

Name: Chinmay Mhatre Roll No 22102B2001 BE CMPN B Github link: <https://github.com/chinmay0910/ML-Lab---VIT>

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
import seaborn as sns
```

```
!pip install ucimlrepo
```

```

Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.0.7)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ucimlrepo) (2.1.4)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.10/dist-packages (from ucimlrepo) (2024.8.30)
Requirement already satisfied: numpy<2, >=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.0->ucimlrepo) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.0->ucin

```

```
from ucimlrepo import fetch_ucirepo
```

```
# fetch dataset
wine_quality = fetch_ucirepo(id=186)
```

```
# data (as pandas dataframes)
X = wine_quality.data.features
y = wine_quality.data.targets
```

```
# metadata
print(wine_quality.metadata)
```

```
# variable information
print(wine_quality.variables)
```

```
➡ {'uci_id': 186, 'name': 'Wine Quality', 'repository_url': 'https://archive.ics.uci.edu/dataset/186/wine+quality', 'data_url': 'http://archive.ics.uci.edu/ml/machine-learning-databases/winequality/winequality.data'
```

	name	role	type	demographic \
0	fixed_acidity	Feature	Continuous	None
1	volatile_acidity	Feature	Continuous	None
2	citric_acid	Feature	Continuous	None
3	residual_sugar	Feature	Continuous	None
4	chlorides	Feature	Continuous	None
5	free_sulfur_dioxide	Feature	Continuous	None
6	total_sulfur_dioxide	Feature	Continuous	None
7	density	Feature	Continuous	None
8	pH	Feature	Continuous	None
9	sulphates	Feature	Continuous	None
10	alcohol	Feature	Continuous	None
11	quality	Target	Integer	None
12	color	Other	Categorical	None

	description	units	missing_values
0	None	None	no
1	None	None	no
2	None	None	no
3	None	None	no
4	None	None	no
5	None	None	no
6	None	None	no
7	None	None	no
8	None	None	no
9	None	None	no
10	None	None	no
11	score between 0 and 10	None	no
12	red or white	None	no

```
data = pd.concat([X, y], axis=1)
```

```
data.head()
```

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51

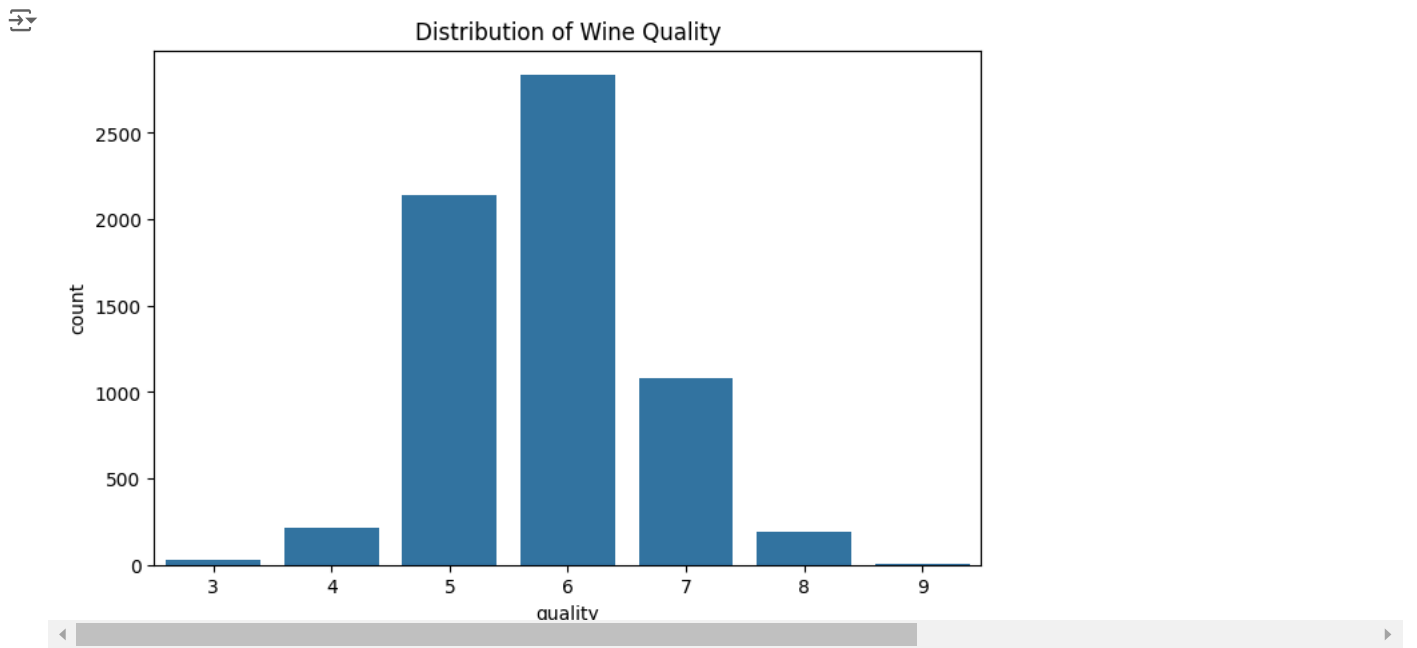
```
data.isnull().sum()
```

	0
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

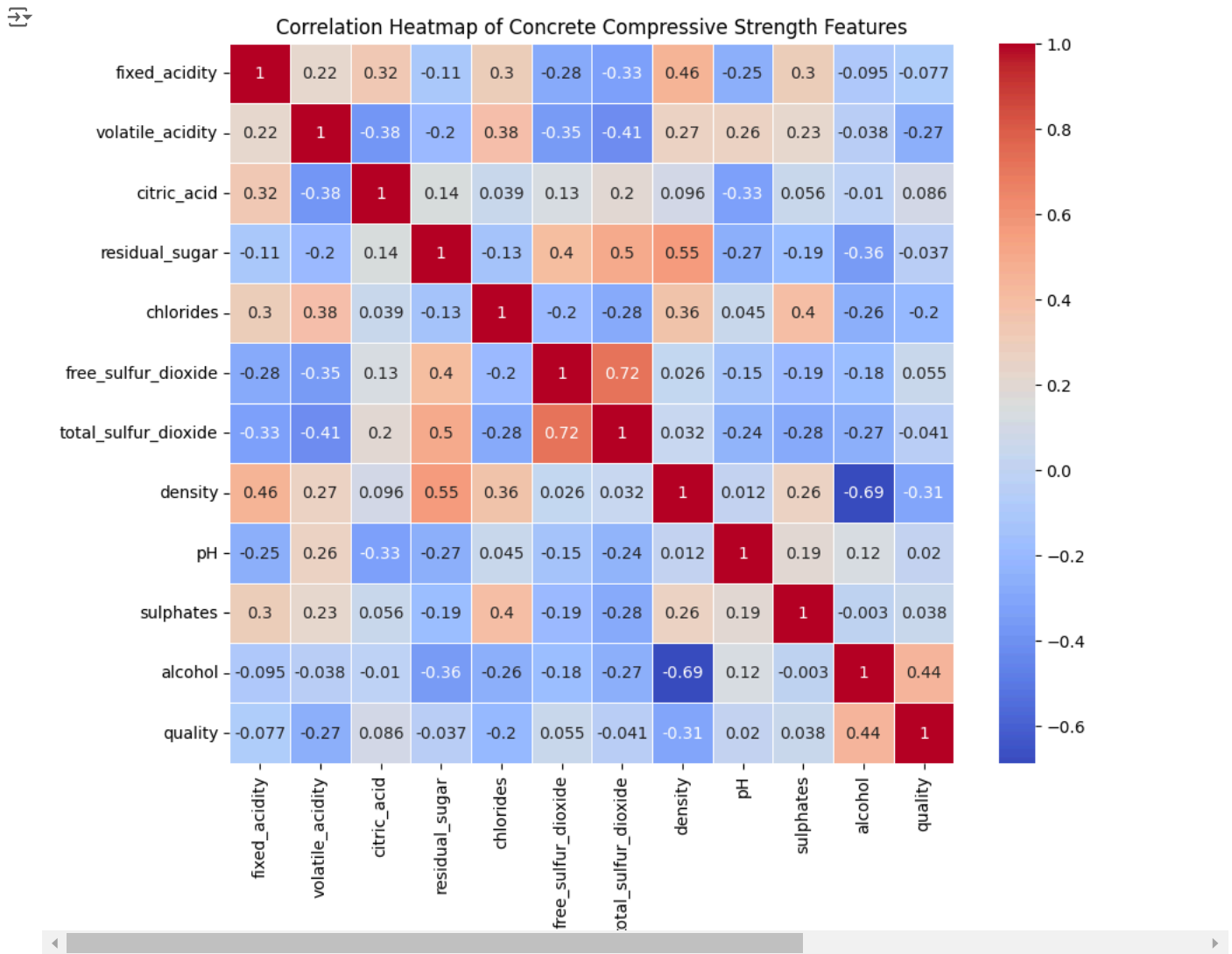
```
data.shape
```

```
(6497, 12)
```

```
plt.figure(figsize=(8, 5))
sns.countplot(x='quality', data=data)
plt.title('Distribution of Wine Quality')
plt.show()
```

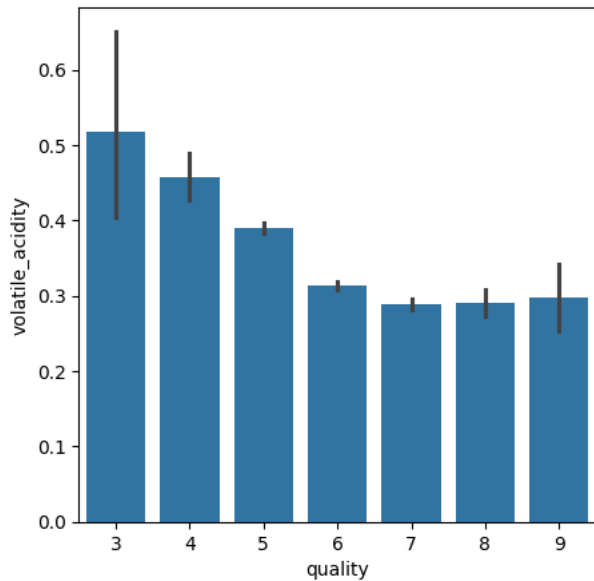


```
plt.figure(figsize=(10, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap of Concrete Compressive Strength Features')
plt.show()
```



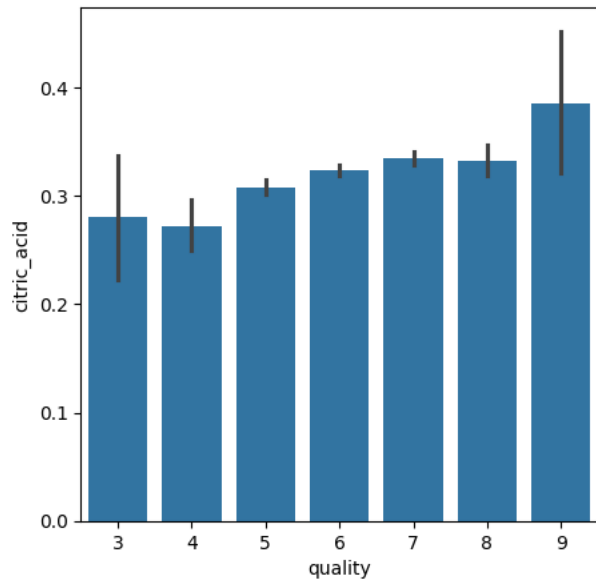
```
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y='volatile_acidity', data=data)
```

<Axes: xlabel='quality', ylabel='volatile_acidity'>



```
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y='citric_acid', data=data)
```

```
<Axes: xlabel='quality', ylabel='citric_acid'>
```



```
X.head()
```

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51

```
#Label Binarization
```

```
y = data['quality'].apply(lambda y_value: 1 if y_value >= 7 else 0)
```

```
y
```

	quality
0	0
1	0
2	0
3	0
4	0
...	...
6492	0
6493	0
6494	0
6495	1
6496	0

6497 rows × 1 columns

dtype: int64

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=24)
```

```
# Standardize the features
```

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```
svm_classifier = SVC(probability=True)
```

```
svm_classifier.fit(X_train, y_train)
```



```
SVC
SVC(probability=True)
```

```
y_pred = svm_classifier.predict(X_test)
```

```
y_prob = svm_classifier.predict_proba(X_test)[:, 1]
```

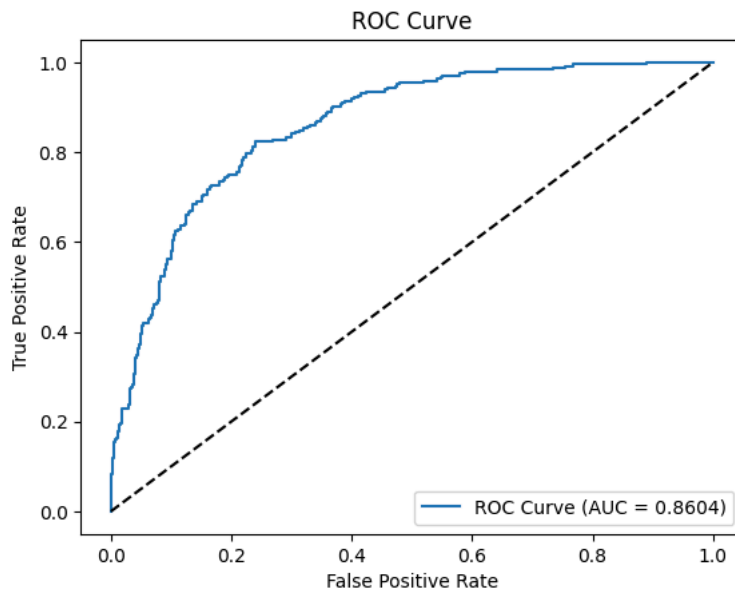
```
# Evaluate performance
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_prob)
```

```
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
print(f"ROC AUC: {roc_auc:.4f}")
```



```
Accuracy: 0.8354
Precision: 0.6476
Recall: 0.2776
F1 Score: 0.3886
ROC AUC: 0.8604
```

```
# Plot ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.4f})")
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```



Start coding or [generate](#) with AI.