```python
# importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

# importig dataset
df = pd.read_csv('winequality-red.csv')

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  3.45
0.58
1595                  39.0                  51.0  0.99512  3.52
0.76
1596                  29.0                  40.0  0.99574  3.42
0.75
1597                  32.0                  44.0  0.99547  3.57
0.71
1598                  18.0                  42.0  0.99549  3.39
0.66

      alcohol  quality
0         9.4        5
1         9.8        5
2         9.8        5
3         9.8        6
4         9.4        5
...       ...      ...
1594     10.5        5
1595     11.2        6
1596     11.0        6
1597     10.2        5
1598     11.0        6

[1599 rows x 12 columns]
```

```
# checking whether the data is balanced or not
df.quality.value_counts()
```

```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

```
df.shape
```

```
(1599, 12)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
```

```
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

```
df.describe()
```

```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide
density  \
count  1599.000000          1599.000000           1599.000000
1599.000000
mean      0.087467            15.874922             46.467792
0.996747
std       0.047065            10.460157             32.895324
0.001887
min       0.012000             1.000000              6.000000
0.990070
25%       0.070000             7.000000             22.000000
0.995600
50%       0.079000            14.000000             38.000000
0.996750
75%       0.090000            21.000000             62.000000
0.997835
max       0.611000            72.000000            289.000000
1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
```

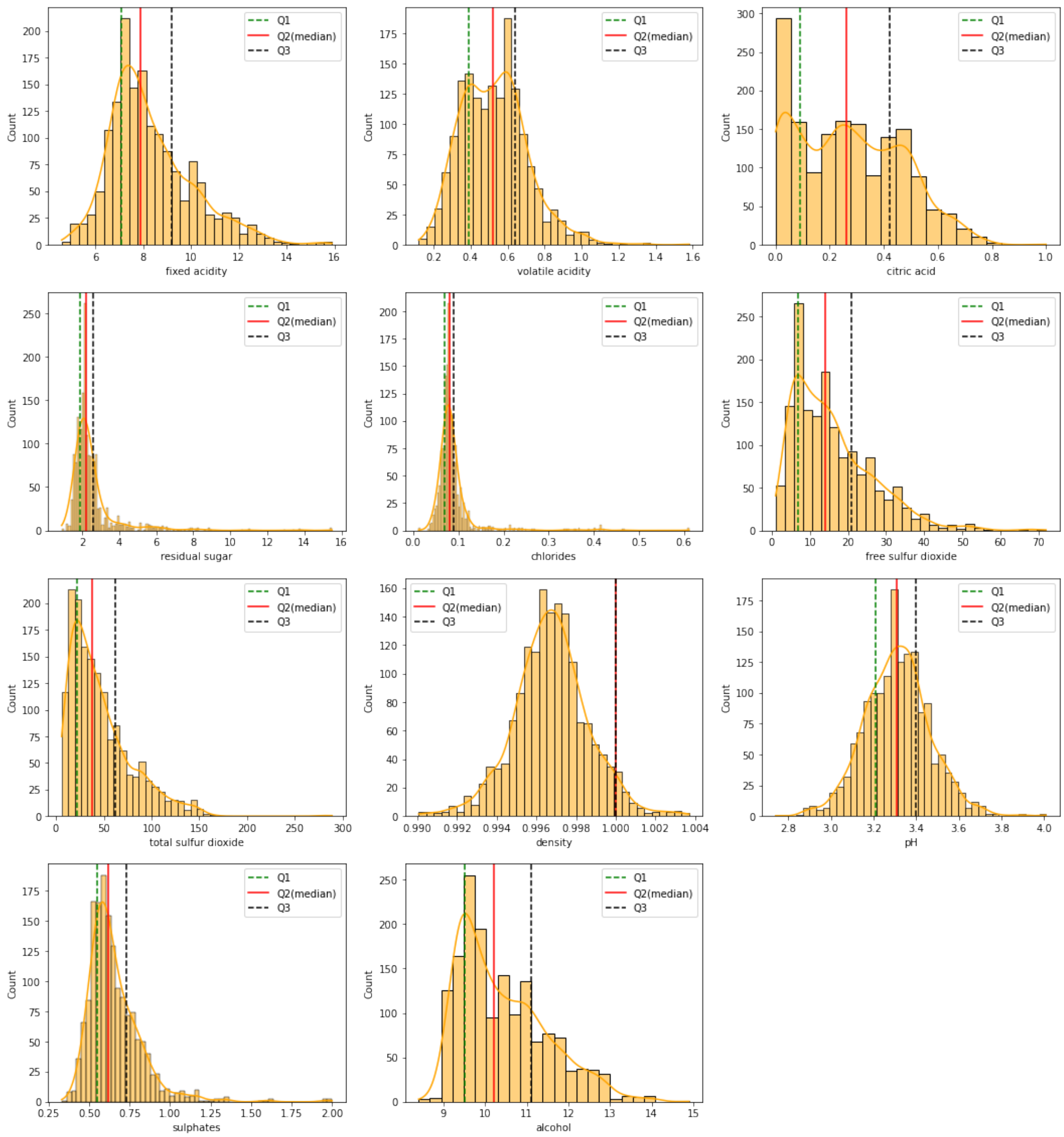
```
25%        3.210000     0.550000     9.500000     5.000000
50%        3.310000     0.620000    10.200000     6.000000
75%        3.400000     0.730000    11.100000     6.000000
max        4.010000     2.000000    14.900000     8.000000
```

```
# checking if there are any null values present in the data
df.isnull().sum()
```

```
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
```

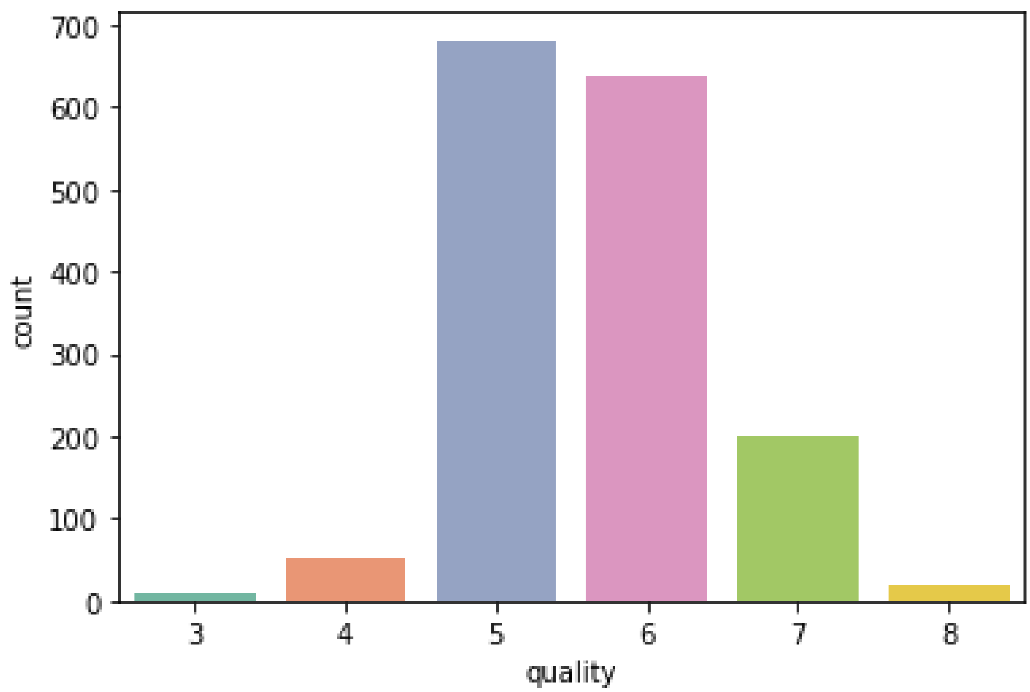
Exploratory Data Analysis

```
# let's check distribution of the data
features_ = df.columns[:-1]

def get_percentile(feature,q_range):
    dist = df[feature].describe()[str(q_range) + '%']
    return round(dist,2)
def render_counterplot():
    fig = plt.figure(figsize=(18,20))
    for column, feature in enumerate(features_):
        fig.add_subplot(4,3, column +1)
        q1 = get_percentile(feature,25)
        q2 = get_percentile(feature,50)
        q3 = get_percentile(feature,75)
        sns.histplot(data = df, x = feature, kde=True, color='orange')
        plt.axvline(q1, linestyle ='--', color='green',label='Q1')
        plt.axvline(q2,color='red',label='Q2(median)')
        plt.axvline(q3,linestyle='--',color='black',label='Q3')
        plt.legend()
render_counterplot()
```

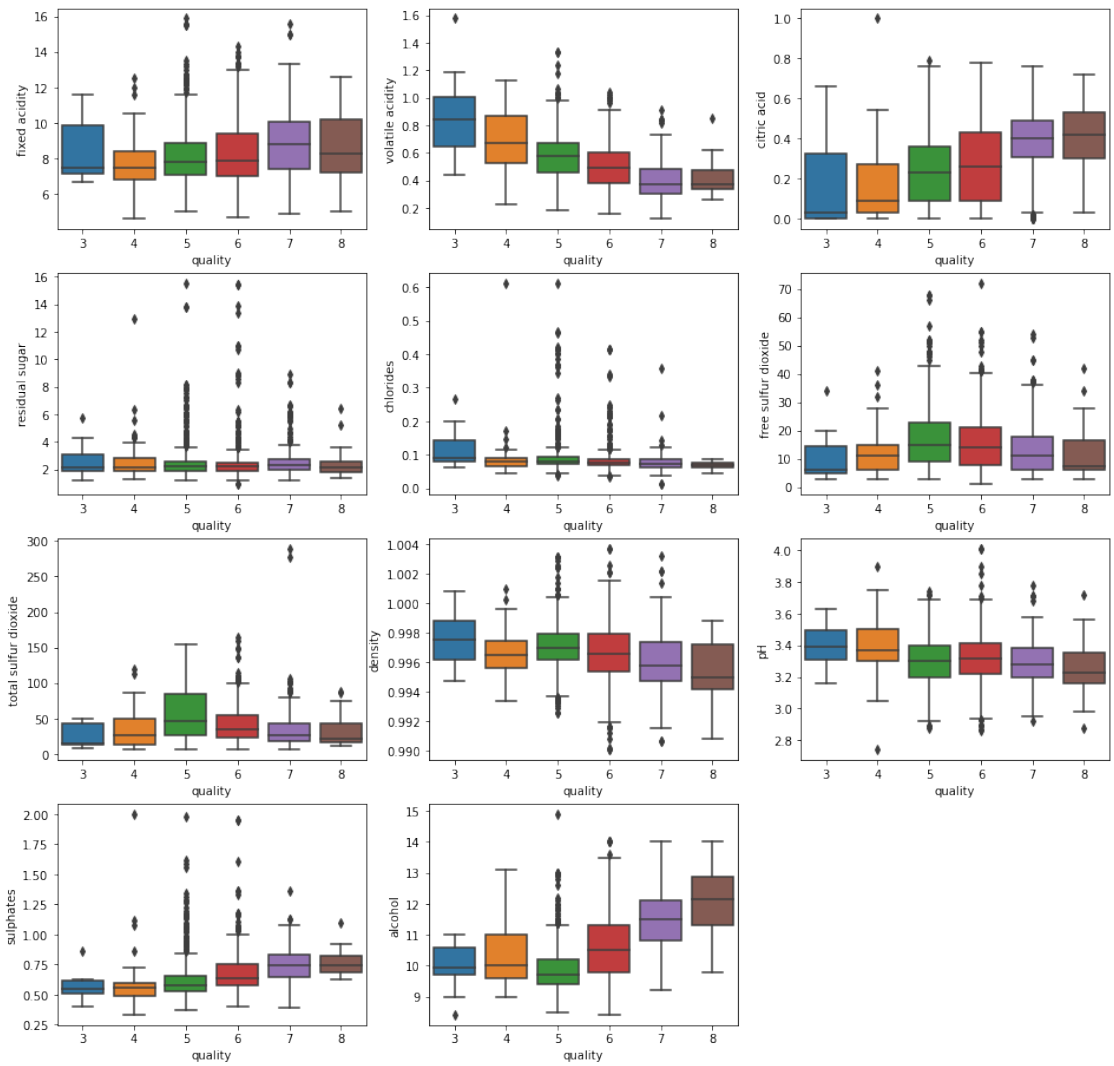



```
# We can clearly look at countplot and can tell that data is
imbalanced
sns.countplot(x='quality',data=df,palette='Set2')
```

<matplotlib.axes._subplots.AxesSubplot at 0x1b5d11a0fd0>




```
# Outlier check using a box plot
features_ = df.columns[:-1]
fig = plt.figure(figsize=(16,20))
for column, feature in enumerate(features_):
    fig.add_subplot(5,3, column+1)
    sns.boxplot(data=df, x='quality',y =feature)
```




```
# Split features and target
x = df.drop('quality',axis= 1)
y = df['quality']

x
```

```
      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  3.45
0.58
1595                 39.0                  51.0  0.99512  3.52
0.76
1596                 29.0                  40.0  0.99574  3.42
0.75
1597                 32.0                  44.0  0.99547  3.57
0.71
1598                 18.0                  42.0  0.99549  3.39
0.66

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

```
[1599 rows x 11 columns]
```

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
dtypes: float64(11)
memory usage: 137.5 KB
```

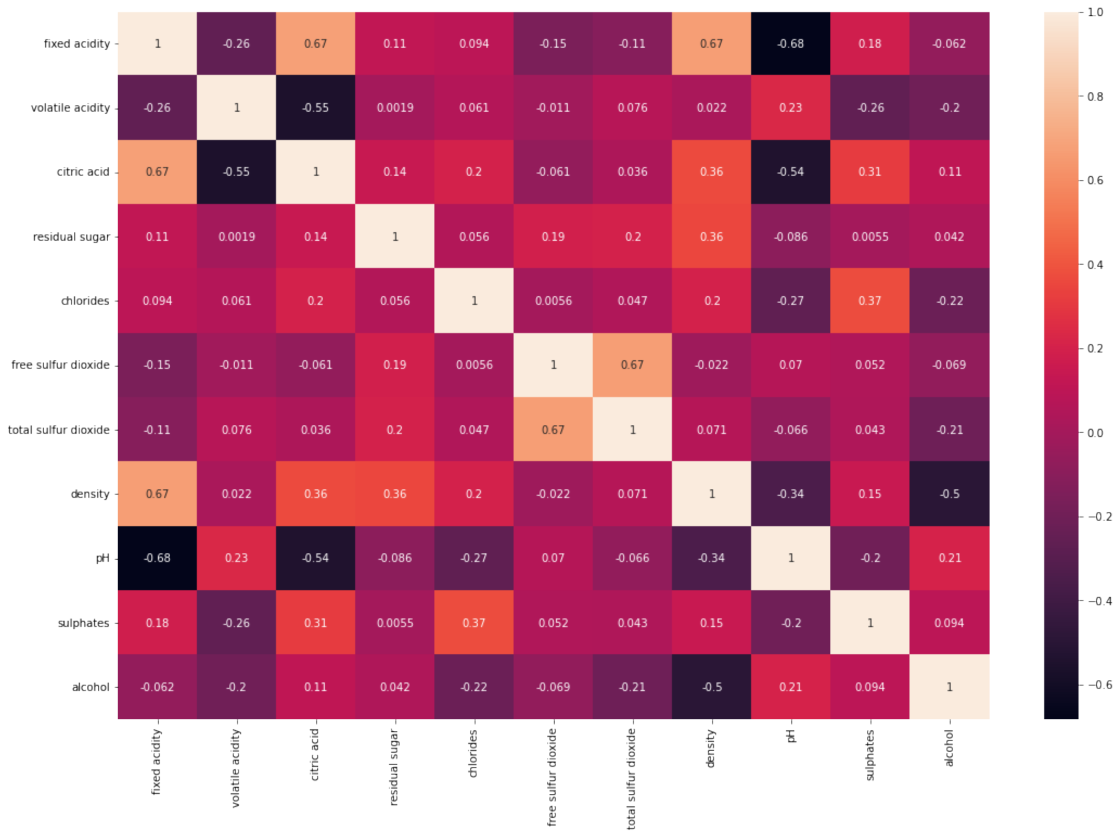
```
# Importing StandardScaler to perform scaling
from sklearn.preprocessing import StandardScaler

# Scaling down the data
scaler = StandardScaler()
x[x.columns] = scaler.fit_transform(x[x.columns])

x
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar 
chlorides  \
0         -0.528360          0.961877    -1.391472       -0.453218  -
0.243707   
1         -0.298547          1.967442    -1.391472        0.043416   
0.223875   
2         -0.298547          1.297065    -1.186070       -0.169427   
0.096353   
3          1.654856         -1.384443     1.484154       -0.453218  -
0.264960   
4         -0.528360          0.961877    -1.391472       -0.453218  -
0.243707   
...             ...               ...          ...             ...   
...   
1594      -1.217796          0.403229    -0.980669       -0.382271   
0.053845   
1595      -1.390155          0.123905    -0.877968       -0.240375  -
0.541259   
```

```
1596      -1.160343         -0.099554    -0.723916       -0.169427  -
0.243707
1597      -1.390155          0.654620    -0.775267       -0.382271  -
0.264960
1598      -1.332702         -1.216849     1.021999        0.752894  -
0.434990

      free sulfur dioxide  total sulfur dioxide   density        pH  \
0               -0.466193             -0.379133  0.558274  1.288643
1                0.872638              0.624363  0.028261 -0.719933
2               -0.083669              0.229047  0.134264 -0.331177
3                0.107592              0.411500  0.664277 -0.979104
4               -0.466193             -0.379133  0.558274  1.288643
...                   ...                   ...       ...       ...
1594             1.542054             -0.075043 -0.978765  0.899886
1595             2.211469              0.137820 -0.862162  1.353436
1596             1.255161             -0.196679 -0.533554  0.705508
1597             1.542054             -0.075043 -0.676657  1.677400
1598             0.203223             -0.135861 -0.666057  0.511130

      sulphates   alcohol
0     -0.579207 -0.960246
1      0.128950 -0.584777
2     -0.048089 -0.584777
3     -0.461180 -0.584777
4     -0.579207 -0.960246
...         ...       ...
1594  -0.461180  0.072294
1595   0.601055  0.729364
1596   0.542042  0.541630
1597   0.305990 -0.209308
1598   0.010924  0.541630

[1599 rows x 11 columns]
```

```
# Plotting heatmap to understand the correlation among the data
plt.figure(figsize=(18,12))
sns.heatmap(x.corr(), annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b5de4f1850>
```



Handling Imbalanced Dataset

```
# importing SMOTE from imblearn.over_sampling
from imblearn.over_sampling import SMOTE

# Balancing the data using SMOTE through oversampling
smote = SMOTE()
x_sm, y_sm = smote.fit_resample(x,y)

# We can see now the data is balanced
y_sm.value_counts()

7    681
5    681
3    681
8    681
6    681
4    681
Name: quality, dtype: int64

# splitting the data into training and testing data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x_sm,y_sm,test_size=0.2, random_state=42)

# importing required libraries for model building
import tensorflow
```

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.layers import Dropout

x_train
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  
chlorides  \
2185      -0.571384          3.417322    -1.380485        1.028101   
0.159421   
2462      -0.841831          0.577521    -1.289872       -0.485580  -
0.458564   
2819      -0.939248          2.857041    -0.996251       -0.177501  -
0.422091   
1769      -0.580848          2.625052    -1.156907       -0.179511  -
0.496395   
1373      -0.356000          1.241200    -0.005010        0.894790   
0.478920   
...             ...               ...          ...             ...   
...   
1130       0.448342          0.403229    -1.391472       -0.453218  -
0.626274   
1294      -0.068735          0.598756    -0.877968       -0.311323  -
0.307468   
860       -0.643266          0.514959    -1.083370        0.114364  -
0.222453   
3507       0.460028         -0.751943     1.196940       -0.476063  -
0.337006   
3174      -1.772759          0.943529    -1.207899       -0.753634  -
0.904831   

      free sulfur dioxide  total sulfur dioxide   density        pH  \
2185            -1.029747             -0.993818 -0.111762  2.010704   
2462            -1.108578             -1.161171 -1.123429  0.213942   
2819             0.291599             -0.449179 -0.183765  1.370640   
1769             0.270529             -0.014968  0.140154  0.171952   
1373             1.733315              1.293361 -0.056541 -0.460762   
...                   ...                   ...       ...       ...   
1130            -1.039977             -1.108948  0.505273 -0.849519   
1294             0.872638              0.411500 -0.194345 -0.136798   
860             -0.083669              1.171725  0.378070  1.288643   
3507            -0.855199             -0.924432 -1.063204 -0.681498   
3174            -0.222034              1.297254 -2.233439  1.641560   

      sulphates   alcohol  
2185  -0.640414  0.184714  
2462  -1.400213  0.111085  
2819  -0.524670  0.548751  
1769  -0.643319 -0.597531  
1373  -1.228350 -1.054113  
```

```
...          ...       ...
1130  -0.166115 -0.021574
1294   0.542042  0.447763
860   -0.697233 -0.866379
3507   0.279989  1.898725
3174   1.425954  2.367053

[3268 rows x 11 columns]
```

```
# creating a sequence for an ANN model
model = Sequential()
model.add(tensorflow.keras.layers.Input(shape=11,))
model.add(tensorflow.keras.layers.Dense(32,activation='relu'))
model.add(tensorflow.keras.layers.Dense(64,activation='relu'))
model.add(tensorflow.keras.layers.Dropout(0.3))
model.add(tensorflow.keras.layers.Dense(128,activation='relu'))
model.add(tensorflow.keras.layers.Dense(6,activation='softmax'))

# LabelEncoding because categorical_crossentropy take data in one-hot
encoded format
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y_train = le.fit_transform(y_train)
y_test = le.fit_transform(y_test)

# Converted into 0-5 but still not one-hot encoded
y_test
```

```
array([3, 2, 4, 4, 1, 1, 2, 3, 2, 3, 5, 4, 2, 0, 5, 3, 2, 2, 5, 1, 3,
0,
       2, 3, 1, 0, 3, 0, 2, 5, 2, 0, 5, 3, 0, 3, 2, 5, 2, 4, 5, 5, 3,
4,
       4, 2, 4, 2, 1, 5, 3, 3, 2, 3, 3, 0, 1, 4, 1, 3, 3, 1, 3, 0, 5,
1,
       0, 4, 4, 3, 1, 0, 2, 4, 1, 2, 1, 1, 4, 0, 5, 3, 2, 3, 3, 5, 5,
2,
       5, 1, 0, 3, 3, 2, 3, 3, 3, 4, 5, 5, 1, 5, 2, 2, 3, 1, 3, 0, 4,
4,
       3, 1, 4, 3, 2, 3, 0, 5, 1, 2, 2, 2, 3, 2, 3, 4, 2, 4, 2, 0, 4,
4,
       2, 0, 1, 0, 5, 2, 3, 0, 4, 2, 3, 3, 4, 5, 4, 3, 5, 1, 1, 2, 1,
2,
       1, 1, 0, 5, 4, 5, 1, 0, 5, 3, 4, 0, 5, 3, 1, 1, 1, 5, 1, 1, 5,
3,
       3, 0, 3, 1, 1, 2, 5, 5, 4, 4, 2, 4, 2, 4, 2, 3, 2, 4, 3, 1, 1,
0,
       1, 5, 4, 5, 0, 1, 0, 4, 1, 2, 1, 3, 5, 4, 2, 3, 5, 4, 1, 2, 4,
0,
       2, 0, 5, 5, 4, 0, 2, 0, 3, 5, 4, 1, 5, 1, 2, 3, 5, 4, 1, 1, 4,
5,
       0, 1, 3, 3, 1, 2, 3, 4, 1, 3, 3, 5, 1, 4, 2, 1, 2, 2, 1, 4, 0,
```

3,
3, 0, 4, 4, 5, 0, 4, 1, 3, 3, 0, 3, 3, 3, 0, 5, 2, 5, 2, 3, 2,
2,
0, 1, 1, 5, 5, 0, 1, 2, 4, 0, 0, 2, 4, 1, 3, 5, 1, 4, 1, 1, 3,
0,
3, 2, 5, 3, 4, 1, 0, 3, 2, 1, 1, 4, 0, 4, 3, 1, 5, 3, 1, 5, 3,
5,
2, 0, 5, 3, 3, 5, 3, 3, 2, 0, 0, 3, 0, 5, 5, 3, 5, 2, 4, 5, 5,
0,
2, 0, 1, 5, 5, 1, 2, 2, 5, 2, 4, 0, 3, 1, 4, 5, 3, 0, 1, 2, 5,
3,
4, 1, 2, 0, 2, 5, 4, 4, 0, 0, 2, 3, 1, 1, 2, 0, 0, 3, 0, 5, 5,
0,
2, 5, 3, 1, 0, 3, 2, 3, 2, 3, 4, 2, 1, 3, 2, 0, 3, 5, 2, 1, 0,
1,
2, 0, 0, 5, 0, 4, 2, 2, 0, 2, 4, 3, 3, 2, 4, 4, 4, 3, 4, 5, 1,
0,
1, 4, 4, 1, 5, 0, 3, 2, 4, 4, 1, 5, 5, 0, 2, 5, 5, 3, 3, 3, 5,
1,
4, 0, 0, 1, 2, 5, 0, 0, 3, 3, 0, 0, 2, 0, 1, 5, 0, 3, 3, 1, 5,
5,
2, 3, 0, 4, 3, 1, 1, 1, 2, 2, 1, 3, 3, 1, 4, 0, 4, 2, 1, 2, 0,
3,
3, 5, 2, 4, 0, 2, 5, 5, 5, 5, 5, 5, 4, 1, 4, 1, 4, 4, 1, 4, 5,
2,
2, 0, 0, 1, 2, 5, 1, 2, 1, 2, 5, 0, 4, 3, 0, 1, 5, 2, 2, 2, 2,
3,
2, 0, 3, 5, 2, 3, 4, 0, 2, 4, 2, 1, 1, 0, 5, 3, 3, 5, 4, 0, 4,
5,
4, 5, 3, 2, 4, 2, 3, 5, 3, 4, 5, 0, 1, 2, 0, 0, 4, 0, 4, 2, 2,
3,
0, 5, 2, 2, 2, 2, 4, 5, 1, 0, 0, 2, 5, 5, 4, 2, 3, 1, 5, 3, 3,
5,
5, 0, 4, 4, 1, 3, 5, 0, 1, 1, 2, 1, 2, 1, 5, 5, 1, 2, 0, 3, 2,
2,
2, 4, 3, 4, 2, 1, 1, 5, 0, 2, 4, 3, 3, 3, 3, 3, 3, 0, 5, 2, 0,
3,
0, 0, 2, 1, 3, 3, 4, 4, 5, 4, 4, 5, 3, 4, 2, 3, 0, 1, 3, 0, 0,
3,
2, 0, 3, 1, 0, 3, 0, 1, 5, 1, 5, 1, 4, 3, 2, 1, 5, 2, 3, 4, 3,
1,
1, 0, 3, 4, 5, 1, 5, 1, 4, 0, 0, 0, 2, 0, 5, 2, 0, 4, 4, 1, 2,
1,
2, 0, 1, 4, 0, 5, 0, 5, 3, 0, 4, 3, 2, 3, 2, 0, 1, 0, 3, 0, 0,
4,
1, 1, 2, 5, 4, 3, 0, 0, 0, 2, 2, 5, 5, 1, 3, 4, 4, 4, 2, 2, 0,
5,
3, 1, 4, 3, 5, 4, 1, 0, 1, 5, 3, 3, 0, 3, 5, 3, 3, 5, 3, 0, 3,
2,
2, 3, 2, 0, 0, 5, 5, 2, 5, 5, 2, 2, 4, 5, 3, 1, 2, 2, 5, 5, 1,

```
2,
       2, 0, 1, 2], dtype=int64)
```

```
# Flattening the array to feed the data to input layers
y_train = pd.DataFrame(y_train.reshape(len(y_train),1))
y_test = pd.DataFrame(y_test.reshape(len(y_test),1))
```

```
y_train
```

```
      0
0     0
1     1
2     1
3     0
4     2
...  ..
3263  3
3264  3
3265  2
3266  5
3267  4

[3268 rows x 1 columns]
```

```
# Now converting the flattened array into one-hot encoded format
y_train = tensorflow.keras.utils.to_categorical(y_train,6)
y_test = tensorflow.keras.utils.to_categorical(y_test,6)
```

```
# Building the model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics
=['accuracy'])
```

```
# Training the model
model.fit(x_train,y_train,epochs=400,verbose=1)
```

```
Epoch 1/400
103/103 [==============================] - 1s 2ms/step - loss: 1.4219
- accuracy: 0.4058
Epoch 2/400
103/103 [==============================] - 0s 2ms/step - loss: 1.0639
- accuracy: 0.5370
Epoch 3/400
103/103 [==============================] - 0s 2ms/step - loss: 0.9492
- accuracy: 0.6019
Epoch 4/400
103/103 [==============================] - 0s 2ms/step - loss: 0.8820
- accuracy: 0.6343
Epoch 5/400
103/103 [==============================] - 0s 2ms/step - loss: 0.8318
- accuracy: 0.6717
Epoch 6/400
103/103 [==============================] - 0s 2ms/step - loss: 0.7854
```

```
- accuracy: 0.6818
Epoch 7/400
103/103 [==============================] - 0s 2ms/step - loss: 0.7468
- accuracy: 0.7032
Epoch 8/400
103/103 [==============================] - 0s 2ms/step - loss: 0.7194
- accuracy: 0.7111
Epoch 9/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6900
- accuracy: 0.7258
Epoch 10/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6520
- accuracy: 0.7405
Epoch 11/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6434
- accuracy: 0.7460
Epoch 12/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6321
- accuracy: 0.7448
Epoch 13/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6196
- accuracy: 0.7555
Epoch 14/400
103/103 [==============================] - 0s 2ms/step - loss: 0.6019
- accuracy: 0.7650
Epoch 15/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5881
- accuracy: 0.7653
Epoch 16/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5782
- accuracy: 0.7739
Epoch 17/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5611
- accuracy: 0.7778
Epoch 18/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5447
- accuracy: 0.7861
Epoch 19/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5396
- accuracy: 0.7812
Epoch 20/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5290
- accuracy: 0.7855
Epoch 21/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5214
- accuracy: 0.7870
Epoch 22/400
103/103 [==============================] - 0s 2ms/step - loss: 0.5066
- accuracy: 0.7980
Epoch 23/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.4975
- accuracy: 0.7935
Epoch 24/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4945
- accuracy: 0.8011
Epoch 25/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4873
- accuracy: 0.8057
Epoch 26/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4865
- accuracy: 0.8039
Epoch 27/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4633
- accuracy: 0.8109
Epoch 28/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4723
- accuracy: 0.8048
Epoch 29/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4521
- accuracy: 0.8277
Epoch 30/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4535
- accuracy: 0.8146
Epoch 31/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4474
- accuracy: 0.8179
Epoch 32/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4421
- accuracy: 0.8250
Epoch 33/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4410
- accuracy: 0.8265
Epoch 34/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4330
- accuracy: 0.8259
Epoch 35/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4274
- accuracy: 0.8305
Epoch 36/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4199
- accuracy: 0.8268
Epoch 37/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4114
- accuracy: 0.8409
Epoch 38/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4114
- accuracy: 0.8320
Epoch 39/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4031
- accuracy: 0.8363
```

```
Epoch 40/400
103/103 [==============================] - 0s 2ms/step - loss: 0.4040
- accuracy: 0.8360
Epoch 41/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3967
- accuracy: 0.8406
Epoch 42/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3938
- accuracy: 0.8375
Epoch 43/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3940
- accuracy: 0.8470
Epoch 44/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3807
- accuracy: 0.8452
Epoch 45/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3941
- accuracy: 0.8461
Epoch 46/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3773
- accuracy: 0.8436
Epoch 47/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3668
- accuracy: 0.8528
Epoch 48/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3637
- accuracy: 0.8568
Epoch 49/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3704
- accuracy: 0.8494
Epoch 50/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3623
- accuracy: 0.8513
Epoch 51/400
103/103 [==============================] - 0s 1ms/step - loss: 0.3553
- accuracy: 0.8556
Epoch 52/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3625
- accuracy: 0.8586
Epoch 53/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3587
- accuracy: 0.8519
Epoch 54/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3533
- accuracy: 0.8550
Epoch 55/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3497
- accuracy: 0.8617
Epoch 56/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3440
```

```
- accuracy: 0.8614
Epoch 57/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3458
- accuracy: 0.8647
Epoch 58/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3456
- accuracy: 0.8684
Epoch 59/400
103/103 [==============================] - 0s 1ms/step - loss: 0.3292
- accuracy: 0.8672
Epoch 60/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3374
- accuracy: 0.8635
Epoch 61/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3217
- accuracy: 0.8675
Epoch 62/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3184
- accuracy: 0.8785
Epoch 63/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3310
- accuracy: 0.8660
Epoch 64/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3250
- accuracy: 0.8693
Epoch 65/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3188
- accuracy: 0.8730
Epoch 66/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3153
- accuracy: 0.8813
Epoch 67/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3092
- accuracy: 0.8748
Epoch 68/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3170
- accuracy: 0.8758
Epoch 69/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3043
- accuracy: 0.8837
Epoch 70/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3090
- accuracy: 0.8767
Epoch 71/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3129
- accuracy: 0.8752
Epoch 72/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3209
- accuracy: 0.8730
Epoch 73/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.3050
- accuracy: 0.8782
Epoch 74/400
103/103 [==============================] - 0s 2ms/step - loss: 0.3018
- accuracy: 0.8794
Epoch 75/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2910
- accuracy: 0.8837
Epoch 76/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2968
- accuracy: 0.8739
Epoch 77/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2911
- accuracy: 0.8846
Epoch 78/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2954
- accuracy: 0.8785
Epoch 79/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2961
- accuracy: 0.8819
Epoch 80/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2969
- accuracy: 0.8807
Epoch 81/400

103/103 [==============================] - 0s 2ms/step - loss: 0.2954
- accuracy: 0.8810
Epoch 82/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2860
- accuracy: 0.8810
Epoch 83/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2778
- accuracy: 0.8905
Epoch 84/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2797
- accuracy: 0.8874
Epoch 85/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2829
- accuracy: 0.8880
Epoch 86/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2772
- accuracy: 0.8862
Epoch 87/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2709
- accuracy: 0.8953
Epoch 88/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2723
- accuracy: 0.8944
Epoch 89/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2716
```

```
- accuracy: 0.8923
Epoch 90/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2707
- accuracy: 0.8941
Epoch 91/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2868
- accuracy: 0.8831
Epoch 92/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2625
- accuracy: 0.8969
Epoch 93/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2656
- accuracy: 0.8944
Epoch 94/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2645
- accuracy: 0.8935
Epoch 95/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2604
- accuracy: 0.8950
Epoch 96/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2582
- accuracy: 0.8990
Epoch 97/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2547
- accuracy: 0.8938
Epoch 98/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2567
- accuracy: 0.8957
Epoch 99/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2509
- accuracy: 0.8966
Epoch 100/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2647
- accuracy: 0.8941
Epoch 101/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2502
- accuracy: 0.9018
Epoch 102/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2492
- accuracy: 0.8999
Epoch 103/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2481
- accuracy: 0.9073
Epoch 104/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2464
- accuracy: 0.9064
Epoch 105/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2457
- accuracy: 0.8984
Epoch 106/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.2445
- accuracy: 0.9039
Epoch 107/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2362
- accuracy: 0.9067
Epoch 108/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2429
- accuracy: 0.9085
Epoch 109/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2496
- accuracy: 0.9021
Epoch 110/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2477
- accuracy: 0.9033
Epoch 111/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2315
- accuracy: 0.9125
Epoch 112/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2384
- accuracy: 0.9073
Epoch 113/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2428
- accuracy: 0.9079
Epoch 114/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2384
- accuracy: 0.9064
Epoch 115/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2307
- accuracy: 0.9036
Epoch 116/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2302
- accuracy: 0.9079
Epoch 117/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2423
- accuracy: 0.9021
Epoch 118/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2434
- accuracy: 0.8984
Epoch 119/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2393
- accuracy: 0.9082
Epoch 120/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2369
- accuracy: 0.9116
Epoch 121/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2271
- accuracy: 0.9058
Epoch 122/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2310
- accuracy: 0.9067
```

```
Epoch 123/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2332
- accuracy: 0.9051
Epoch 124/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2194
- accuracy: 0.9122
Epoch 125/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2435
- accuracy: 0.9048
Epoch 126/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2228
- accuracy: 0.9119
Epoch 127/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2373
- accuracy: 0.9039
Epoch 128/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2130
- accuracy: 0.9171
Epoch 129/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2212
- accuracy: 0.9177
Epoch 130/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2206
- accuracy: 0.9128
Epoch 131/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2272
- accuracy: 0.9091
Epoch 132/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2218
- accuracy: 0.9106
Epoch 133/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2196
- accuracy: 0.9137
Epoch 134/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2159
- accuracy: 0.9137
Epoch 135/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2126
- accuracy: 0.9152
Epoch 136/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2119
- accuracy: 0.9189
Epoch 137/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2211
- accuracy: 0.9106
Epoch 138/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2171
- accuracy: 0.9134
Epoch 139/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2098
```

```
- accuracy: 0.9183
Epoch 140/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2142
- accuracy: 0.9159
Epoch 141/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2099
- accuracy: 0.9192
Epoch 142/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2144
- accuracy: 0.9103
Epoch 143/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2121
- accuracy: 0.9189
Epoch 144/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2206
- accuracy: 0.9119
Epoch 145/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2086
- accuracy: 0.9195
Epoch 146/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2088
- accuracy: 0.9186
Epoch 147/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1903
- accuracy: 0.9238
Epoch 148/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2105
- accuracy: 0.9183
Epoch 149/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2108
- accuracy: 0.9165
Epoch 150/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2149
- accuracy: 0.9165
Epoch 151/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2117
- accuracy: 0.9174
Epoch 152/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2053
- accuracy: 0.9198
Epoch 153/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1935
- accuracy: 0.9217
Epoch 154/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1890
- accuracy: 0.9287
Epoch 155/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1963
- accuracy: 0.9244
Epoch 156/400
```

```
103/103 [==============================] - 0s 1ms/step - loss: 0.1906
- accuracy: 0.9250
Epoch 157/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1964
- accuracy: 0.9238
Epoch 158/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2054
- accuracy: 0.9162
Epoch 159/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1947
- accuracy: 0.9266
Epoch 160/400

103/103 [==============================] - 0s 2ms/step - loss: 0.2019
- accuracy: 0.9229
Epoch 161/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1896
- accuracy: 0.9229
Epoch 162/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1930
- accuracy: 0.9263
Epoch 163/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1977
- accuracy: 0.9226
Epoch 164/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1791
- accuracy: 0.9296
Epoch 165/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1900
- accuracy: 0.9281
Epoch 166/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1858
- accuracy: 0.9281
Epoch 167/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1922
- accuracy: 0.9220
Epoch 168/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1932
- accuracy: 0.9229
Epoch 169/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1857
- accuracy: 0.9263
Epoch 170/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1779
- accuracy: 0.9302
Epoch 171/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1807
- accuracy: 0.9284
Epoch 172/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1889
```

```
- accuracy: 0.9269
Epoch 173/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1818
- accuracy: 0.9321
Epoch 174/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1848
- accuracy: 0.9259
Epoch 175/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1745
- accuracy: 0.9360
Epoch 176/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2024
- accuracy: 0.9223
Epoch 177/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1858
- accuracy: 0.9256
Epoch 178/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1848
- accuracy: 0.9324
Epoch 179/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1859
- accuracy: 0.9275
Epoch 180/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1818
- accuracy: 0.9339
Epoch 181/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1857
- accuracy: 0.9272
Epoch 182/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2015
- accuracy: 0.9201
Epoch 183/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2093
- accuracy: 0.9171
Epoch 184/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1800
- accuracy: 0.9318
Epoch 185/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1651
- accuracy: 0.9367
Epoch 186/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1884
- accuracy: 0.9253
Epoch 187/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1895
- accuracy: 0.9183
Epoch 188/400
103/103 [==============================] - 0s 2ms/step - loss: 0.2022
- accuracy: 0.9238
Epoch 189/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.1763
- accuracy: 0.9318
Epoch 190/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1853
- accuracy: 0.9305
Epoch 191/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1874
- accuracy: 0.9235
Epoch 192/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1837
- accuracy: 0.9296
Epoch 193/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1784
- accuracy: 0.9278
Epoch 194/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1768
- accuracy: 0.9339
Epoch 195/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1778
- accuracy: 0.9247
Epoch 196/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1873
- accuracy: 0.9259
Epoch 197/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1625
- accuracy: 0.9400
Epoch 198/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1725
- accuracy: 0.9287
Epoch 199/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1641
- accuracy: 0.9370
Epoch 200/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1649
- accuracy: 0.9391
Epoch 201/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1611
- accuracy: 0.9397
Epoch 202/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1624
- accuracy: 0.9376
Epoch 203/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1662
- accuracy: 0.9339
Epoch 204/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1564
- accuracy: 0.9400
Epoch 205/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1799
- accuracy: 0.9327
```

```
Epoch 206/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1712
- accuracy: 0.9324
Epoch 207/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1631
- accuracy: 0.9333
Epoch 208/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1620
- accuracy: 0.9370
Epoch 209/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1830
- accuracy: 0.9290
Epoch 210/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1609
- accuracy: 0.9370
Epoch 211/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1657
- accuracy: 0.9348
Epoch 212/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1590
- accuracy: 0.9376
Epoch 213/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1740
- accuracy: 0.9385
Epoch 214/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1640
- accuracy: 0.9367
Epoch 215/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1691
- accuracy: 0.9299
Epoch 216/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1631
- accuracy: 0.9373
Epoch 217/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1535
- accuracy: 0.9391
Epoch 218/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1603
- accuracy: 0.9394
Epoch 219/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1810
- accuracy: 0.9299
Epoch 220/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1648
- accuracy: 0.9360
Epoch 221/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1539
- accuracy: 0.9406
Epoch 222/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1584
```

```
- accuracy: 0.9367
Epoch 223/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1466
- accuracy: 0.9452
Epoch 224/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1554
- accuracy: 0.9403
Epoch 225/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1712
- accuracy: 0.9367
Epoch 226/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1542
- accuracy: 0.9443
Epoch 227/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1504
- accuracy: 0.9382
Epoch 228/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1545
- accuracy: 0.9406
Epoch 229/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1617
- accuracy: 0.9364
Epoch 230/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1519
- accuracy: 0.9385
Epoch 231/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1557
- accuracy: 0.9391
Epoch 232/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1654
- accuracy: 0.9385
Epoch 233/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1659
- accuracy: 0.9373
Epoch 234/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1757
- accuracy: 0.9336
Epoch 235/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1601
- accuracy: 0.9373
Epoch 236/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1625
- accuracy: 0.9403
Epoch 237/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1576
- accuracy: 0.9446
Epoch 238/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1464
- accuracy: 0.9443
Epoch 239/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.1474
- accuracy: 0.9480
Epoch 240/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1578
- accuracy: 0.9412
Epoch 241/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1496
- accuracy: 0.9425
Epoch 242/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1536
- accuracy: 0.9367
Epoch 243/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1451
- accuracy: 0.9412
Epoch 244/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1532
- accuracy: 0.9483
Epoch 245/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1548
- accuracy: 0.9406
Epoch 246/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1466
- accuracy: 0.9409
Epoch 247/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1646
- accuracy: 0.9367
Epoch 248/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1432
- accuracy: 0.9416
Epoch 249/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1509
- accuracy: 0.9409
Epoch 250/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1518
- accuracy: 0.9431
Epoch 251/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1660
- accuracy: 0.9428
Epoch 252/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1562
- accuracy: 0.9345
Epoch 253/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1568
- accuracy: 0.9388
Epoch 254/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1423
- accuracy: 0.9486
Epoch 255/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1399
- accuracy: 0.9452
```

```
Epoch 256/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1459
- accuracy: 0.9458
Epoch 257/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1469
- accuracy: 0.9428
Epoch 258/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1385
- accuracy: 0.9492
Epoch 259/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1504
- accuracy: 0.9443
Epoch 260/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1425
- accuracy: 0.9465
Epoch 261/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1432
- accuracy: 0.9446
Epoch 262/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1462
- accuracy: 0.9437
Epoch 263/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1513
- accuracy: 0.9400
Epoch 264/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1517
- accuracy: 0.9428
Epoch 265/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1391
- accuracy: 0.9443
Epoch 266/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1446
- accuracy: 0.9465
Epoch 267/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1437
- accuracy: 0.9498
Epoch 268/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1405
- accuracy: 0.9489
Epoch 269/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1394
- accuracy: 0.9495
Epoch 270/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1570
- accuracy: 0.9406
Epoch 271/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1438
- accuracy: 0.9443
Epoch 272/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1550
```

```
- accuracy: 0.9419
Epoch 273/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1428
- accuracy: 0.9489
Epoch 274/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1509
- accuracy: 0.9431
Epoch 275/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1491
- accuracy: 0.9425
Epoch 276/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1328
- accuracy: 0.9474
Epoch 277/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1382
- accuracy: 0.9510
Epoch 278/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1372
- accuracy: 0.9480
Epoch 279/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1383
- accuracy: 0.9498
Epoch 280/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1446
- accuracy: 0.9486
Epoch 281/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1455
- accuracy: 0.9455
Epoch 282/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1365
- accuracy: 0.9465
Epoch 283/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1310
- accuracy: 0.9517
Epoch 284/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1426
- accuracy: 0.9471
Epoch 285/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1400
- accuracy: 0.9458
Epoch 286/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1275
- accuracy: 0.9526
Epoch 287/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1329
- accuracy: 0.9507
Epoch 288/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1440
- accuracy: 0.9465
Epoch 289/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.1397
- accuracy: 0.9477
Epoch 290/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1307
- accuracy: 0.9517
Epoch 291/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1298
- accuracy: 0.9547
Epoch 292/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1393
- accuracy: 0.9504
Epoch 293/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1474
- accuracy: 0.9471
Epoch 294/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1335
- accuracy: 0.9465
Epoch 295/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1198
- accuracy: 0.9541
Epoch 296/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1283
- accuracy: 0.9495
Epoch 297/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1314
- accuracy: 0.9532
Epoch 298/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1344
- accuracy: 0.9513
Epoch 299/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1441
- accuracy: 0.9480
Epoch 300/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1303
- accuracy: 0.9538
Epoch 301/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1355
- accuracy: 0.9455
Epoch 302/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1344
- accuracy: 0.9538
Epoch 303/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1353
- accuracy: 0.9468
Epoch 304/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1315
- accuracy: 0.9520
Epoch 305/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1569
- accuracy: 0.9422
```

```
Epoch 306/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1304
- accuracy: 0.9520
Epoch 307/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1235
- accuracy: 0.9547
Epoch 308/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1395
- accuracy: 0.9458
Epoch 309/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1267
- accuracy: 0.9513
Epoch 310/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1308
- accuracy: 0.9492
Epoch 311/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1310
- accuracy: 0.9504
Epoch 312/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1329
- accuracy: 0.9465
Epoch 313/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1287
- accuracy: 0.9541
Epoch 314/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1223
- accuracy: 0.9541
Epoch 315/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1221
- accuracy: 0.9547
Epoch 316/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1270
- accuracy: 0.9569
Epoch 317/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1336
- accuracy: 0.9492
Epoch 318/400

103/103 [==============================] - 0s 1ms/step - loss: 0.1276
- accuracy: 0.9559
Epoch 319/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1351
- accuracy: 0.9513
Epoch 320/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1304
- accuracy: 0.9520
Epoch 321/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1420
- accuracy: 0.9455
Epoch 322/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.1345
- accuracy: 0.9510
Epoch 323/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1474
- accuracy: 0.9452
Epoch 324/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1362
- accuracy: 0.9486
Epoch 325/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1481
- accuracy: 0.9465
Epoch 326/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1308
- accuracy: 0.9489
Epoch 327/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1294
- accuracy: 0.9529
Epoch 328/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1357
- accuracy: 0.9513
Epoch 329/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1343
- accuracy: 0.9474
Epoch 330/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1256
- accuracy: 0.9547
Epoch 331/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1289
- accuracy: 0.9535
Epoch 332/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1155
- accuracy: 0.9562
Epoch 333/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1270
- accuracy: 0.9532
Epoch 334/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1326
- accuracy: 0.9520
Epoch 335/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1160
- accuracy: 0.9587
Epoch 336/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1218
- accuracy: 0.9535
Epoch 337/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1287
- accuracy: 0.9526
Epoch 338/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1221
- accuracy: 0.9569
```

```
Epoch 339/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1306
- accuracy: 0.9480
Epoch 340/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1295
- accuracy: 0.9507
Epoch 341/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1179
- accuracy: 0.9578
Epoch 342/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1636
- accuracy: 0.9434
Epoch 343/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1374
- accuracy: 0.9532
Epoch 344/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1055
- accuracy: 0.9648
Epoch 345/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1232
- accuracy: 0.9538
Epoch 346/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1151
- accuracy: 0.9584
Epoch 347/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1113
- accuracy: 0.9611
Epoch 348/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1381
- accuracy: 0.9461
Epoch 349/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1401
- accuracy: 0.9486
Epoch 350/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1209
- accuracy: 0.9562
Epoch 351/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1284
- accuracy: 0.9535
Epoch 352/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1040
- accuracy: 0.9639
Epoch 353/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1392
- accuracy: 0.9498
Epoch 354/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1078
- accuracy: 0.9605
Epoch 355/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1314
```

```
- accuracy: 0.9538
Epoch 356/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1180
- accuracy: 0.9559
Epoch 357/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1188
- accuracy: 0.9541
Epoch 358/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1093
- accuracy: 0.9584
Epoch 359/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1070
- accuracy: 0.9575
Epoch 360/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1212
- accuracy: 0.9550
Epoch 361/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1287
- accuracy: 0.9517
Epoch 362/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1268
- accuracy: 0.9532
Epoch 363/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1220
- accuracy: 0.9501
Epoch 364/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1268
- accuracy: 0.9517
Epoch 365/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1144
- accuracy: 0.9572
Epoch 366/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1229
- accuracy: 0.9538
Epoch 367/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1241
- accuracy: 0.9504
Epoch 368/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1100
- accuracy: 0.9581
Epoch 369/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1208
- accuracy: 0.9569
Epoch 370/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1213
- accuracy: 0.9553
Epoch 371/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1127
- accuracy: 0.9575
Epoch 372/400
```

```
103/103 [==============================] - 0s 2ms/step - loss: 0.1179
- accuracy: 0.9593
Epoch 373/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1128
- accuracy: 0.9575
Epoch 374/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1054
- accuracy: 0.9605
Epoch 375/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1203
- accuracy: 0.9569
Epoch 376/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1093
- accuracy: 0.9578
Epoch 377/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1188
- accuracy: 0.9556
Epoch 378/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1090
- accuracy: 0.9621
Epoch 379/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1251
- accuracy: 0.9556
Epoch 380/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1292
- accuracy: 0.9547
Epoch 381/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1251
- accuracy: 0.9584
Epoch 382/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1059
- accuracy: 0.9578
Epoch 383/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1164
- accuracy: 0.9556
Epoch 384/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1094
- accuracy: 0.9605
Epoch 385/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1116
- accuracy: 0.9581
Epoch 386/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1177
- accuracy: 0.9569
Epoch 387/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1126
- accuracy: 0.9602
Epoch 388/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1077
- accuracy: 0.9584
```

```
Epoch 389/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1199
- accuracy: 0.9553
Epoch 390/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1179
- accuracy: 0.9581
Epoch 391/400
103/103 [==============================] - 0s 1ms/step - loss: 0.1140
- accuracy: 0.9535
Epoch 392/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1039
- accuracy: 0.9611
Epoch 393/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1110
- accuracy: 0.9581
Epoch 394/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1209
- accuracy: 0.9547
Epoch 395/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1199
- accuracy: 0.9605
Epoch 396/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1153
- accuracy: 0.9541
Epoch 397/400

103/103 [==============================] - 0s 2ms/step - loss: 0.1067
- accuracy: 0.9621
Epoch 398/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1214
- accuracy: 0.9569
Epoch 399/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1037
- accuracy: 0.9624
Epoch 400/400
103/103 [==============================] - 0s 2ms/step - loss: 0.1167
- accuracy: 0.9584

<keras.callbacks.History at 0x1b5e8656910>
```

```
y_train
```

```
array([[1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       ...,
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1.],
       [0., 0., 0., 0., 1., 0.]], dtype=float32)
```

```python
# Predicting by rounding the value returned by softmax because metrics accepts values less than 1
y_pred = model.predict(x_test).round()
```

```
26/26 [==============================] - 0s 2ms/step
```

```python
y_pred
```

```
array([[0., 0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0.],
       ...,
       [1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.]], dtype=float32)
```

```python
# Finding out maximum value to get the original data
y_pred = np.argmax(y_pred,axis=1)
y_pred
```

```
array([2, 2, 4, 4, 1, 1, 3, 2, 2, 1, 5, 4, 2, 0, 5, 2, 0, 3, 5, 1, 3,
0,
       2, 2, 1, 0, 3, 0, 3, 5, 2, 0, 5, 3, 0, 3, 2, 5, 2, 4, 5, 5, 3,
4,
       4, 1, 4, 1, 1, 5, 3, 3, 2, 3, 3, 0, 1, 4, 1, 3, 2, 1, 3, 0, 5,
1,
       0, 4, 4, 3, 1, 0, 2, 4, 1, 2, 1, 1, 4, 0, 5, 3, 3, 5, 2, 5, 5,
2,
       5, 1, 0, 3, 0, 2, 1, 3, 3, 4, 5, 5, 1, 5, 2, 2, 3, 1, 2, 0, 4,
4,
       3, 1, 4, 1, 2, 4, 0, 5, 1, 2, 2, 3, 3, 2, 3, 4, 2, 4, 2, 0, 4,
4,
       2, 0, 1, 0, 5, 3, 2, 0, 4, 2, 1, 3, 4, 5, 4, 3, 5, 1, 1, 2, 1,
2,
       1, 1, 0, 5, 4, 5, 1, 0, 5, 3, 4, 0, 5, 3, 1, 1, 1, 5, 1, 1, 5,
4,
       1, 0, 3, 1, 1, 3, 5, 5, 4, 4, 3, 4, 3, 4, 4, 3, 3, 4, 2, 1, 1,
0,
       1, 5, 4, 5, 0, 1, 0, 4, 1, 2, 2, 3, 5, 4, 3, 2, 5, 4, 1, 3, 4,
0,
       2, 0, 5, 5, 4, 0, 2, 0, 3, 5, 4, 1, 5, 1, 3, 3, 5, 4, 1, 1, 4,
5,
       0, 1, 2, 3, 1, 3, 3, 4, 1, 3, 3, 5, 1, 4, 2, 1, 2, 2, 1, 4, 0,
4,
       3, 0, 4, 4, 5, 0, 4, 1, 3, 4, 0, 3, 3, 4, 0, 5, 2, 5, 1, 4, 2,
2,
       0, 1, 1, 5, 5, 0, 1, 2, 4, 0, 0, 2, 4, 1, 2, 5, 1, 4, 1, 1, 3,
0,
       3, 2, 5, 3, 4, 1, 0, 2, 2, 1, 1, 4, 0, 4, 3, 1, 5, 3, 1, 5, 3,
5,
       2, 0, 5, 2, 2, 5, 3, 4, 3, 0, 0, 2, 0, 5, 5, 3, 5, 2, 4, 5, 5,
```

```
0,
       2, 0, 1, 5, 5, 1, 2, 2, 5, 0, 4, 0, 3, 1, 4, 5, 2, 0, 1, 2, 5,
4,
       4, 1, 3, 0, 2, 5, 4, 4, 0, 0, 3, 3, 1, 1, 2, 0, 0, 2, 0, 5, 5,
0,
       2, 5, 3, 1, 0, 3, 3, 4, 2, 3, 4, 2, 1, 3, 1, 0, 5, 5, 2, 1, 0,
1,
       2, 0, 0, 5, 0, 4, 2, 2, 0, 2, 4, 3, 2, 2, 4, 4, 4, 3, 4, 5, 1,
0,
       1, 4, 4, 1, 5, 0, 3, 2, 4, 4, 1, 5, 5, 0, 3, 5, 5, 3, 2, 3, 5,
1,
       4, 0, 0, 1, 0, 5, 0, 0, 3, 3, 0, 0, 2, 0, 1, 5, 0, 3, 2, 1, 5,
5,
       2, 3, 0, 4, 3, 1, 1, 1, 2, 2, 1, 2, 2, 1, 4, 0, 4, 2, 1, 2, 0,
3,
       3, 5, 0, 4, 0, 2, 5, 5, 5, 5, 5, 5, 5, 1, 4, 1, 4, 4, 1, 4, 5,
2,
       2, 0, 0, 1, 2, 5, 1, 3, 1, 3, 5, 0, 4, 3, 0, 1, 5, 2, 2, 2, 2,
2,
       2, 0, 4, 5, 0, 3, 4, 0, 2, 4, 2, 1, 1, 0, 5, 3, 1, 5, 4, 0, 4,
5,
       4, 5, 1, 2, 4, 3, 2, 5, 2, 4, 5, 0, 1, 2, 0, 0, 4, 0, 4, 2, 2,
3,
       0, 5, 2, 2, 3, 2, 4, 5, 1, 0, 0, 2, 5, 5, 4, 2, 3, 1, 5, 0, 2,
5,
       5, 0, 4, 4, 1, 3, 5, 0, 1, 1, 2, 1, 3, 1, 5, 5, 1, 2, 0, 3, 2,
2,
       3, 4, 3, 4, 2, 1, 1, 5, 0, 2, 4, 3, 3, 4, 2, 3, 3, 0, 5, 2, 0,
3,
       0, 0, 2, 1, 3, 2, 4, 4, 5, 0, 4, 5, 2, 4, 2, 3, 0, 1, 3, 0, 0,
3,
       2, 0, 2, 1, 0, 3, 0, 1, 5, 1, 5, 1, 4, 2, 0, 1, 5, 1, 3, 4, 2,
1,
       1, 0, 3, 4, 5, 1, 5, 1, 4, 0, 0, 0, 2, 0, 5, 2, 0, 4, 4, 1, 2,
1,
       2, 0, 1, 4, 0, 5, 0, 5, 3, 0, 4, 3, 2, 5, 2, 0, 1, 0, 3, 0, 0,
4,
       1, 1, 2, 5, 4, 3, 0, 0, 0, 2, 2, 5, 5, 1, 3, 4, 4, 4, 2, 2, 0,
5,
       3, 1, 4, 3, 5, 4, 1, 0, 1, 5, 3, 1, 0, 3, 5, 3, 4, 5, 3, 0, 2,
4,
       1, 3, 2, 0, 0, 5, 5, 2, 5, 5, 2, 2, 4, 5, 2, 1, 2, 0, 5, 5, 1,
2,
       1, 0, 1, 2], dtype=int64)
```

```
y_test
```

```
array([[0., 0., 0., 1., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0.],
       ...,
```

```
       [1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.]], dtype=float32)
```

```
# Finding out maximum value to get the original data
y_test = np.argmax(y_test,axis=1)
y_test
```

```
array([3, 2, 4, 4, 1, 1, 2, 3, 2, 3, 5, 4, 2, 0, 5, 3, 2, 2, 5, 1, 3,
0,
       2, 3, 1, 0, 3, 0, 2, 5, 2, 0, 5, 3, 0, 3, 2, 5, 2, 4, 5, 5, 3,
4,
       4, 2, 4, 2, 1, 5, 3, 3, 2, 3, 3, 0, 1, 4, 1, 3, 3, 1, 3, 0, 5,
1,
       0, 4, 4, 3, 1, 0, 2, 4, 1, 2, 1, 1, 4, 0, 5, 3, 2, 3, 3, 5, 5,
2,
       5, 1, 0, 3, 3, 2, 3, 3, 3, 4, 5, 5, 1, 5, 2, 2, 3, 1, 3, 0, 4,
4,
       3, 1, 4, 3, 2, 3, 0, 5, 1, 2, 2, 2, 3, 2, 3, 4, 2, 4, 2, 0, 4,
4,
       2, 0, 1, 0, 5, 2, 3, 0, 4, 2, 3, 3, 4, 5, 4, 3, 5, 1, 1, 2, 1,
2,
       1, 1, 0, 5, 4, 5, 1, 0, 5, 3, 4, 0, 5, 3, 1, 1, 1, 5, 1, 1, 5,
3,
       3, 0, 3, 1, 1, 2, 5, 5, 4, 4, 2, 4, 2, 4, 2, 3, 2, 4, 3, 1, 1,
0,
       1, 5, 4, 5, 0, 1, 0, 4, 1, 2, 1, 3, 5, 4, 2, 3, 5, 4, 1, 2, 4,
0,
       2, 0, 5, 5, 4, 0, 2, 0, 3, 5, 4, 1, 5, 1, 2, 3, 5, 4, 1, 1, 4,
5,
       0, 1, 3, 3, 1, 2, 3, 4, 1, 3, 3, 5, 1, 4, 2, 1, 2, 2, 1, 4, 0,
3,
       3, 0, 4, 4, 5, 0, 4, 1, 3, 3, 0, 3, 3, 3, 0, 5, 2, 5, 2, 3, 2,
2,
       0, 1, 1, 5, 5, 0, 1, 2, 4, 0, 0, 2, 4, 1, 3, 5, 1, 4, 1, 1, 3,
0,
       3, 2, 5, 3, 4, 1, 0, 3, 2, 1, 1, 4, 0, 4, 3, 1, 5, 3, 1, 5, 3,
5,
       2, 0, 5, 3, 3, 5, 3, 3, 2, 0, 0, 3, 0, 5, 5, 3, 5, 2, 4, 5, 5,
0,
       2, 0, 1, 5, 5, 1, 2, 2, 5, 2, 4, 0, 3, 1, 4, 5, 3, 0, 1, 2, 5,
3,
       4, 1, 2, 0, 2, 5, 4, 4, 0, 0, 2, 3, 1, 1, 2, 0, 0, 3, 0, 5, 5,
0,
       2, 5, 3, 1, 0, 3, 2, 3, 2, 3, 4, 2, 1, 3, 2, 0, 3, 5, 2, 1, 0,
1,
       2, 0, 0, 5, 0, 4, 2, 2, 0, 2, 4, 3, 3, 2, 4, 4, 4, 3, 4, 5, 1,
0,
       1, 4, 4, 1, 5, 0, 3, 2, 4, 4, 1, 5, 5, 0, 2, 5, 5, 3, 3, 3, 5,
1,
       4, 0, 0, 1, 2, 5, 0, 0, 3, 3, 0, 0, 2, 0, 1, 5, 0, 3, 3, 1, 5,
```

```
5,
       2, 3, 0, 4, 3, 1, 1, 1, 2, 2, 1, 3, 3, 1, 4, 0, 4, 2, 1, 2, 0,
3,
       3, 5, 2, 4, 0, 2, 5, 5, 5, 5, 5, 5, 4, 1, 4, 1, 4, 4, 1, 4, 5,
2,
       2, 0, 0, 1, 2, 5, 1, 2, 1, 2, 5, 0, 4, 3, 0, 1, 5, 2, 2, 2, 2,
3,
       2, 0, 3, 5, 2, 3, 4, 0, 2, 4, 2, 1, 1, 0, 5, 3, 3, 5, 4, 0, 4,
5,
       4, 5, 3, 2, 4, 2, 3, 5, 3, 4, 5, 0, 1, 2, 0, 0, 4, 0, 4, 2, 2,
3,
       0, 5, 2, 2, 2, 2, 4, 5, 1, 0, 0, 2, 5, 5, 4, 2, 3, 1, 5, 3, 3,
5,
       5, 0, 4, 4, 1, 3, 5, 0, 1, 1, 2, 1, 2, 1, 5, 5, 1, 2, 0, 3, 2,
2,
       2, 4, 3, 4, 2, 1, 1, 5, 0, 2, 4, 3, 3, 3, 3, 3, 3, 0, 5, 2, 0,
3,
       0, 0, 2, 1, 3, 3, 4, 4, 5, 4, 4, 5, 3, 4, 2, 3, 0, 1, 3, 0, 0,
3,
       2, 0, 3, 1, 0, 3, 0, 1, 5, 1, 5, 1, 4, 3, 2, 1, 5, 2, 3, 4, 3,
1,
       1, 0, 3, 4, 5, 1, 5, 1, 4, 0, 0, 0, 2, 0, 5, 2, 0, 4, 4, 1, 2,
1,
       2, 0, 1, 4, 0, 5, 0, 5, 3, 0, 4, 3, 2, 3, 2, 0, 1, 0, 3, 0, 0,
4,
       1, 1, 2, 5, 4, 3, 0, 0, 0, 2, 2, 5, 5, 1, 3, 4, 4, 4, 2, 2, 0,
5,
       3, 1, 4, 3, 5, 4, 1, 0, 1, 5, 3, 3, 0, 3, 5, 3, 3, 5, 3, 0, 3,
2,
       2, 3, 2, 0, 0, 5, 5, 2, 5, 5, 2, 2, 4, 5, 3, 1, 2, 2, 5, 5, 1,
2,
       2, 0, 1, 2], dtype=int64)
```

```python
# Accuracy Score
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)*100
```

```
87.28606356968214
```

```python
# Classification Report
from sklearn.metrics import classification_report
print(classification_report(y_pred,y_test))
```

```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96       142
           1       0.99      0.90      0.94       145
           2       0.73      0.75      0.74       145
           3       0.61      0.79      0.69       120
           4       0.98      0.89      0.93       128
           5       1.00      0.97      0.99       138
```

accuracy			0.87	818
macro avg	0.89	0.87	0.88	818
weighted avg	0.89	0.87	0.88	818