Missing values:
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

Duplicate rows: 0
```

```
# Create a correlation matrix
corr = df.corr()
plt.figure(figure=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```

```
# Create a pairplot for selected features
sns.pairplot(df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
                 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
                 'pH', 'sulphates', 'alcohol', 'quality']], hue='quality')
plt.show()
```

```
# Create a boxplot for quality vs. alcohol
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='alcohol', data=df)
plt.title('Alcohol by Quality')
plt.show()
```

```
# Create a boxplot for quality vs. sulphates
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='sulphates', data=df)
plt.title('Sulphates by Quality')
plt.show
```

```
# Create a boxplot for quality vs. volatile acidity
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='volatile acidity', data=df)
plt.title('Volatile Acidity by Quality')
plt.show
```

```
# Create a boxplot for quality vs. chlorides
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='chlorides', data=df)
plt.title('Chlorides by Quality')
plt.show
```

```
# Create a boxplot for quality vs. density
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='density', data=df)
plt.title('Density by Quality')
plt.show
```



```
# Create a boxplot for quality vs. pH
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='pH', data=df)
plt.title('pH by Quality')
plt.show
```

```
# Create a boxplot for quality vs. residual sugar
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='residual sugar', data=df)
plt.title('Residual Sugar by Quality')
plt.show
```



```
# Create a boxplot for quality vs. free sulfur dioxide
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='free sulfur dioxide', data=df)
plt.title('Free Sulfur Dioxide by Quality')
plt.show
```



```
# Create a boxplot for quality vs. total sulfur dioxide
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='total sulfur dioxide', data=df)
plt.title('Total Sulfur Dioxide by Quality')
plt.show
```



```
# Create a boxplot for quality vs. density
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='density', data=df)
plt.title('Density by Quality')
plt.show
```



```
# Create a boxplot for quality vs. pH
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='pH', data=df)
plt.title('pH by Quality')
plt.show
```



```
# Create a boxplot for quality vs. sulphates
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='sulphates', data=df)
plt.title('Sulphates by Quality')
plt.show
```



```
# Create a boxplot for quality vs. alcohol
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='alcohol', data=df)
plt.title('Alcohol by Quality')
plt.show
```



```
# Create a boxplot for quality vs. volatile acidity
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='volatile acidity', data=df)
plt.title('Volatile Acidity by Quality')
plt.show
```



```
# Create a boxplot for quality vs. chlorides
plt.figure(figure=(10, 6))
sns.boxplot(x='quality', y='chlorides', data=df)
plt.title('Chlorides by Quality')
plt.show
```


Chlorides by Quality
chlorides