```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[100] df=pd.read_csv("/content/sample_data/winequalityN.csv")
      df
```

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	white	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.45	8.8	6
1	white	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.49	9.5	6
2	white	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.44	10.1	6
3	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
4	white	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.40	9.9	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6492	red	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5

Wine_quality_prediction.ipynb - Google Colaboratory

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

Wine_quality_prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
df.describe()
```

0s

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	qua
count	6487.000000	6489.000000	6494.000000	6495.000000	6495.000000	6497.000000	6497.000000	6497.000000	6488.000000	6493.000000	6497.000000	6497.0(
mean	7.216579	0.339691	0.318722	5.444326	0.056042	30.525319	115.744574	0.994697	3.218395	0.531215	10.491801	5.8
std	1.296750	0.164649	0.145265	4.758125	0.035036	17.749400	56.521855	0.002999	0.160748	0.148814	1.192712	0.8
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000000	0.987110	2.720000	0.220000	8.000000	3.0(
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000000	0.992340	3.110000	0.430000	9.500000	5.0(
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000000	0.994890	3.210000	0.510000	10.300000	6.0(
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000000	0.996990	3.320000	0.600000	11.300000	6.0(
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000000	1.038980	4.010000	2.000000	14.900000	9.0(

✓ 1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:16 08-02-2023

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   type                  6497 non-null   object 
 1   fixed acidity         6487 non-null   float64
 2   volatile acidity      6489 non-null   float64
 3   citric acid           6494 non-null   float64
 4   residual sugar        6495 non-null   float64
 5   chlorides             6495 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6488 non-null   float64
 10  sulphates             6493 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64  
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB
```

Wine_quality_prediction.ipynb - Google Colaboratory

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

Wine_quality_prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

[104]
```
12  quality               6497 non-null   int64
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB
```

[ ]
```
df.isna().sum()
```

```
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
```

[106]
```
# create dist plot
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:17 08-02-2023

Wine_quality_prediction.ipynb - Google Colaboratory

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

Wine_quality_prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

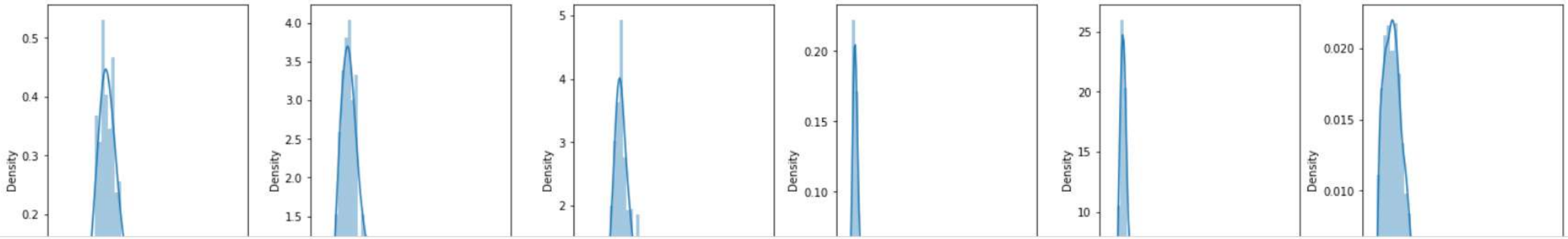
+ Code + Text

RAM Disk

```
# create dist plot
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')

fig, ax = plt.subplots(ncols=6, nrows=2, figsize=(20,10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    if col != 'type':
        sns.distplot(value, ax=ax[index])
        index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:17 08-02-2023

Wine_quality_prediction.ipynb

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

Wine_quality_prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

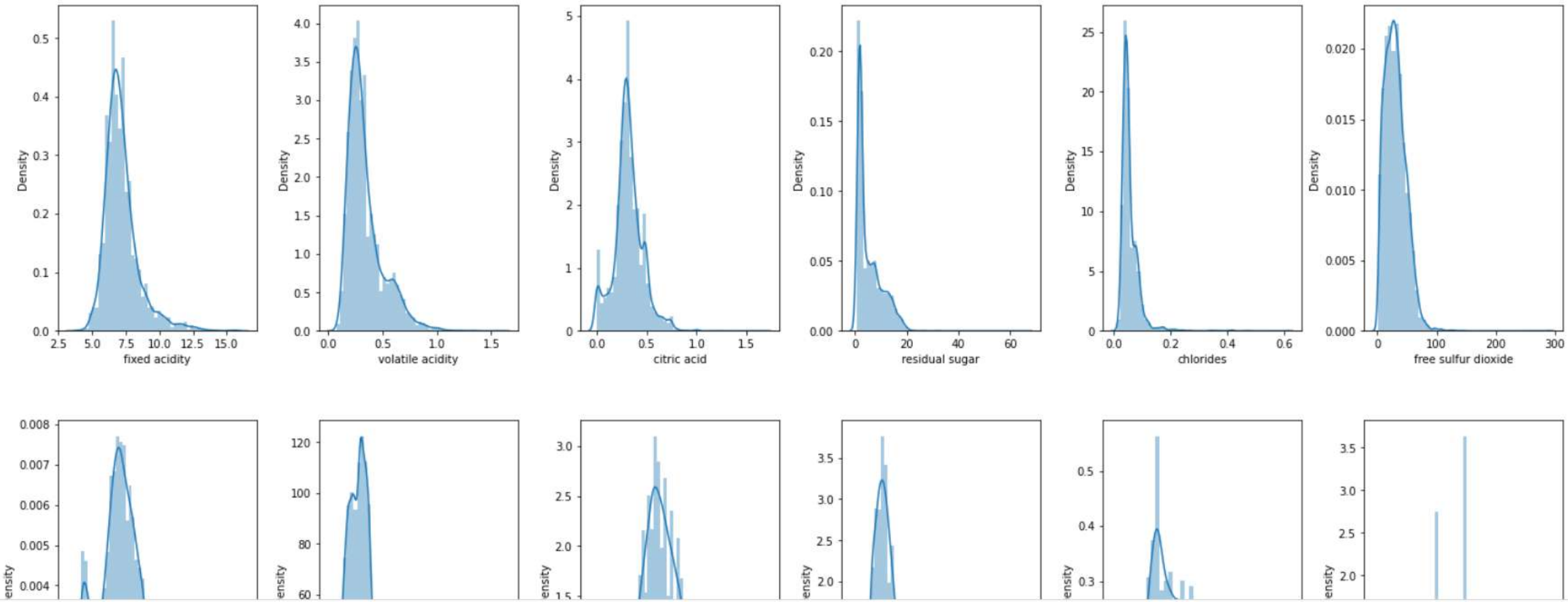
Comment Share

+ Code + Text

RAM Disk

```
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```

6s



1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:17 08-02-2023

Wine_quality_prediction.ipynb - Google Colaboratory

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

Wine_quality_prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

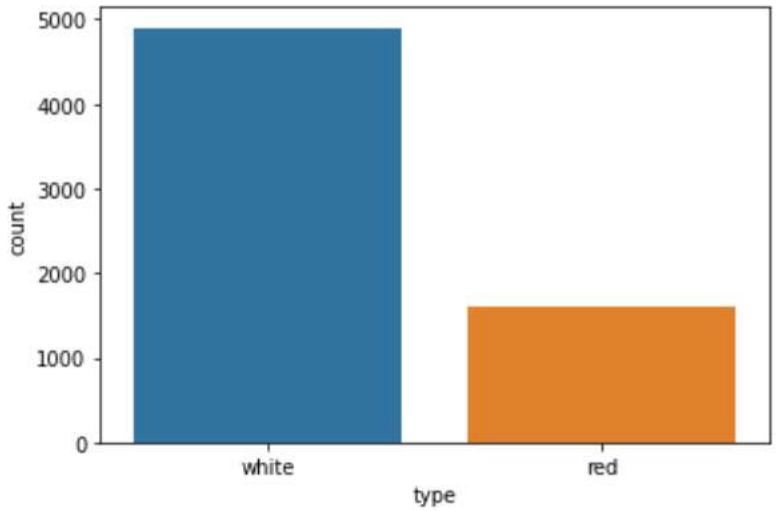
+ Code + Text

RAM Disk

```
sns.countplot('type',data=df)
```

0s

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f890ee4cac0>
```




```
[108] sns.jointplot(x='free sulfur dioxide',y='total sulfur dioxide',data=df,kind='reg')
```

2s

```
<seaborn.axisgrid.JointGrid at 0x7f890d0b30d0>
```

1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:17 08-02-2023

Wine_quality_prediction.ipynb -

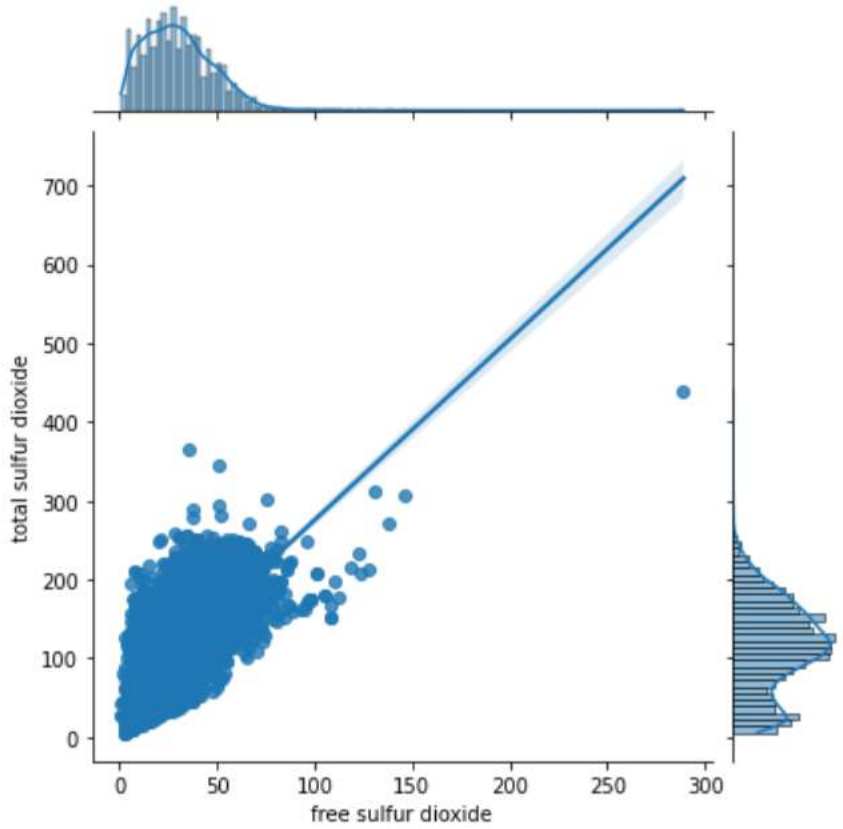
colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
sns.jointplot(x='free sulfur dioxide',y='total sulfur dioxide',data=df,kind='reg')
```

```
<seaborn.axisgrid.JointGrid at 0x7f890d0b30d0>
```



1s completed at 9:12 AM

76°F Sunny

Search

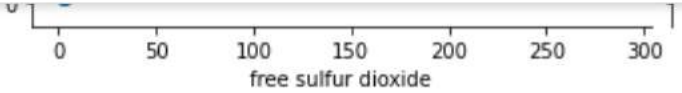
ENG IN

09:18 08-02-2023

+ Code + Text

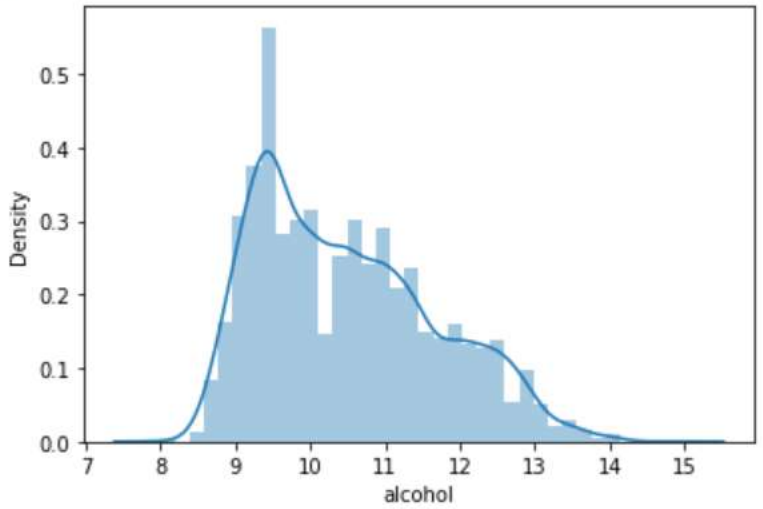
RAM Disk

[108]




```
sns.distplot(df['alcohol'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f894e0ea910>
```




```
[110] sns.countplot(df['quality'])
```

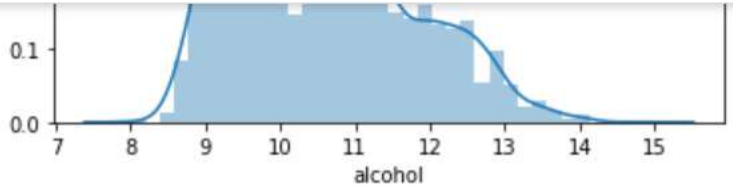
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8909ebc940>
```

1s completed at 9:12 AM

+ Code + Text

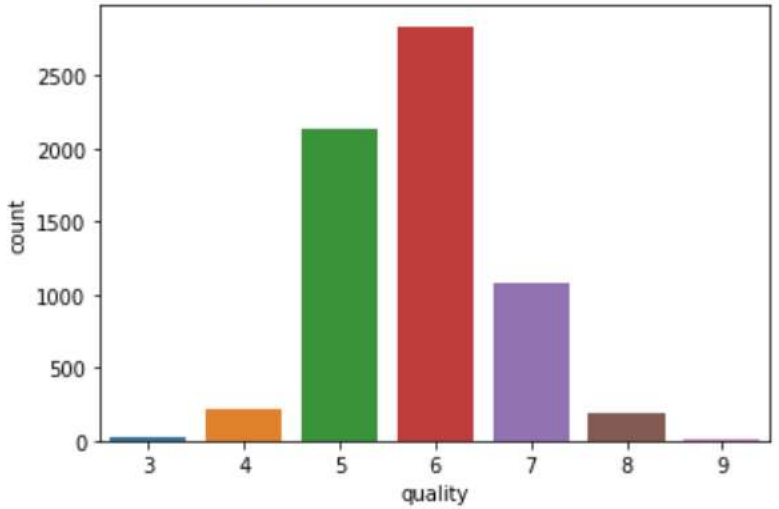
RAM Disk

[109]




```
sns.countplot(df['quality'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8909ebc940>
```




```
[111] #coorelation matrix
      corr=df.corr()
      plt.figure(figsize=(20,10))
```

1s completed at 9:12 AM

Wine_quality_prediction.ipynb

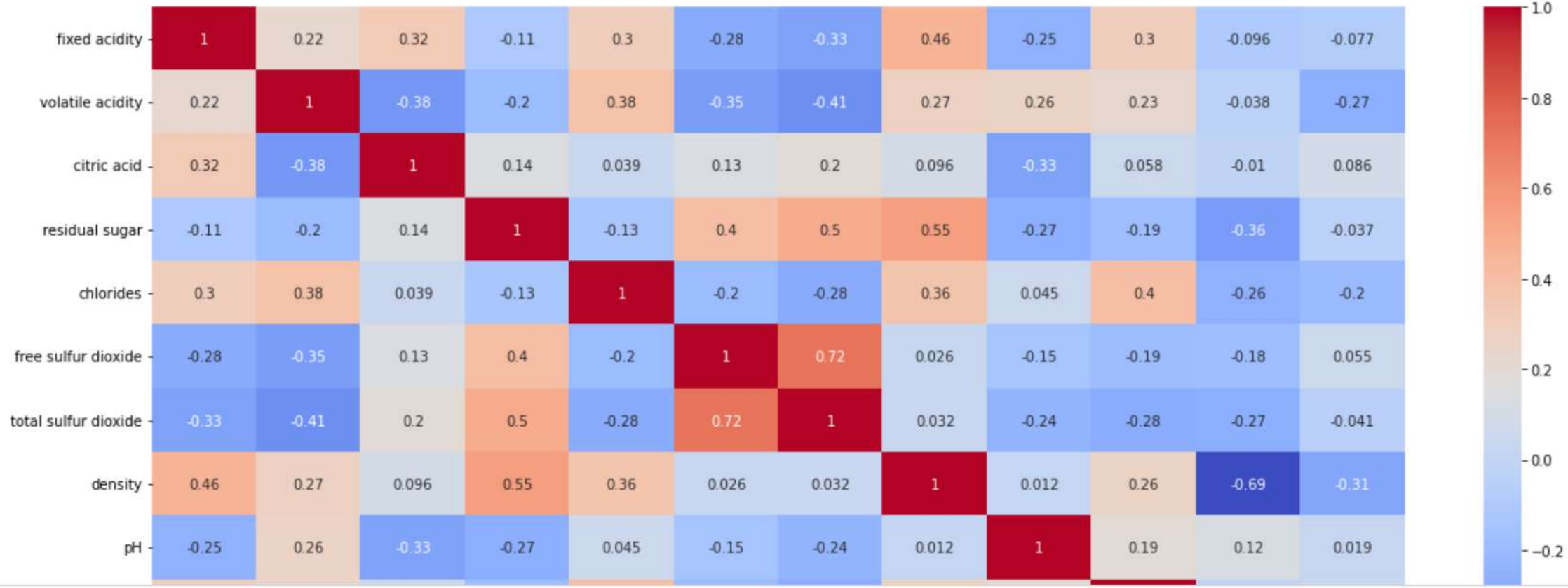
colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
#coorelation matrix
corr=df.corr()
plt.figure(figsize=(20,10))
sns.heatmap(corr,annot=True,cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f890d0f2eb0>
```



1s completed at 9:12 AM

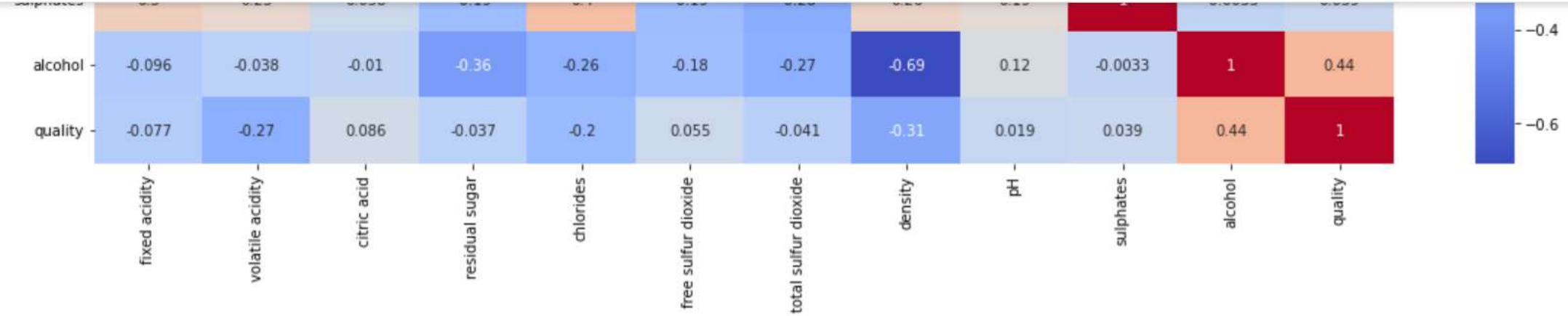
76°F Sunny

Search

ENG IN

09:19 08-02-2023

[111]




```
df.isna().sum()
```

0s

```
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
```

✓ 1s completed at 9:12 AM

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
[113] df['fixed acidity']=df['fixed acidity'].fillna(df['fixed acidity'].mean())
      df['volatile acidity']=df['volatile acidity'].fillna(df['volatile acidity'].mean())
      df['citric acid']=df['citric acid'].fillna(df['citric acid'].mean())
      df['residual sugar']=df['residual sugar'].fillna(df['residual sugar'].mean())
      df['chlorides']=df['chlorides'].fillna(df['chlorides'].mean())
      df['pH']=df['pH'].fillna(df['pH'].mean())
      df['sulphates']=df['sulphates'].fillna(df['sulphates'].mean())
```

```
df.isna().sum()
```

```
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
```

```
[115] df= df.drop(['type'],axis =1)
```

1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:19 08-02-2023

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
df= df.drop(['type'],axis =1)
df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	0.450000	8.8	6
1	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	0.490000	9.5	6
2	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	0.440000	10.1	6
3	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.400000	9.9	6
4	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	0.400000	9.9	6
...	...	...	...	...	...	...	...	...	...	...	...	...
6492	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.580000	10.5	5
6493	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.531215	11.2	6
6494	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.750000	11.0	6
6495	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.710000	10.2	5
6496	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.660000	11.0	6

6497 rows × 12 columns

1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:19 08-02-2023

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
[116] x=df.iloc[:,:-1].values
      y=df.iloc[:,-1].values
      y
```

```
array([6, 6, 6, ..., 6, 5, 6])
```

```
[117] from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=42)
      y_test
```

```
array([7, 7, 6, ..., 6, 6, 6])
```

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
x_test
```

```
array([[-0.17304369, -0.54359665,  0.90004528, ..., -0.36283779,
        -0.35375297,  1.17959322],
       [ 0.28607349, -1.20786028,  2.89927908, ..., -0.92242828,
        -0.88352406,  0.256953  ],
       [-0.78519994, -1.14747268, -0.34085846, ..., -0.61154467,
        -1.01596683, -1.16894553],
       ...,
```

1s completed at 9:12 AM

76°F Sunny

Search

ENG IN

09:20 08-02-2023

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

[119] 0s
```
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier(criterion='entropy')
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

```
array([7, 7, 6, ..., 5, 6, 6])
```

0s
```
df1=pd.DataFrame({'Actual_Value':y_test,'Predicted_Value':y_pred})
df1
```

	Actual_Value	Predicted_Value
0	7	7
1	7	7
2	6	6
3	6	6
4	5	5
...	...	...
1295	7	7
1296	5	6
1297	6	5

1s completed at 9:12 AM

81°F Mostly sunny

Search

ENG IN

09:20 08-02-2023

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
[121] from sklearn.metrics import classification_report,accuracy_score,confusion_matrix,ConfusionMatrixDisplay
      result=confusion_matrix(y_test,y_pred)
      score=accuracy_score(y_test,y_pred)
      score
```

```
0.6138461538461538
```

```
      from sklearn.neighbors import KNeighborsClassifier
      classifier=KNeighborsClassifier(n_neighbors=7)
      classifier.fit(x_train,y_train)
      y_pred=classifier.predict(x_test)
      print(y_pred)
```

```
[7 6 6 ... 5 6 6]
```

```
[123] from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
      result=confusion_matrix(y_test,y_pred)
      score=accuracy_score(y_test,y_pred)
      score
```

```
0.5553846153846154
```

```
[125] from sklearn.svm import SVC
      classifier=SVC()
      classifier.fit(x_train,y_train)
      y_pred=classifier.predict(x_test)
```

1s completed at 9:12 AM

81°F Mostly sunny

Search

ENG IN

09:20 08-02-2023

Wine_quality_prediction.ipynb -

colab.research.google.com/drive/1utQ1qDQ38ZI8sMTF1RnWhFAso27eMKP8?authuser=1#scrollTo=Q5hvxai84dqF

+ Code + Text

RAM Disk

```
[125] from sklearn.svm import SVC
      classifier=SVC()
      classifier.fit(x_train,y_train)
      y_pred=classifier.predict(x_test)
      print(y_pred)
```

```
[6 6 6 ... 5 6 6]
```

```
[126] from sklearn.metrics import classification_report,accuracy_score,confusion_matrix,ConfusionMatrixDisplay
      result=confusion_matrix(y_test,y_pred)
      score=accuracy_score(y_test,y_pred)
      score
```

```
0.5815384615384616
```

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators = 100)
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)

from sklearn import metrics
print()

print("Accuracy_Score: ", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy_Score:  0.6907692307692308
```

1s completed at 9:12 AM

81°F Mostly sunny

Search

ENG IN

09:21 08-02-2023