

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

```
In [2]: df = pd.read_csv('wine_fraud.csv')
```

```
In [3]: df.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4

```
In [4]: df['type'].unique()
```

Out[4]: array(['red', 'white'], dtype=object)

```
In [5]: df['quality'].unique()
```

Out[5]: array(['Legit', 'Fraud'], dtype=object)

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

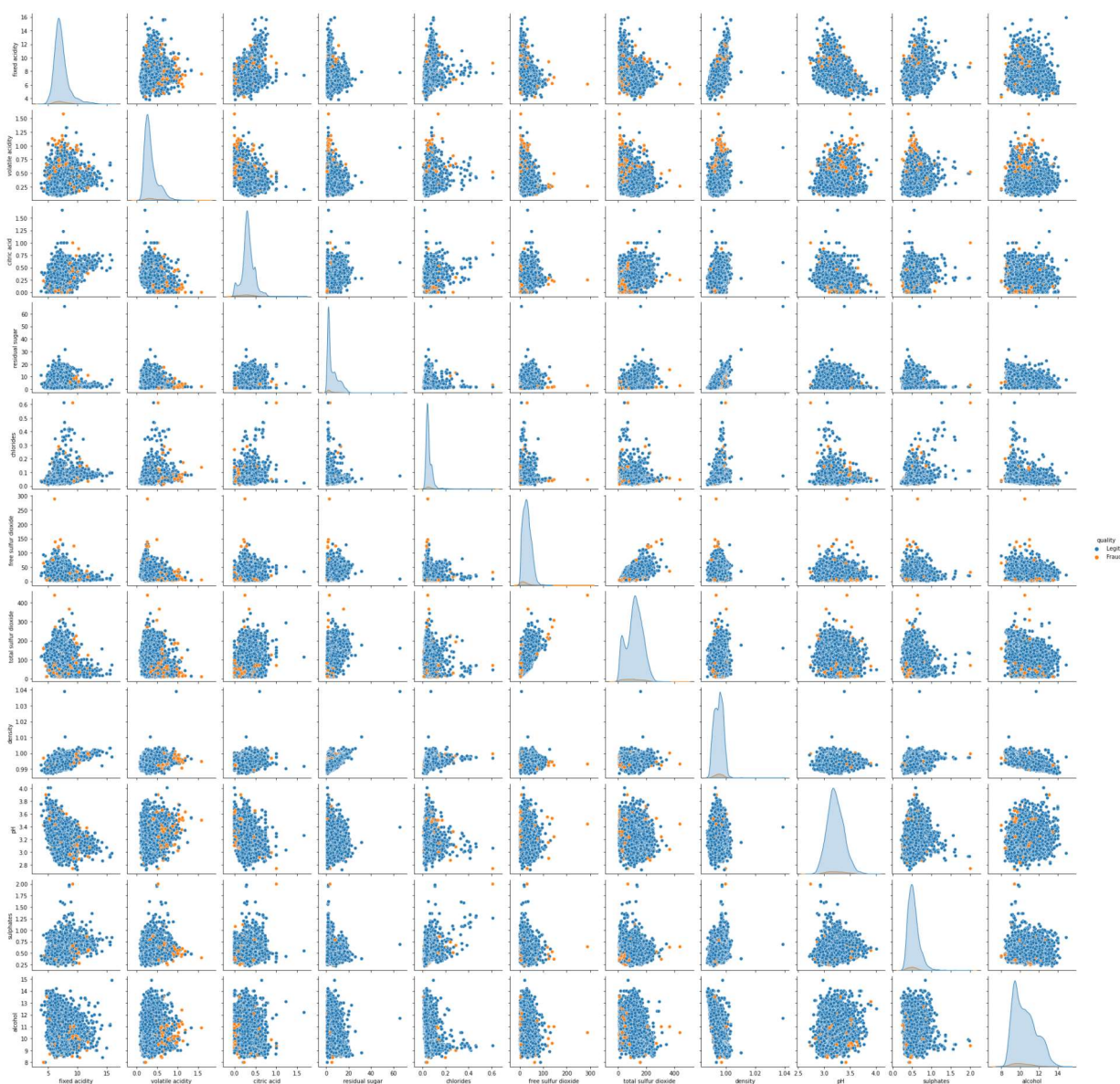
```
Out[6]: fixed acidity      0
volatile acidity      0
citric acid           0
residual sugar        0
chlorides             0
free sulfur dioxide   0
total sulfur dioxide  0
density              0
pH                  0
sulphates            0
alcohol              0
quality              0
type                 0
dtype: int64
```

```
In [7]: df.shape
```

Out[7]: (6497, 13)

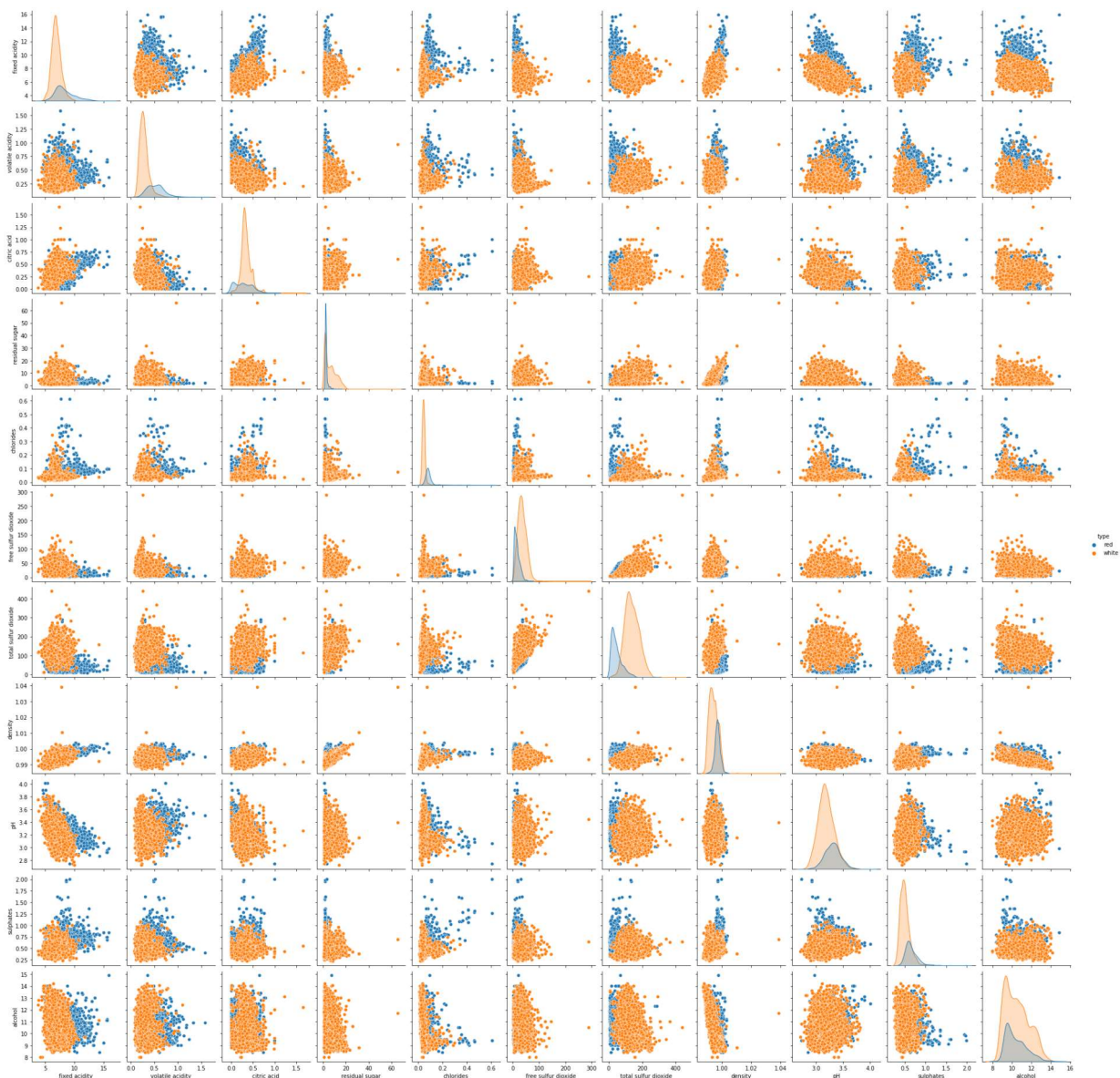
```
In [8]: sns.pairplot(data=df, hue='quality')
```

```
Out[8]: <seaborn.axisgrid.PairGrid at 0x22de8b9e850>
```



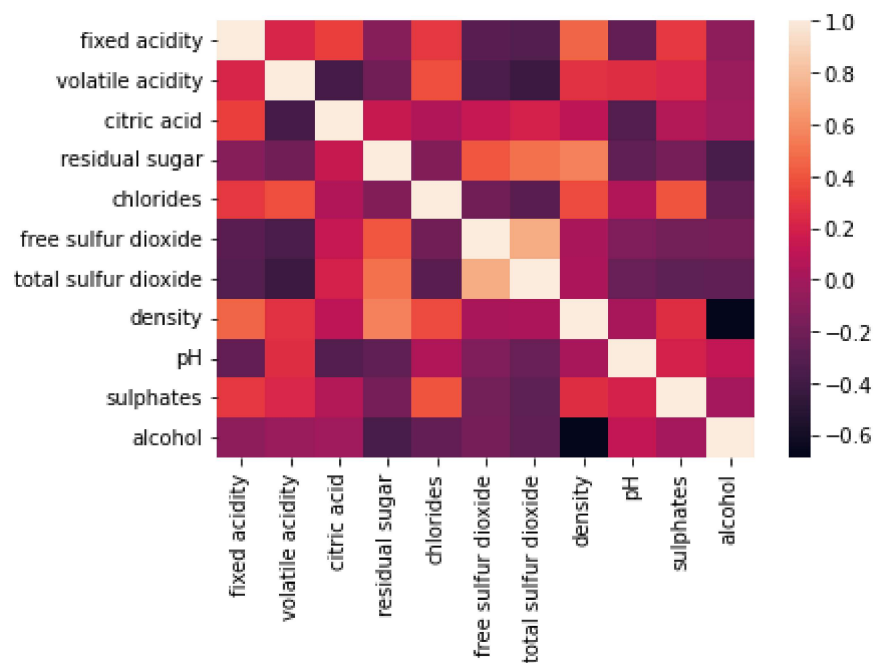
```
In [9]: sns.pairplot(data=df, hue='type')
```

```
Out[9]: <seaborn.axisgrid.PairGrid at 0x22df412ec70>
```



```
In [10]: sns.heatmap(data=df.corr())
```

```
Out[10]: <AxesSubplot:>
```

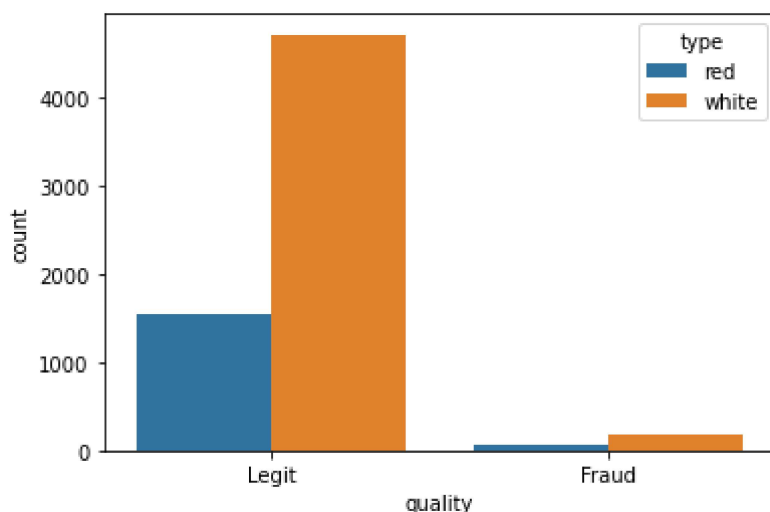


```
In [11]: sns.countplot('quality', data=df, hue='type')
```

C:\Users\aarav\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

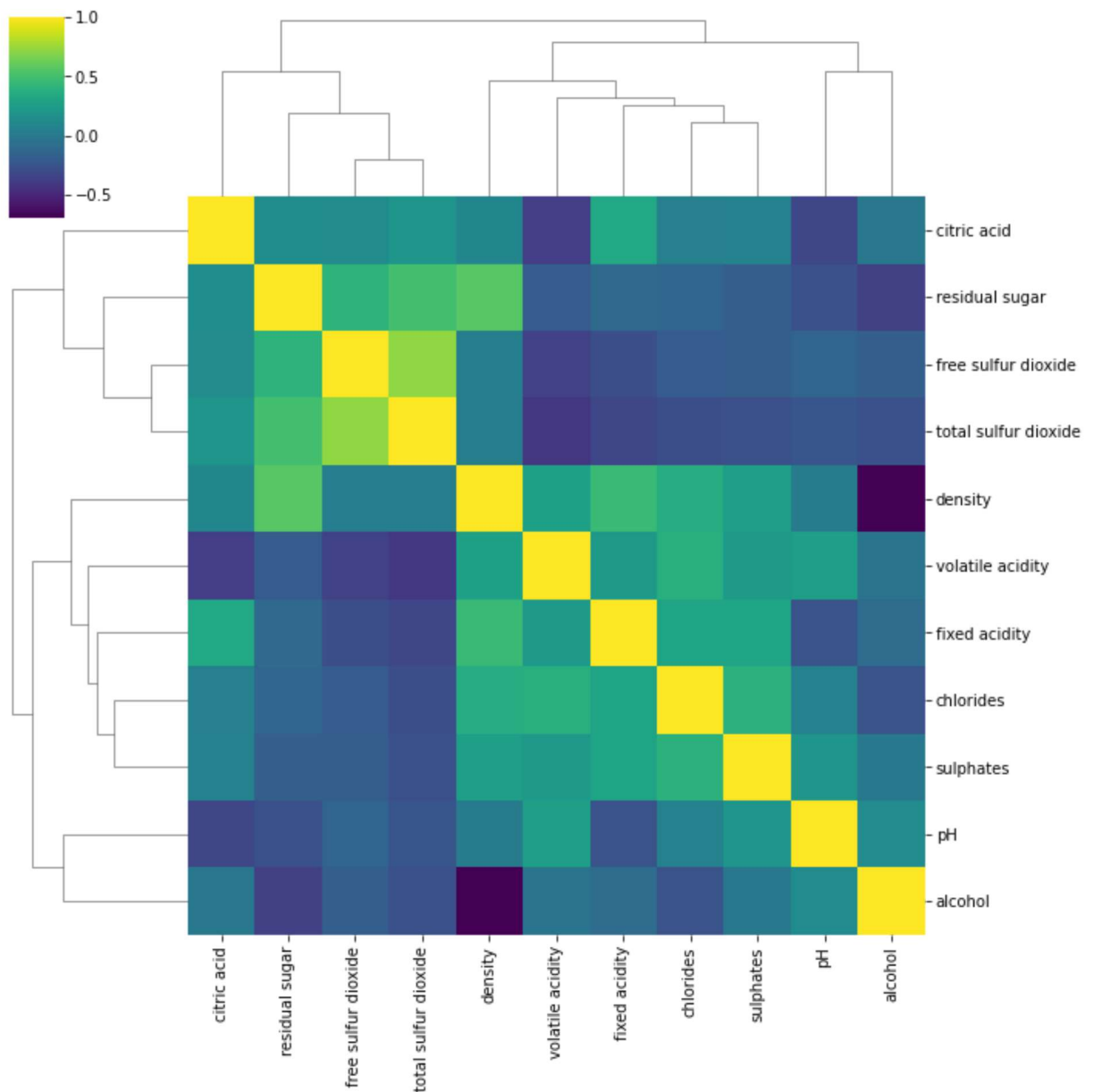
```
warnings.warn(
```

```
Out[11]: <AxesSubplot:xlabel='quality', ylabel='count'>
```



```
In [12]: sns.clustermap(df.corr(), cmap='viridis')
```

```
Out[12]: <seaborn.matrix.ClusterGrid at 0x22d8005a610>
```



```
In [13]: df['quality'] = df['quality'].map({'Legit':1,'Fraud':0})
```

```
In [14]: df['type'] = df['type'].map({'red':1,'white':0})
```

```
In [15]: X = df.drop('quality',axis=1)
```

```
In [16]: y = df['quality']
```

```
In [17]: from sklearn.model_selection import train_test_split
```

```
In [18]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_
```

```
In [19]: from sklearn.preprocessing import StandardScaler
```

```
In [20]: from sklearn.pipeline import Pipeline
```

```
In [21]: from sklearn.svm import SVC
```

```
In [22]: from sklearn.model_selection import GridSearchCV
```

```
In [23]: from sklearn.metrics import classification_report, accuracy_score, plot_confusion_m
```

```
In [24]: scaler = StandardScaler()
```

```
In [25]: svm = SVC(class_weight='balanced') #c, kernel, gamma
```

```
In [26]: operations = [('scaler', scaler), ('svm', svm)]
```

```
In [27]: pipe = Pipeline(operations)
```

```
In [28]: C = [0.001, 0.01, 0.1, 0.5, 1.0]
kernel = ['rbf', 'linear', 'poly', 'sigmoid']
gamma = ['auto', 'scaled']
```

```
In [29]: param_grid = {'svm__C': C, 'svm__kernel': kernel, 'svm__gamma': gamma}
```

```
In [30]: full_grid_classifier = GridSearchCV(pipe, param_grid)
```



```
In [31]: full_grid_classifier.fit(X_train,y_train)
```

```
C:\Users\aarav\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:372: FitFailedWarning:
100 fits failed out of a total of 200.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
```

Below are more details about the failures:

```
-----
-
100 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\aarav\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py", line 680, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\aarav\anaconda3\lib\site-packages\sklearn\pipeline.py", line 394, in fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "C:\Users\aarav\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 243, in fit
    raise ValueError(
ValueError: When 'gamma' is a string, it should be either 'scale' or 'auto'. Got 'scaled' instead.

warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\aarav\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:969: UserWarning: One or more of the test scores are non-finite: [0.03745505 0.80468908 0.95536078 0.15720924      nan      nan
      nan      nan 0.72498812 0.7916893  0.92748115 0.78176921
      nan      nan      nan      nan 0.81939491 0.78108559
0.89259298 0.61107909      nan      nan      nan      nan
0.84197144 0.78142791 0.87121771 0.55498381      nan      nan
      nan      nan 0.8546295  0.78074415 0.86728404 0.55601018
      nan      nan      nan      nan]
warnings.warn(
```

```
Out[31]: GridSearchCV(estimator=Pipeline(steps=[('scaler', StandardScaler()),
                                              ('svm', SVC(class_weight='balanced'))]),
                      param_grid={'svm__C': [0.001, 0.01, 0.1, 0.5, 1.0],
                                   'svm__gamma': ['auto', 'scaled'],
                                   'svm__kernel': ['rbf', 'linear', 'poly', 'sigmoid']})
```

```
In [32]: full_grid_classifier.best_params_
```

```
Out[32]: {'svm__C': 0.001, 'svm__gamma': 'auto', 'svm__kernel': 'poly'}
```

```
In [33]: y_pred = full_grid_classifier.predict(X_test)
```

```
In [34]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.32	0.26	0.29	27
1	0.97	0.98	0.97	623
accuracy			0.95	650
macro avg	0.64	0.62	0.63	650
weighted avg	0.94	0.95	0.94	650

```
In [35]: plot_confusion_matrix(full_grid_classifier,X_test,y_test)
```

C:\Users\aarav\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function plot\_confusion\_matrix is deprecated; Function `plot\_confusion\_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: ConfusionMatrixDisplay.from\_predictions or ConfusionMatrixDisplay.from\_estimator.

warnings.warn(msg, category=FutureWarning)

```
Out[35]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x22d80179760>
```

