WHITE WINE QUALITY PREDICTION

```
[ ] #Importing libraries
    import pandas as pd

    d= pd.read_csv('/content/archive.zip', sep=';')
    d.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

```
d.shape
```

```
(4898, 12)
```

```
[ ] d.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality

```
[ ] d.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quali
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.0000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5.8778
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0.8856
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3.0000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5.0000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6.0000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6.0000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9.0000

```
#Feature scaling
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(d.drop('quality', axis=1))
```

Defining Target Variable (y) and Feature Variables (X)

```
y=d['quality']
y
```

	quality
0	6
1	6
2	6
3	6
4	6
...	...
4893	6
4894	5
4895	6
4896	7
4897	6

4898 rows × 1 columns

dtype: int64

```python
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

+ Code + Text

```
x = d[['alcohol','volatile acidity','fixed acidity','citric acid','residual sugar','chlorides','free sulfur dioxide','total sulfur dioxide','density','pH','sulphates','alcohol']
x
```

	alcohol	volatile acidity	fixed acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	8.8	0.27	7.0	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	8.8
1	9.5	0.30	6.3	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	9.5
2	10.1	0.28	8.1	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	10.1
3	9.9	0.23	7.2	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9
4	9.9	0.23	7.2	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9
...	...	...	...	...	...	...	...	...	...	...	...	...
4893	11.2	0.21	6.2	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0.50	11.2
4894	9.6	0.32	6.6	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0.46	9.6
4895	9.4	0.24	6.5	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0.46	9.4
4896	12.8	0.29	5.5	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0.38	12.8
4897	11.8	0.21	6.0	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0.32	11.8

4898 rows × 12 columns

4898 rows × 12 columns

TRAIN TEST SPLIT

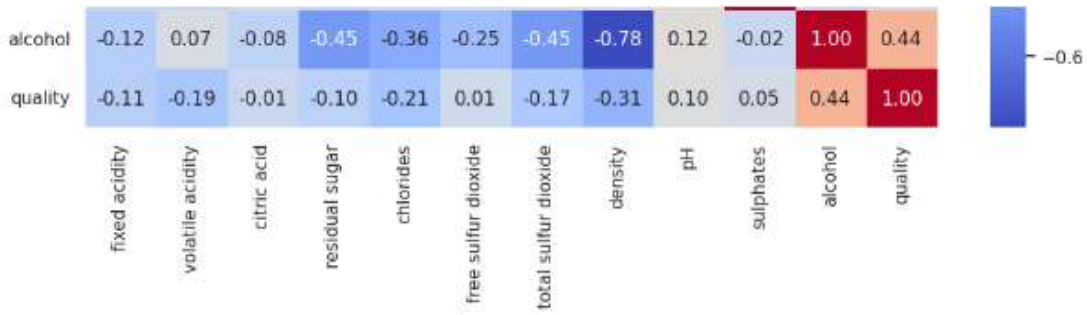
```
from sklearn.model_selection import train_test_split

X = d.drop('quality', axis=1)
y = d['quality']

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

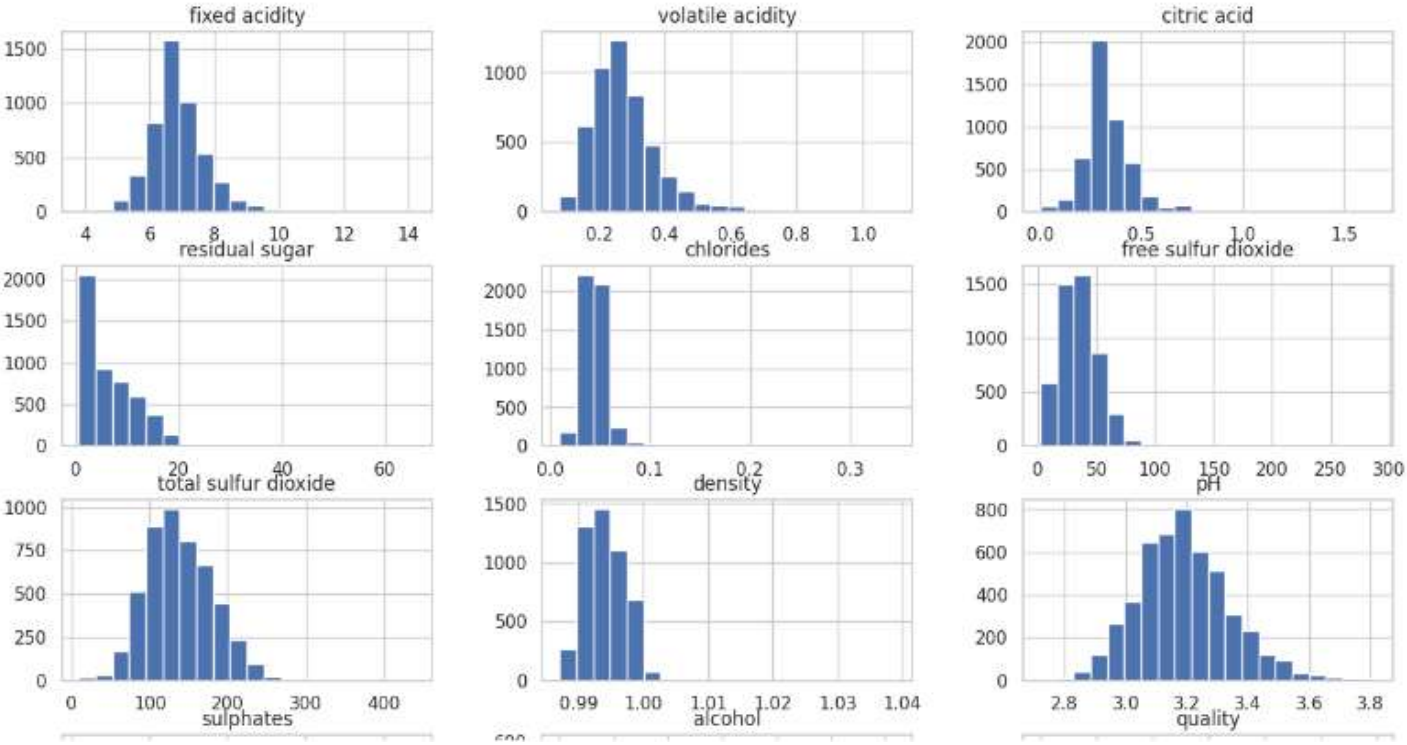
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

DATA VISUALIZATION

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot style
sns.set(style="whitegrid")

# Correlation matrix to visualize relationships between variables
plt.figure(figsize=(12, 8))
corr_matrix = d.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Wine Quality Features')
plt.show()

# Histograms for each feature to see the distribution
d.hist(bins=20, figsize=(15, 10), layout=(4, 3))
plt.suptitle('Feature Distributions')
plt.show()
```



Feature Distributions

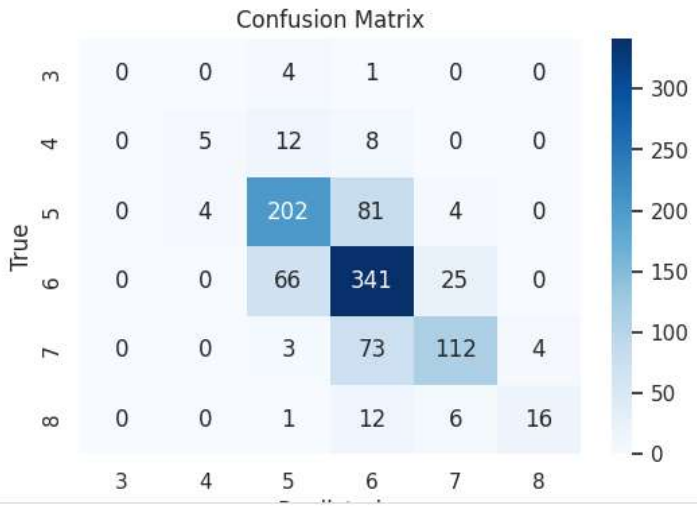


```python
#Classification report
from sklearn.metrics import classification_report, confusion_matrix

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.56      0.20      0.29        25
           5       0.70      0.69      0.70       291
           6       0.66      0.79      0.72       432
           7       0.76      0.58      0.66       192
           8       0.80      0.46      0.58        35

    accuracy                           0.69       980
   macro avg       0.58      0.45      0.49       980
weighted avg       0.69      0.69      0.68       980

[[  0   0   4   1   0   0]
 [  0   5  12   8   0   0]
 [  0   4 202  81   4   0]
 [  0   0  66 341  25   0]
 [  0   0   3  73 112   4]
 [  0   0   1  12   6  16]]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_d
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_d
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_d
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

CONFUSION MATRIX

```
[ ] from sklearn.metrics import confusion_matrix, classification_report
    import seaborn as sns
    import matplotlib.pyplot as plt

    # Generate confusion matrix
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Define labels for the confusion matrix
    labels = ['3', '4', '5', '6', '7', '8'] # Example labels, replace with your actual labels

    # Plot confusion matrix
    plt.figure(figsize=(6, 4))
    sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=labels, yticklabels=labels)
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()
```



```
[ ] from sklearn.metrics import mean_squared_error, r2_score

    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
```

```
[ ] from sklearn.model_selection import cross_val_score

    scores = cross_val_score(model, X, y, cv=5)
    print(scores.mean())
```

```
0.5206226678618332
```

DEFINITION

The White Wine Quality dataset consists of 4898 entries, each representing a sample of white wine. The dataset is made up of 11 physicochemical properties (features) that describe the wine, and one target variable, quality, which is a score between 0 and 10. Below is a summary of the key elements of the dataset:

DATASET OVERVIEW

1.Total Records: 4898

2.Total Features: 11

3.Target Variable: quality (integer value ranging from 0 to 10, representing the quality of the wine)

Features (Inputs)

fixed acidity,volatile acidity,citric acid,residual sugar,chlorides,free sulfur dioxide,total sulfur dioxide,density,pH,sulphates,alcohol.

Target Variable (Output)

Quality: Integer score (from 0 to 10) representing the quality of the wine. This score is typically used for regression or classification tasks in machine learning projects.

General Observation:

1.No missing values: All entries in the dataset are complete, with no missing data.

2.Feature Correlations: Several features, such as alcohol and density, show potential correlations with the target quality score, which can be further explored through analysis and visualization.