```python
# -*- coding: utf-8 -*-
"""Wine Quality Prediction.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1bfOtUALZQ-auBDcbWNtesfKlayjL6Dpi

Import Dependencies
"""

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.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

"""Data Collect"""

# loading the dataset to a Pandas DataFrame
wine_dataset = pd.read_csv('/content/Red Wine Quality.csv')

# number of rows & columns in the dataset
wine_dataset.shape

# first 5 rows of the dataset
wine_dataset.head()

# checking for missing values
wine_dataset.isnull().sum()

"""Data Analysis and Visulaization"""

# statistical measures of the dataset
wine_dataset.describe()

# number of values for each quality
sns.catplot(x='quality', data = wine_dataset, kind = 'count')

# volatile acidity vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y = 'volatile acidity', data = wine_dataset)

# citric acid vs Quality
plot = plt.figure(figsize=(5,5))
sns.barplot(x='quality', y = 'citric acid', data = wine_dataset)

"""
Correlation

1. Positive Correlation
2. Negative Correlation"""

correlation = wine_dataset.corr()

# constructing a heatmap to understand the correlation between the columns
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt = '.1f', annot = True, annot_kws={'size':8}, cmap = 'Blues')

"""Data Preprocessing"""

# separate the data and Label
X = wine_dataset.drop('quality',axis=1)

print(X)

"""Label Binarizaton"""

Y = wine_dataset['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)

print(Y)

"""Train & Test Split"""
```

```python
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)

print(Y.shape, Y_train.shape, Y_test.shape)

"""Model Training:

Random Forest Classifier
"""

model = RandomForestClassifier()

model.fit(X_train, Y_train)

"""Model Evaluation

Accuracy Score
"""

# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print('Accuracy : ', test_data_accuracy)

"""Building a Predictive System"""

input_data = (7.5,0.5,0.36,6.1,0.071,17.0,102.0,0.9978,3.35,0.8,10.5)

# changing the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the data as we are predicting the label for only one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]==1):
  print('Good Quality Wine')
else:
  print('Bad Quality Wine')
```