# wine-prediction-ultimate

June 21, 2024

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('winequalityN.csv')
```

[2]:
```
df.head()
```

[2]:
```
    type  fixed acidity  volatile acidity  citric acid  residual sugar  \
0  white            7.0              0.27         0.36            20.7
1  white            6.3              0.30         0.34             1.6
2  white            8.1              0.28         0.40             6.9
3  white            7.2              0.23         0.32             8.5
4  white            7.2              0.23         0.32             8.5

   chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
0      0.045                 45.0                 170.0   1.0010  3.00
1      0.049                 14.0                 132.0   0.9940  3.30
2      0.050                 30.0                  97.0   0.9951  3.26
3      0.058                 47.0                 186.0   0.9956  3.19
4      0.058                 47.0                 186.0   0.9956  3.19

   sulphates  alcohol  quality
0       0.45      8.8        6
1       0.49      9.5        6
2       0.44     10.1        6
3       0.40      9.9        6
4       0.40      9.9        6
```

[3]:
```
df.tail()
```

[3]:
```
      type  fixed acidity  volatile acidity  citric acid  residual sugar  \
6492   red            6.2             0.600         0.08             2.0
6493   red            5.9             0.550         0.10             2.2
6494   red            6.3             0.510         0.13             2.3
6495   red            5.9             0.645         0.12             2.0
```

```
6496  red            6.0             0.310         0.47             3.6

      chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
6492      0.090                 32.0                  44.0  0.99490  3.45
6493      0.062                 39.0                  51.0  0.99512  3.52
6494      0.076                 29.0                  40.0  0.99574  3.42
6495      0.075                 32.0                  44.0  0.99547  3.57
6496      0.067                 18.0                  42.0  0.99549  3.39

      sulphates  alcohol  quality
6492       0.58     10.5        5
6493        NaN     11.2        6
6494       0.75     11.0        6
6495       0.71     10.2        5
6496       0.66     11.0        6
```

[4]:
```
df
```

[4]:
```
        type  fixed acidity  volatile acidity  citric acid  residual sugar  \
0     white            7.0             0.270         0.36            20.7
1     white            6.3             0.300         0.34             1.6
2     white            8.1             0.280         0.40             6.9
3     white            7.2             0.230         0.32             8.5
4     white            7.2             0.230         0.32             8.5
…     …            …               …          …             …
6492    red            6.2             0.600         0.08             2.0
6493    red            5.9             0.550         0.10             2.2
6494    red            6.3             0.510         0.13             2.3
6495    red            5.9             0.645         0.12             2.0
6496    red            6.0             0.310         0.47             3.6

      chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
0         0.045                 45.0                 170.0  1.00100  3.00
1         0.049                 14.0                 132.0  0.99400  3.30
2         0.050                 30.0                  97.0  0.99510  3.26
3         0.058                 47.0                 186.0  0.99560  3.19
4         0.058                 47.0                 186.0  0.99560  3.19
…         …                  …                   …      …   …
6492      0.090                 32.0                  44.0  0.99490  3.45
6493      0.062                 39.0                  51.0  0.99512  3.52
6494      0.076                 29.0                  40.0  0.99574  3.42
6495      0.075                 32.0                  44.0  0.99547  3.57
6496      0.067                 18.0                  42.0  0.99549  3.39

      sulphates  alcohol  quality
0          0.45      8.8        6
1          0.49      9.5        6
```

```
2          0.44     10.1        6
3          0.40      9.9        6
4          0.40      9.9        6
…         …      …      …
6492       0.58     10.5        5
6493        NaN     11.2        6
6494       0.75     11.0        6
6495       0.71     10.2        5
6496       0.66     11.0        6

[6497 rows x 13 columns]
```

[5]:
```
df.loc[df['quality'] >= 7]
```

[5]:
```
       type  fixed acidity  volatile acidity  citric acid  residual sugar  \
13    white            6.6              0.16         0.40             1.5
15    white            6.6              0.17         0.38             1.5
17    white            NaN              0.66         0.48             1.2
20    white            6.2              0.66         0.48             1.2
21    white            6.4              0.31         0.38             2.9
…     …             …                …          …             …
6439    red            7.4              0.25         0.29             2.2
6442    red            8.4              0.37         0.43             2.3
6447    red            7.4              0.36         0.30             1.8
6453    red            7.0              0.56         0.17             1.7
6482    red            6.7              0.32         0.44             2.4

      chlorides  free sulfur dioxide  total sulfur dioxide  density    pH  \
13        0.044                 48.0                 143.0  0.99120  3.54
15        0.032                 28.0                 112.0  0.99140  3.25
17        0.029                 29.0                  75.0  0.98920  3.33
20        0.029                 29.0                  75.0  0.98920  3.33
21        0.038                 19.0                 102.0  0.99120  3.17
…         …                   …                   …      …   …
6439      0.054                 19.0                  49.0  0.99666  3.40
6442      0.063                 12.0                  19.0  0.99550  3.17
6447      0.074                 17.0                  24.0  0.99419  3.24
6453      0.065                 15.0                  24.0  0.99514  3.44
6482      0.061                 24.0                  34.0  0.99484  3.29

      sulphates  alcohol  quality
13         0.52    12.40        7
15         0.55    11.40        7
17         0.39    12.80        8
20         0.39    12.80        8
21         0.35    11.00        7
…         …      …      …
```

```
6439       0.76    10.90        7
6442       0.81    11.20        7
6447       0.70    11.40        8
6453       0.68    10.55        7
6482       0.80    11.60        7

[1277 rows x 13 columns]
```

```
[6]: df.isnull().sum()
```

```
[6]: type                     0
     fixed acidity           10
     volatile acidity         8
     citric acid              3
     residual sugar           2
     chlorides                2
     free sulfur dioxide      0
     total sulfur dioxide     0
     density                  0
     pH                       9
     sulphates                4
     alcohol                  0
     quality                  0
     dtype: int64
```

```
[7]: df.dropna(inplace=True)
```

```
[8]: df.isnull().sum()
```

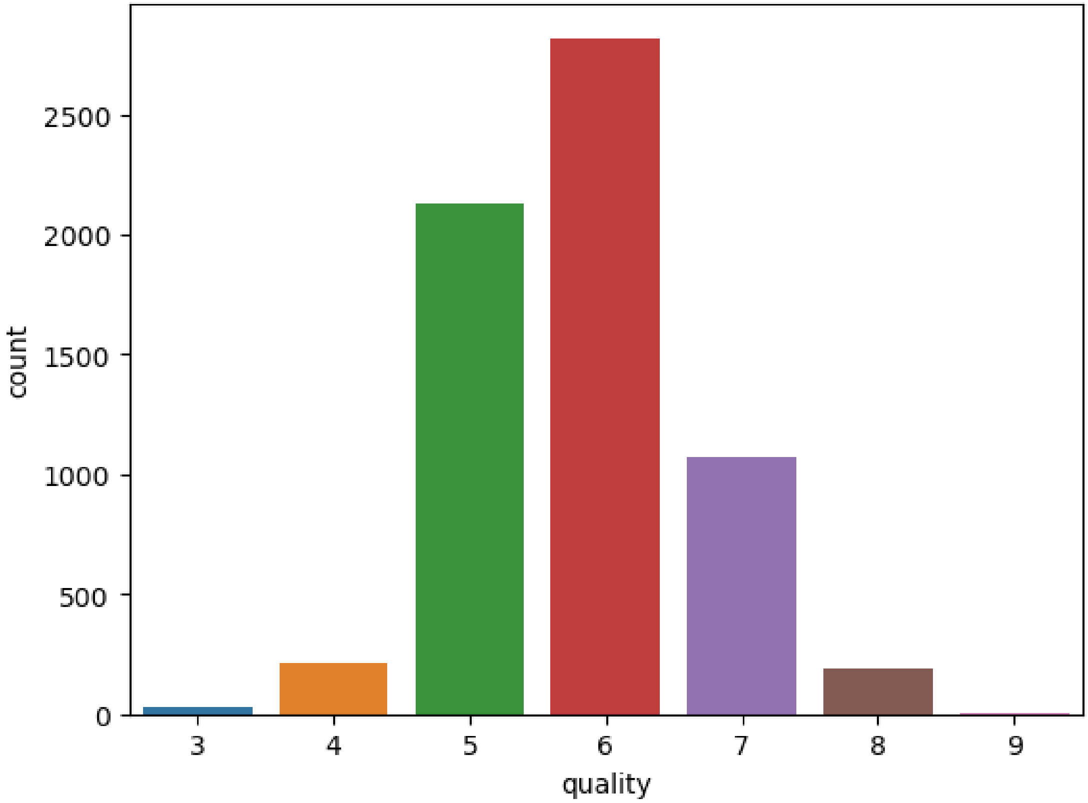
```
[8]: type                    0
     fixed acidity           0
     volatile acidity        0
     citric acid             0
     residual sugar          0
     chlorides               0
     free sulfur dioxide     0
     total sulfur dioxide    0
     density                 0
     pH                      0
     sulphates               0
     alcohol                 0
     quality                 0
     dtype: int64
```

```
[9]: df['type'].value_counts()
```

[9]:
```
type
white    4870
red      1593
Name: count, dtype: int64
```

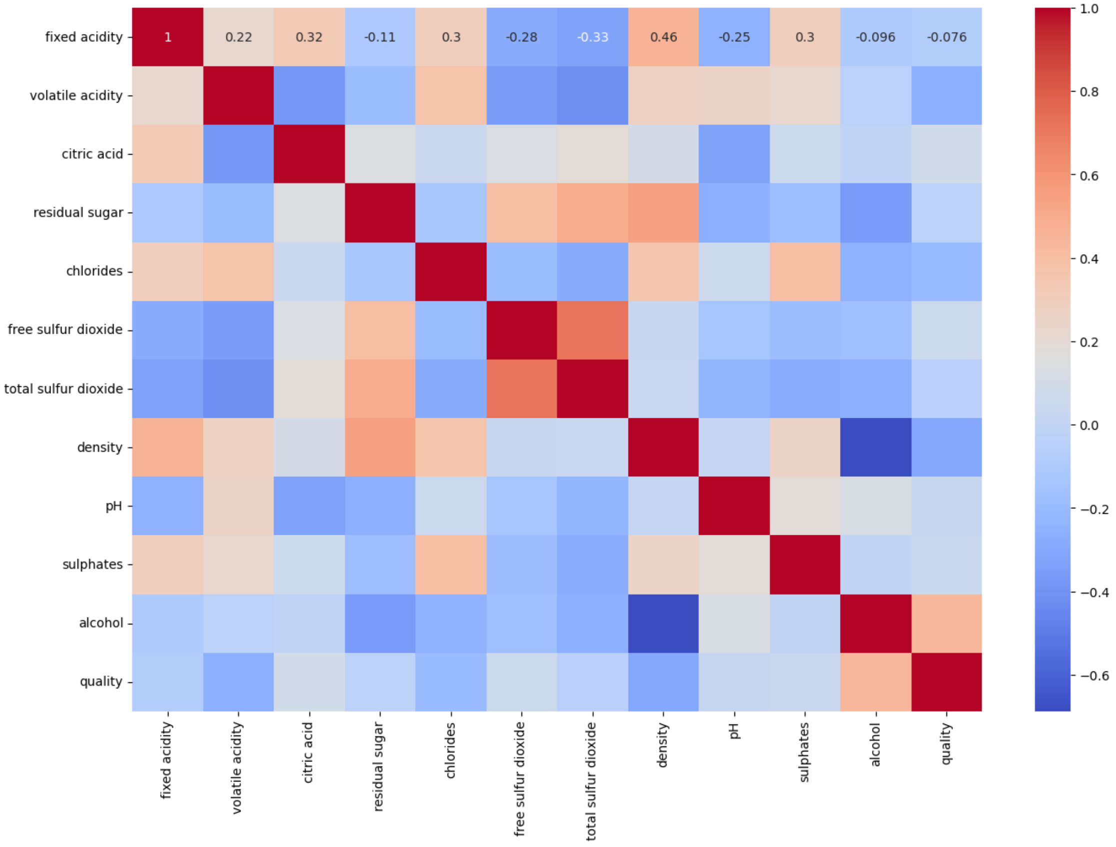
[10]:
```
sns.countplot(x='quality', data=df)
```

[10]:
```
<Axes: xlabel='quality', ylabel='count'>
```



[11]:
```
plt.figure(figsize = (15,10))
sns.heatmap(df.corr(numeric_only=True), cmap = 'coolwarm', annot = True)
```

[11]:
```
<Axes: >
```



# 1 Feature Engineering

[12]:
```
# Alcohol is mediumly +ve correlated to quality
# Density is mediumly -ve correlated to quality
# A possible feature could be alcohol/density which also remodensityves heavy␣
 ↪correlation between alc and den
# Also remove free sulfur dioxide as its very correlated to total sulfur dioxide
# Removing former not latter because latter is more correlated to quality

df_new = df.drop('free sulfur dioxide', axis = 1)
```

[13]:
```
df_new['alcohol density'] = (df_new['alcohol']**5)/df_new['density']
```

[14]:
```
df_new.head()
```

[14]:
```
    type  fixed acidity  volatile acidity  citric acid  residual sugar  \
0  white            7.0              0.27         0.36            20.7
1  white            6.3              0.30         0.34             1.6
2  white            8.1              0.28         0.40             6.9
```

```
3  white            7.2              0.23         0.32            8.5
4  white            7.2              0.23         0.32            8.5

   chlorides  total sulfur dioxide  density    pH  sulphates  alcohol  \
0      0.045                 170.0   1.0010  3.00       0.45      8.8
1      0.049                 132.0   0.9940  3.30       0.49      9.5
2      0.050                  97.0   0.9951  3.26       0.44     10.1
3      0.058                 186.0   0.9956  3.19       0.40      9.9
4      0.058                 186.0   0.9956  3.19       0.40      9.9

   quality  alcohol density
0        6     52720.471209
1        6     77845.164738
2        6    105618.535836
3        6     95519.289865
4        6     95519.289865
```

[15]:
```
df_ml = pd.get_dummies(df_new, drop_first=True)
```

[16]:
```
df_ml.head()
```

[16]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.0              0.27         0.36            20.7      0.045
1            6.3              0.30         0.34             1.6      0.049
2            8.1              0.28         0.40             6.9      0.050
3            7.2              0.23         0.32             8.5      0.058
4            7.2              0.23         0.32             8.5      0.058

   total sulfur dioxide  density    pH  sulphates  alcohol  quality  \
0                 170.0   1.0010  3.00       0.45      8.8        6
1                 132.0   0.9940  3.30       0.49      9.5        6
2                  97.0   0.9951  3.26       0.44     10.1        6
3                 186.0   0.9956  3.19       0.40      9.9        6
4                 186.0   0.9956  3.19       0.40      9.9        6

   alcohol density  type_white
0     52720.471209        True
1     77845.164738        True
2    105618.535836        True
3     95519.289865        True
4     95519.289865        True
```

[17]:
```
df_ml = df_ml.drop(['density', 'alcohol'], axis = 1)
```

[18]:
```
df_ml.head()
```

```
[18]:    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
      0            7.0              0.27         0.36            20.7      0.045
      1            6.3              0.30         0.34             1.6      0.049
      2            8.1              0.28         0.40             6.9      0.050
      3            7.2              0.23         0.32             8.5      0.058
      4            7.2              0.23         0.32             8.5      0.058

         total sulfur dioxide    pH  sulphates  quality  alcohol density  type_white
      0                 170.0  3.00       0.45        6     52720.471209        True
      1                 132.0  3.30       0.49        6     77845.164738        True
      2                  97.0  3.26       0.44        6    105618.535836        True
      3                 186.0  3.19       0.40        6     95519.289865        True
      4                 186.0  3.19       0.40        6     95519.289865        True
```

```
[19]: df_ml.isnull().sum()
```

```
[19]: fixed acidity           0
      volatile acidity        0
      citric acid             0
      residual sugar          0
      chlorides               0
      total sulfur dioxide    0
      pH                      0
      sulphates               0
      quality                 0
      alcohol density         0
      type_white              0
      dtype: int64
```

```
[20]: Y = df_ml['quality'].apply(lambda y: 1 if y>=6 else 0)
      Y
```

```
[20]: 0       1
      1       1
      2       1
      3       1
      4       1
             ..
      6491    1
      6492    0
      6494    1
      6495    0
      6496    1
      Name: quality, Length: 6463, dtype: int64
```

```
[21]: X = df_ml.drop('quality', axis = 1)
      X.head()
```

```
[21]:    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
      0            7.0              0.27         0.36            20.7      0.045
      1            6.3              0.30         0.34             1.6      0.049
      2            8.1              0.28         0.40             6.9      0.050
      3            7.2              0.23         0.32             8.5      0.058
      4            7.2              0.23         0.32             8.5      0.058

         total sulfur dioxide    pH  sulphates  alcohol density  type_white
      0                 170.0  3.00       0.45     52720.471209        True
      1                 132.0  3.30       0.49     77845.164738        True
      2                  97.0  3.26       0.44    105618.535836        True
      3                 186.0  3.19       0.40     95519.289865        True
      4                 186.0  3.19       0.40     95519.289865        True
```

```
[22]: # Standardize feature values so that high valued feautures don't influence␣
       ↪others

      from sklearn.preprocessing import StandardScaler

      scaler = StandardScaler()
      scaler.fit(X)
      X_standard = scaler.transform(X)

      X_standard #standardised numpy array of features
```

```
[22]: array([[-0.16778609, -0.42270958,  0.2839587 , …, -0.5449872 ,
              -1.03523983,  0.5719307 ],
             [-0.70715516, -0.2404789 ,  0.14625658, …, -0.27635393,
              -0.75614416,  0.5719307 ],
             [ 0.67979387, -0.36196602,  0.55936296, …, -0.61214551,
              -0.44762587,  0.5719307 ],
             …,
             [-0.70715516,  1.03513588, -1.29961576, …,  1.46976231,
               0.1757951 , -1.74846359],
             [-1.01536606,  1.85517396, -1.36846682, …,  1.20112905,
              -0.38884046, -1.74846359],
             [-0.93831333, -0.17973534,  1.0413204 , …,  0.86533746,
               0.1762463 , -1.74846359]])
```

```
[23]: X = X_standard
```

```
[24]: from sklearn.model_selection import train_test_split
```

```
[25]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1,␣
       ↪random_state = 25)
```

```
[26]: from sklearn.ensemble import RandomForestClassifier

      rfc = RandomForestClassifier()
      rfc.fit(X_train, Y_train)
      rfc_pred = rfc.predict(X_test)
```

```
[27]: from sklearn.metrics import accuracy_score, classification_report,␣
        ↪confusion_matrix
```

```
[28]: accuracy_score(Y_test,rfc_pred)
```

```
[28]: 0.8408037094281299
```

```
[29]: print(classification_report(Y_test, rfc_pred))
```

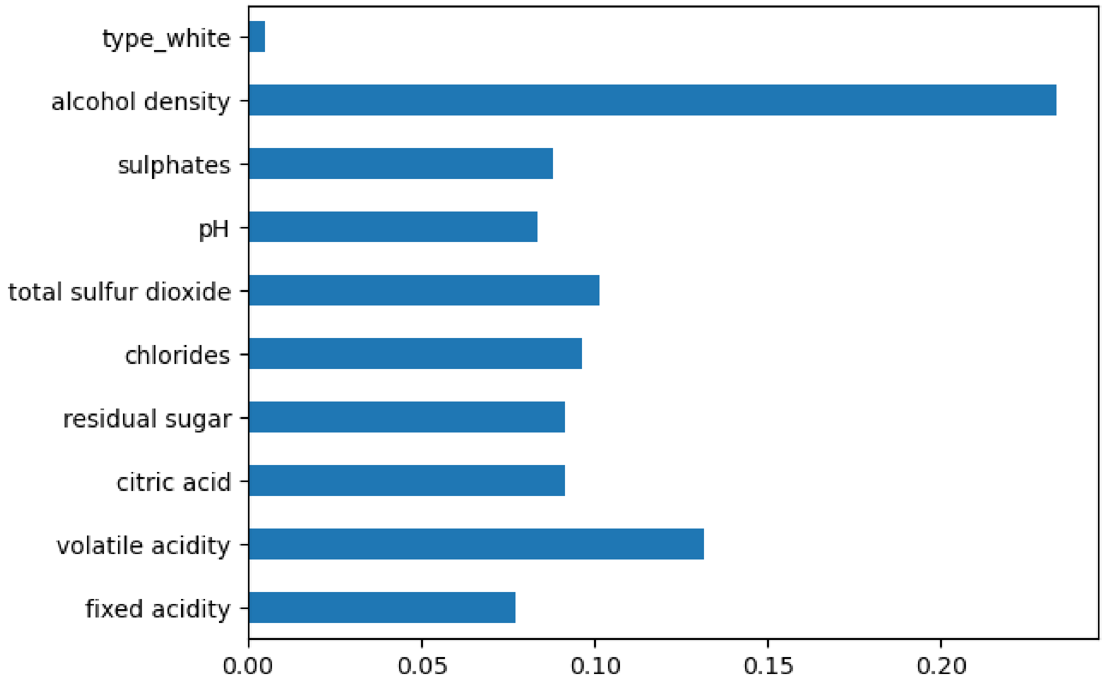
```
              precision    recall  f1-score   support

           0       0.82      0.74      0.78       247
           1       0.85      0.90      0.88       400

    accuracy                           0.84       647
   macro avg       0.84      0.82      0.83       647
weighted avg       0.84      0.84      0.84       647
```

```
[30]: print(confusion_matrix(Y_test, rfc_pred))
```

```
[[183  64]
 [ 39 361]]
```

```
[526]: pd.Series(rfc.feature_importances_, index=df_ml.drop('quality', axis = 1).
         ↪columns).plot(kind = 'barh')
```

```
[526]: <Axes: >
```



```
[527]: from sklearn.ensemble import GradientBoostingClassifier
```

```
[528]: gbc = GradientBoostingClassifier().fit(X_train, Y_train)
```

```
[529]: gbc_pred = gbc.predict(X_test)
```

```
[530]: accuracy_score(Y_test, gbc_pred)
```

```
[530]: 0.7511591962905718
```

```
[531]: print(classification_report(Y_test, gbc_pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.58      0.63       242
           1       0.77      0.85      0.81       405

    accuracy                           0.75       647
   macro avg       0.74      0.72      0.72       647
weighted avg       0.75      0.75      0.75       647
```

```
[542]: i_fixed_acidity = float(input("Enter fixed acidity [3.8-15.9]: "))
       i_volatile_acidity = float(input("Enter volatile acidity [0.08-1.58]: "))
```

```
i_citric_acid = float(input("Enter citric acid [0.0-1.66]: "))
i_residual_sugar = float(input("Enter residual sugar [0.6-65.8]: "))
i_chlorides = float(input("Enter chlorides [0.009-0.611]: "))
i_total_sulfur_dioxide = float(input("Enter total sulfur dioxide [6.0-440.0]:␣
 ↪"))
i_density = float(input("Enter density [0.98-1.04]: "))
i_ph = float(input("Enter pH [2.72-4.01]: "))
i_sulphates = float(input("Enter sulphates [0.22-2.0]: "))
i_type_white = bool(input("Enter 1 if wine is white, 0 if red: "))
i_alcohol = float(input("Enter alcohol percentage [8.0-14.9]: "))

i_predict = np.array([i_fixed_acidity, i_volatile_acidity, i_citric_acid,␣
 ↪i_residual_sugar, i_chlorides, i_total_sulfur_dioxide, i_ph, i_sulphates,␣
 ↪((i_alcohol**5)/i_density), i_type_white]).reshape(1,-1)
scaler.fit(i_predict)

i_predict_std = scaler.transform(i_predict)

predict_output = rfc.predict(i_predict_std)
predict_output_int = predict_output[0]
print(predict_output_int)
```

```
Enter fixed acidity [3.8-15.9]:  4
Enter volatile acidity [0.08-1.58]:  1
Enter citric acid [0.0-1.66]:  1
Enter residual sugar [0.6-65.8]:  2
Enter chlorides [0.009-0.611]:  0.5
Enter total sulfur dioxide [6.0-440.0]:  22
Enter density [0.98-1.04]:  1
Enter pH [2.72-4.01]:  2
Enter sulphates [0.22-2.0]:  1.2
Enter 1 if wine is white, 0 if red:  1
Enter alcohol percentage [8.0-14.9]:  17

1
```

[ ]:

```
#6.7        0.23        0.31        2.1        0.046        30.0        0.
 ↪99260        3.33        0.64        10.7
```

[1]:

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 1
----> 1 predict_output_int = int(predict_output[0])
      2 print(predict_output_int)
```

```
NameError: name 'predict_output' is not defined
```

```
[ ]:
```