

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.preprocessing import StandardScaler
```

```
In [2]: df=pd.read_csv("WineQT.csv")
```

```
In [3]: df
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	0
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5	1
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5	2
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6	3
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5	4
...
1138	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6	1592
1139	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	0.82	9.5	6	1593
1140	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5	1594
1141	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6	1595
1142	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5	1597

1143 rows × 13 columns

In [8]: `df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide    1143 non-null   float64
 6   total sulfur dioxide   1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                   1143 non-null   float64
 9   sulphates             1143 non-null   float64
10   alcohol               1143 non-null   float64
11   quality               1143 non-null   int64  
12   Id                   1143 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 116.2 KB

```

```
In [5]: df.describe()
```

Out[5]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
count	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000	1143.000000
mean	8.311111	0.531339	0.268364	2.532152	0.086933	15.615486	45.914698	0.996730	3.311015	0.657708
std	1.747595	0.179633	0.196686	1.355917	0.047267	10.250486	32.782130	0.001925	0.156664	0.170399
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000
25%	7.100000	0.392500	0.090000	1.900000	0.070000	7.000000	21.000000	0.995570	3.205000	0.550000
50%	7.900000	0.520000	0.250000	2.200000	0.079000	13.000000	37.000000	0.996680	3.310000	0.620000
75%	9.100000	0.640000	0.420000	2.600000	0.090000	21.000000	61.000000	0.997845	3.400000	0.730000
max	15.900000	1.580000	1.000000	15.500000	0.611000	68.000000	289.000000	1.003690	4.010000	2.000000

In [6]: `df.head()`

Out[6]:

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

In [7]: `df.shape`

Out[7]: (1143, 13)

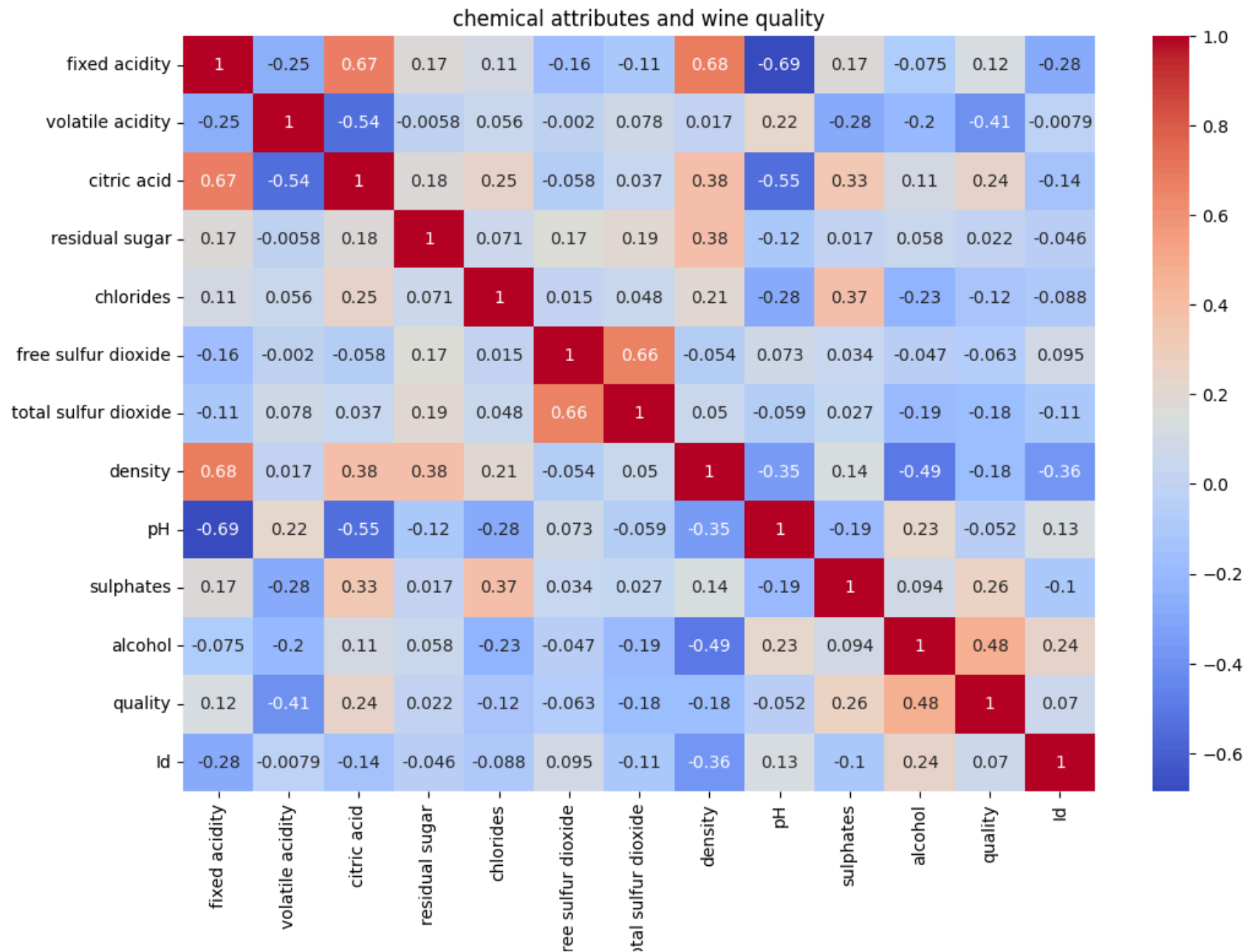
In [9]: `df.shape[0]`

Out[9]: 1143

```
In [10]: print(df.isnull().sum())
```

```
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
Id                0
dtype: int64
```

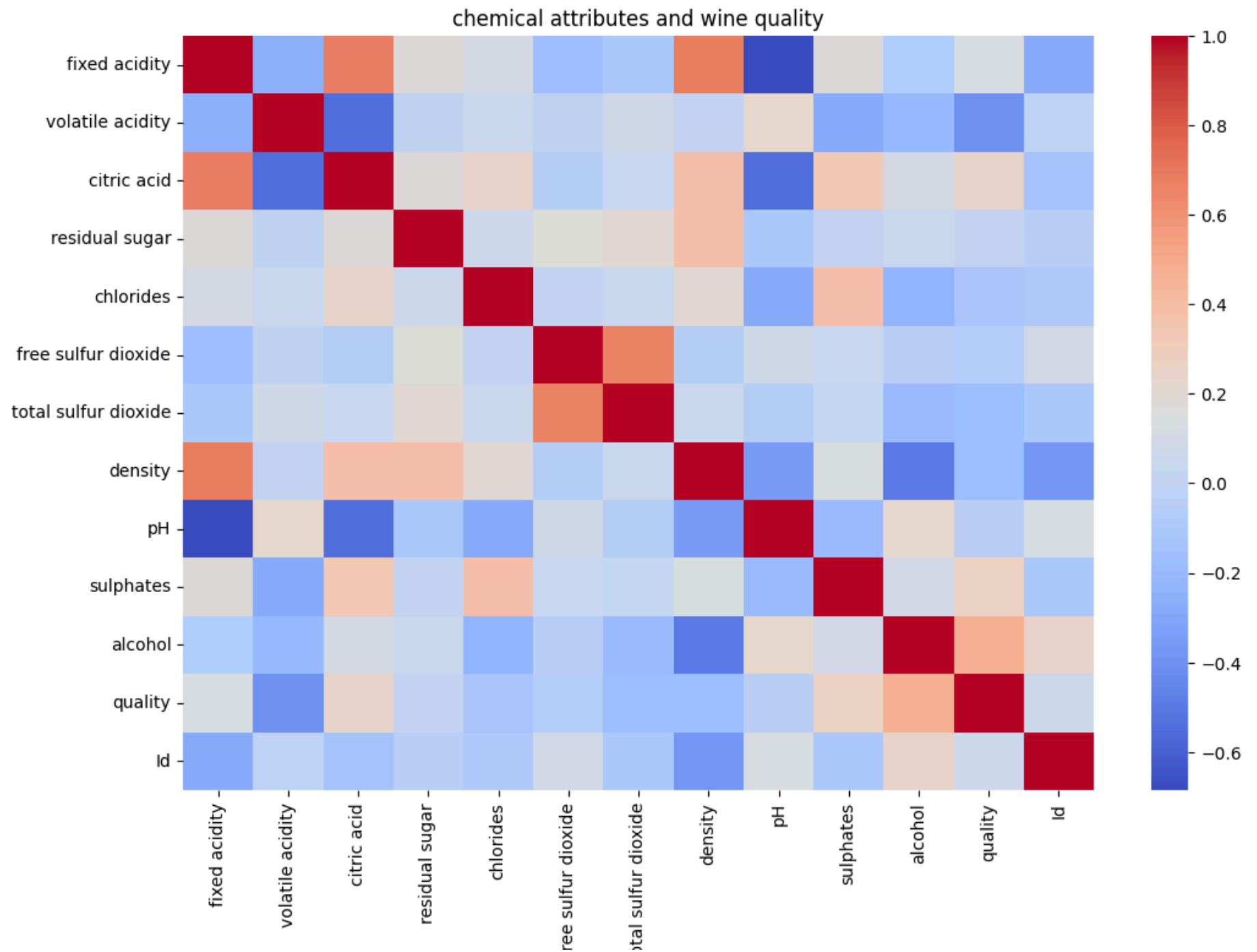
```
In [12]: plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('chemical attributes and wine quality')
plt.show()
```



f

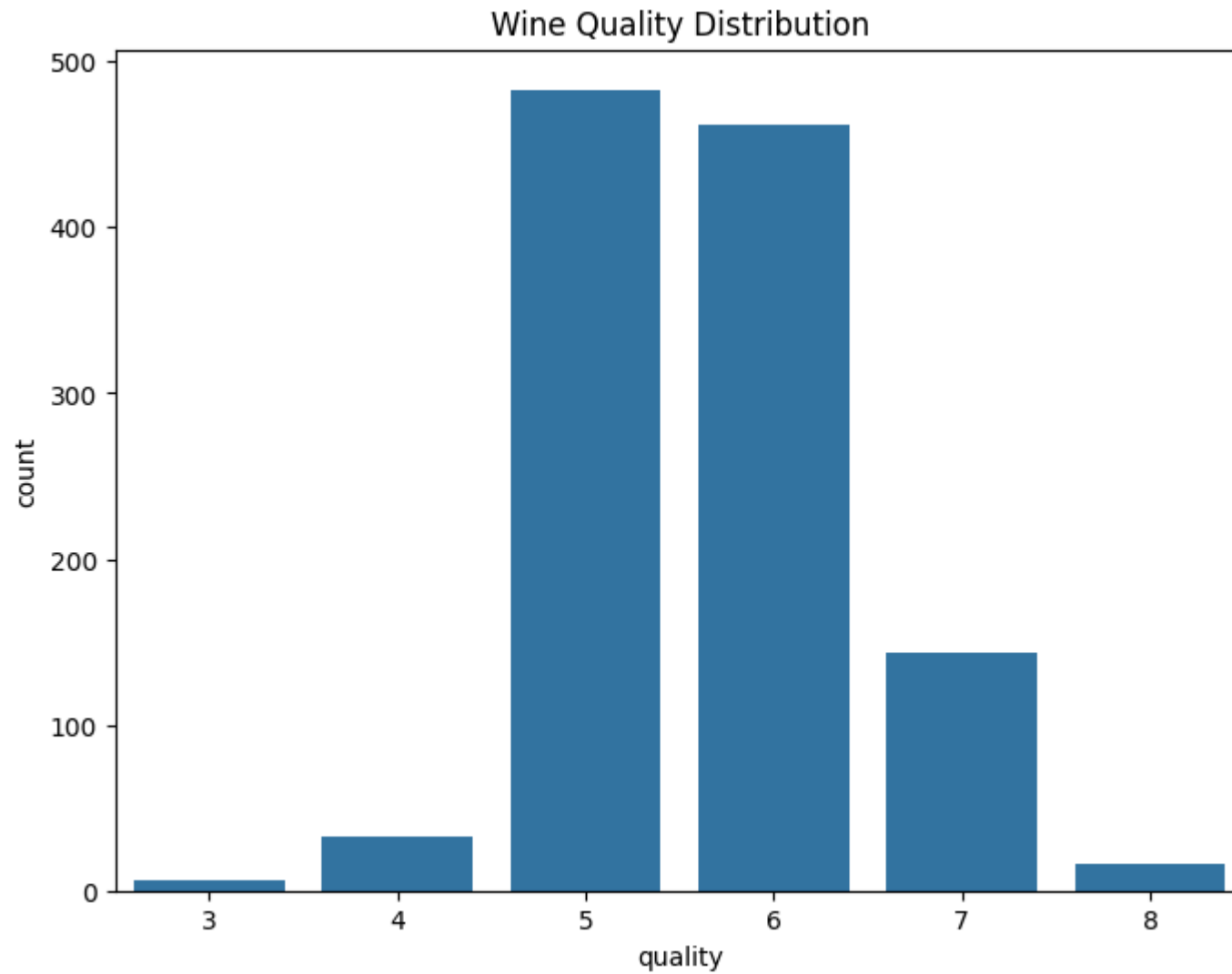
tc

```
In [37]: plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=False, cmap='coolwarm')
plt.title('chemical attributes and wine quality')
plt.show()
```

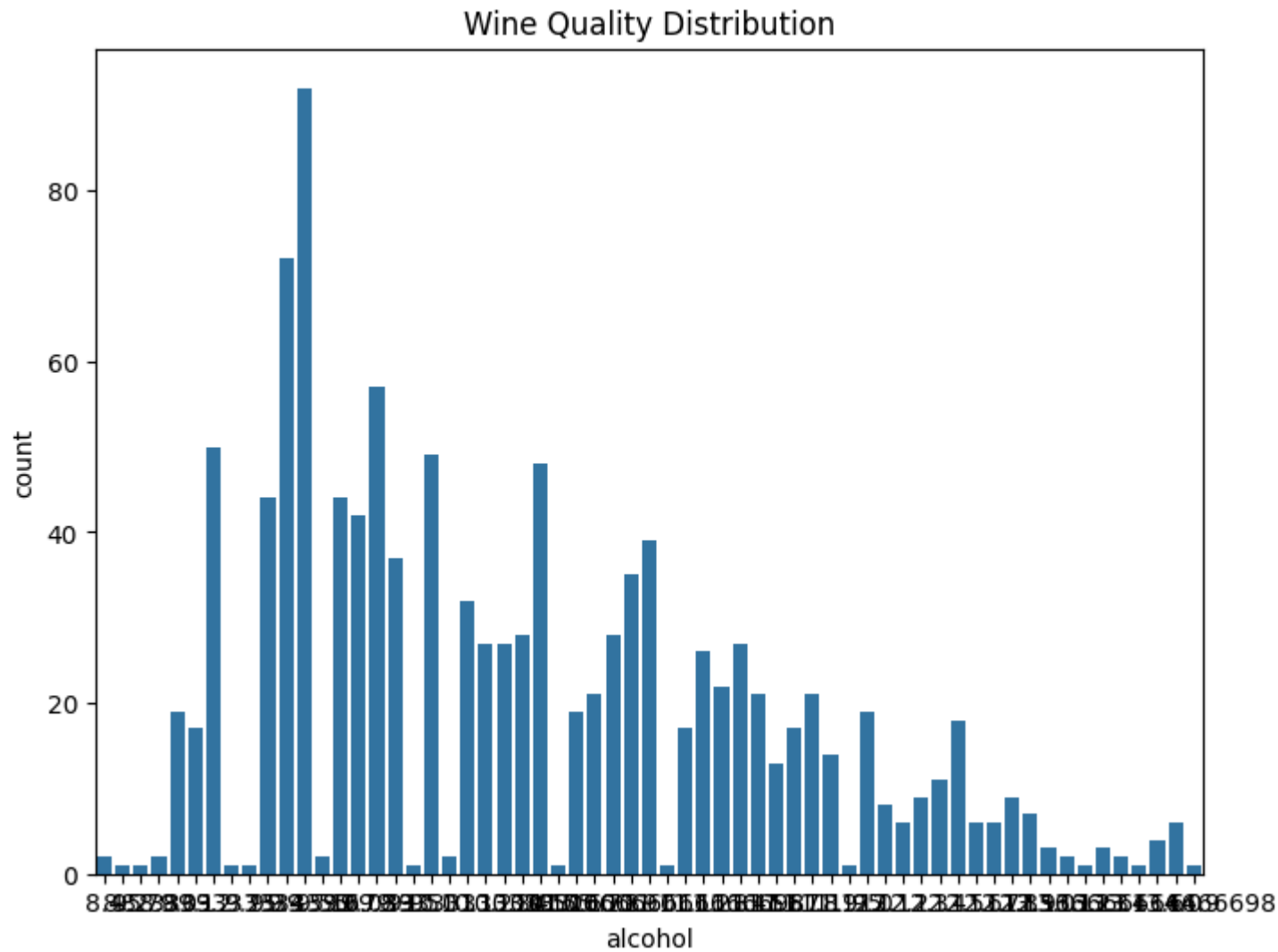


f t

```
In [18]: plt.figure(figsize=(8, 6))
sns.countplot(x='quality', data=df)
plt.title('Wine Quality Distribution')
plt.show()
```



```
In [19]: plt.figure(figsize=(8, 6))
sns.countplot(x='alcohol', data=df)
plt.title('Wine Quality Distribution')
plt.show()
```



```
In [21]: plt.figure(figsize=(8, 6))
sns.countplot(x='Id', data=df)
```

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



```
In [23]: X = df.drop('quality', axis=1)  
         y = df['quality']
```

```
In [24]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [25]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [26]: rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
```

```
In [27]: print("Random Forest Classifier Accuracy:", accuracy_score(y_test, rf_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, rf_pred))
print("Classification Report:\n", classification_report(y_test, rf_pred))
```

Random Forest Classifier Accuracy: 0.6899563318777293

Confusion Matrix:

```
[[ 0  3  3  0  0]
 [ 0 72 24  0  0]
 [ 0 24 70  5  0]
 [ 0  0 10 16  0]
 [ 0  0  2  0  0]]
```

Classification Report:

	precision	recall	f1-score	support
4	0.00	0.00	0.00	6
5	0.73	0.75	0.74	96
6	0.64	0.71	0.67	99
7	0.76	0.62	0.68	26
8	0.00	0.00	0.00	2
accuracy			0.69	229
macro avg	0.43	0.41	0.42	229
weighted avg	0.67	0.69	0.68	229

```
C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
In [28]: sgd_model = SGDClassifier(random_state=42)
sgd_model.fit(X_train, y_train)
sgd_pred = sgd_model.predict(X_test)
```

```
In [29]: print("\nStochastic Gradient Descent Accuracy:", accuracy_score(y_test, sgd_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, sgd_pred))
print("Classification Report:\n", classification_report(y_test, sgd_pred))
```

Stochastic Gradient Descent Accuracy: 0.5851528384279476

Confusion Matrix:

```
[[ 0  2  4  0  0]
 [ 0 75 12  8  1]
 [ 0 31 39 29  0]
 [ 0  2  4 20  0]
 [ 0  0  0  2  0]]
```

Classification Report:

	precision	recall	f1-score	support
4	0.00	0.00	0.00	6
5	0.68	0.78	0.73	96
6	0.66	0.39	0.49	99
7	0.34	0.77	0.47	26
8	0.00	0.00	0.00	2
accuracy			0.59	229
macro avg	0.34	0.39	0.34	229
weighted avg	0.61	0.59	0.57	229

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
In [30]: svc_model = SVC(random_state=42)
svc_model.fit(X_train, y_train)
svc_pred = svc_model.predict(X_test)
```

```
In [31]: print("\nSupport Vector Classifier Accuracy:", accuracy_score(y_test, svc_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, svc_pred))
print("Classification Report:\n", classification_report(y_test, svc_pred))
```

Support Vector Classifier Accuracy: 0.6550218340611353

Confusion Matrix:

```
[[ 0  3  3  0  0]
 [ 0 72 24  0  0]
 [ 0 27 69  3  0]
 [ 0  1 16  9  0]
 [ 0  0  1  1  0]]
```

Classification Report:

	precision	recall	f1-score	support
4	0.00	0.00	0.00	6
5	0.70	0.75	0.72	96
6	0.61	0.70	0.65	99
7	0.69	0.35	0.46	26
8	0.00	0.00	0.00	2
accuracy			0.66	229
macro avg	0.40	0.36	0.37	229
weighted avg	0.64	0.66	0.64	229

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

C:\Users\rushi\AppData\Local\Programs\Python\Python312\Lib\site-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_division` parameter to control this behavior.

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
In [32]: feature_importance = rf_model.feature_importances_
sorted_idx = np.argsort(feature_importance)

plt.figure(figsize=(10, 6))
plt.barh(range(len(sorted_idx)), feature_importance[sorted_idx], align='center')
plt.yticks(range(len(sorted_idx)), X.columns[sorted_idx])
```

```
plt.title('Rushikesh karpe')  
plt.show()
```

