```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/content/drive/MyDrive/1788410-1767134-1729261-1613779-Red_wine__(1).csv')
print(df)
```

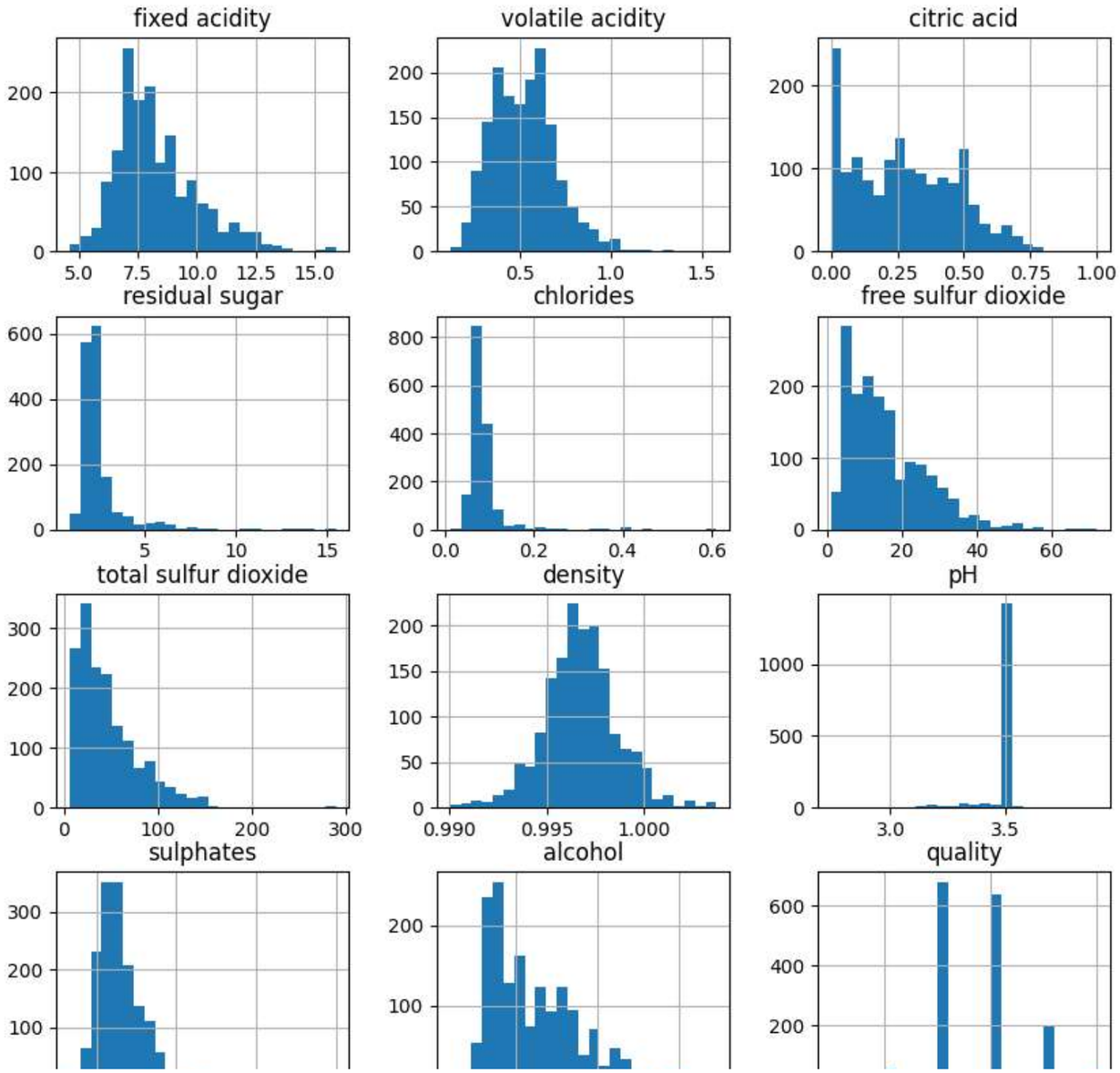
```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.52       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.52       0.75
1597                 32.0                  44.0  0.99547  3.52       0.71
1598                 18.0                  42.0  0.99549  3.52       0.66

      alcohol  quality
0         9.4      5.0
1         9.8      5.0
2         9.8      5.0
3         9.8      6.0
4         9.4      5.0
...       ...      ...
1594     10.5      5.0
1595     11.2      6.0
1596     11.0      6.0
1597     10.2      5.0
1598     11.0      6.0

[1599 rows x 12 columns]
```

```
df.hist(bins=25,figsize=(10,10))
# display histogram
plt.show()
```




```
plt.figure(figsize=[10,6])
# plot bar graph
plt.bar(df['quality'],df['alcohol'],color='red')
# label x-axis
plt.xlabel('quality')
#label y-axis
plt.ylabel('alcohol')
```

```
Text(0, 0.5, 'alcohol')
```


```
# ploting heatmap
plt.figure(figsize=[19,10],facecolor='blue')
hm = sns.heatmap(df.corr(),annot=True)
```

```
for a in range(len(df.corr().columns)):
    for b in range(a):
        if abs(df.corr().iloc[a,b]) >0.7:
            name = df.corr().columns[a]
            print(name)


new_df=df.drop('total sulfur dioxide',axis=1)


  new_df.isnull().sum()

     fixed acidity          0
     volatile acidity       0
     citric acid            0
     residual sugar         0
     chlorides              0
     free sulfur dioxide    0
     density                0
     pH                     1
     sulphates              0
     alcohol                0
     quality                1
     dtype: int64


new_df.update(new_df.fillna(new_df.mean()))


# catogerical vars
next_df = pd.get_dummies(new_df,drop_first=True)
# display new dataframe
next_df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	0.99780	3.51	0.56	9.4	5.0
1	7.8	0.880	0.00	2.6	0.098	25.0	0.99680	3.20	0.68	9.8	5.0
2	7.8	0.760	0.04	2.3	0.092	15.0	0.99700	3.26	0.65	9.8	5.0
3	11.2	0.280	0.56	1.9	0.075	17.0	0.99800	3.16	0.58	9.8	6.0
4	7.4	0.700	0.00	1.9	0.076	11.0	0.99780	3.51	0.56	9.4	5.0
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	0.99490	3.52	0.58	10.5	5.0
1595	5.9	0.550	0.10	2.2	0.062	39.0	0.99512	3.52	0.76	11.2	6.0
1596	6.3	0.510	0.13	2.3	0.076	29.0	0.99574	3.52	0.75	11.0	6.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	0.99547	3.52	0.71	10.2	5.0
1598	6.0	0.310	0.47	3.6	0.067	18.0	0.99549	3.52	0.66	11.0	6.0

1599 rows × 11 columns

```
next_df['best quality'] = [ 1 if x>=7 else 0 for x in df.quality]
print(next_df)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  density    pH  sulphates  alcohol  quality  \
0                    11.0  0.99780  3.51       0.56      9.4      5.0
1                    25.0  0.99680  3.20       0.68      9.8      5.0
2                    15.0  0.99700  3.26       0.65      9.8      5.0
3                    17.0  0.99800  3.16       0.58      9.8      6.0
4                    11.0  0.99780  3.51       0.56      9.4      5.0
...                   ...      ...   ...        ...      ...      ...
1594                 32.0  0.99490  3.52       0.58     10.5      5.0
1595                 39.0  0.99512  3.52       0.76     11.2      6.0
1596                 29.0  0.99574  3.52       0.75     11.0      6.0
1597                 32.0  0.99547  3.52       0.71     10.2      5.0
1598                 18.0  0.99549  3.52       0.66     11.0      6.0

      best quality
0                0
1                0
2                0
3                0
4                0
...            ...
1594             0
1595             0
1596             0
1597             0
1598             0

[1599 rows x 12 columns]
```

```
x =next_df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']].values
y=next_df[['quality']].values
print(x,y)
```

```
[[ 7.4    0.7    0.    ...  3.51   0.56   9.4  ]
 [ 7.8    0.88   0.    ...  3.2    0.68   9.8  ]
 [ 7.8    0.76   0.04  ...  3.26   0.65   9.8  ]
 ...
 [ 6.3    0.51   0.13  ...  3.52   0.75  11.   ]
 [ 5.9    0.645  0.12  ...  3.52   0.71  10.2  ]
 [ 6.     0.31   0.47  ...  3.52   0.66  11.   ]] [[5.]
 [5.]
 [5.]
 ...
 [6.]
 [5.]
 [6.]]
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)


print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1279, 10)
(320, 10)
(1279, 1)
(320, 1)
```

```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()


reg.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
m=reg.coef_
c=reg.intercept_
m,c
```

```
(array([[ 7.61299576e-02, -1.11782342e+00, -3.38472674e-01,
          1.52143544e-02, -1.24354762e+00, -1.91370810e-03,
         -4.19066394e+01,  7.06037044e-01,  9.55526413e-01,
          2.56901254e-01]]),
 array([41.77133681]))
```

```
  pred_train=reg.predict(x_train)
pred_test=reg.predict(x_test)


result=reg.predict([[7.4,0.70,0.00,1.9,0.076,11.0,0.9978,3.51,0.56,9.4]])
print(result)
```

```
[[5.07928073]]
```

```
print(pred_test)
```

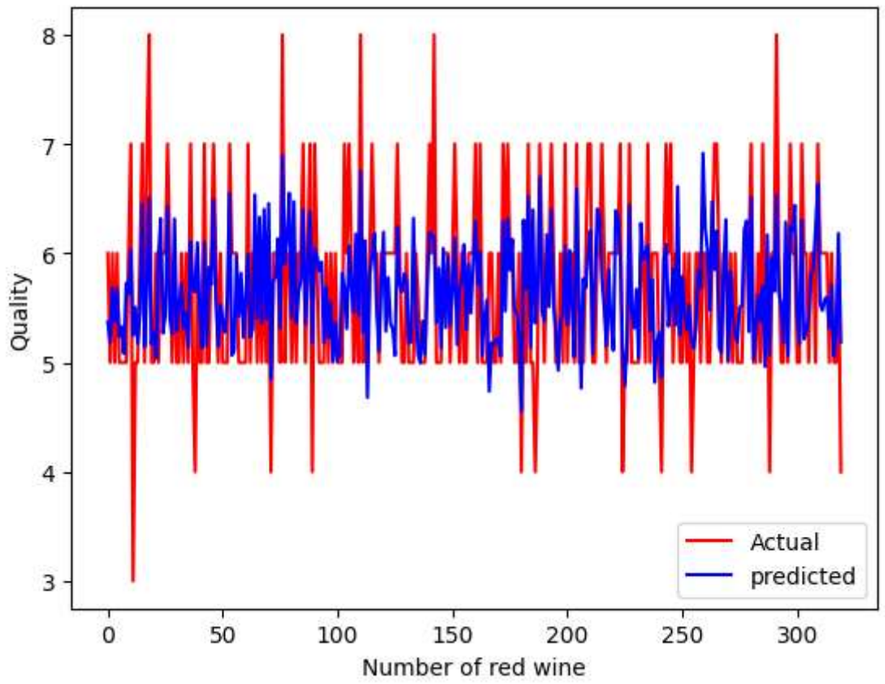
```
[5.13212069]
[5.32330805]
[5.84315814]
[5.68022258]
[6.91446337]
[6.22540645]
[6.02181849]
[5.47810719]
[6.47165341]
[5.85131908]
[6.20383682]
[5.14085342]
[5.08900269]
[5.86087574]
[6.30643251]
[5.01725674]
[5.82854885]
[5.40879205]
[5.24801032]
[5.18790395]
[5.50468925]
[5.51932693]
[6.21625215]
[6.29524526]
[5.28217232]
[6.51317673]
[5.03139834]
[5.29518795]
[5.61867683]
[5.38023124]
[5.69979676]
[4.96234158]
[6.16791303]
[5.07029908]
[5.95648617]
[5.65208193]
[6.53485691]
[5.64355972]
[5.52328544]
[5.18021361]
[6.28527943]
[5.06213686]
[6.2457012 ]
[6.21070236]
[6.43820173]
[5.4409003 ]
[5.1731981 ]
[6.30357192]
```

```
from sklearn.metrics import r2_score
r2_train=r2_score(y_train,pred_train)
r2_test=r2_score(y_test,pred_test)
print(r2_train)
print(r2_test)
```

```
0.3415530519610981
0.40407582696460875
```

```
plt.plot(y_test,color='red',label='Actual')
plt.plot(pred_test,color='blue',label='predicted')
plt.xlabel('Number of red wine')
```

```
plt.ylabel('Quality')
plt.legend()
plt.show()
```




```
dataset=pd.DataFrame({'Actual':y_test.flatten(),'Predicted' :pred_test.flatten()})
dataset
```

	Actual	Predicted
0	6.0	5.370998
1	5.0	5.184550
2	6.0	5.679594
3	5.0	5.371354
4	6.0	5.674634
...	...	...
315	6.0	5.710216
316	5.0	5.056126
317	5.0	5.281007
318	6.0	6.180738
319	4.0	5.185488

320 rows × 2 columns

```
dataset.head(40).plot(kind='bar')
plt.xlabel('number of red wine')
```

```
plt.ylabel('Quality')
```

```
Text(0, 0.5, 'Quality')
```