```
from sklearn.datasets import load_wine
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
dataset=load_wine()
```

```
df=pd.DataFrame(dataset['data'],columns=dataset['feature_names'])
df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	n
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

```
df['Wine Quality']=dataset["target"]
df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	n
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

```
dataset['target_names']
```

```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

```
df['wine_name']=df['Wine Quality'].replace(to_replace=[0,1,2],value=dataset['target_names'])
```

```
df['wine name']=df['wine Quality'].replace(to_replace=[0,1,2],value=dataset['target_names'])
df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	n
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

```
df.tail()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76

```
from sklearn.model_selection import train_test_split
X=df.iloc[:,0:13]
Y=df.iloc[:,14:15]
xtrain,xtest,ytrain,ytest=train_test_split(X,Y,test_size=0.20,random_state=1)
```

```
xtest.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
161	13.69	3.26	2.54	20.0	107.0	1.83	0.56
117	12.42	1.61	2.19	22.5	108.0	2.00	2.09
19	13.64	3.10	2.56	15.2	116.0	2.70	3.03

```
print("Xtest shape",xtest.shape)
print("Xtrain shape",xtrain.shape)
print("Ytrain shape",ytrain.shape)
```

```
     Xtest shape (36, 13)
     Xtrain shape (142, 13)
     Ytrain shape (142, 1)
```

```
from sklearn.preprocessing import  MinMaxScaler
```

```
scaler=MinMaxScaler()
xtrain_transform=scaler.fit_transform(xtrain)
xtest_transform=scaler.fit_transform(xtest)
```

```
xtrain_transform[0:10]
```

```
     array([[0.25526316, 0.1244898 , 0.56684492, 0.63687151, 0.17391304,
             0.16206897, 0.19198312, 0.74      , 0.38485804, 0.19795222,
             0.43103448, 0.50549451, 0.12268188],
            [0.44473684, 0.18571429, 0.44919786, 0.45810056, 0.17391304,
             0.42068966, 0.46202532, 0.26      , 0.42902208, 0.22354949,
             0.52586207, 0.68498168, 0.31098431],
            [0.27631579, 0.1       , 0.60962567, 0.66480447, 0.15217391,
             0.54482759, 0.41139241, 0.6       , 0.19873817, 0.13822526,
             0.32758621, 0.7032967 , 0.07631954],
            [0.80789474, 0.22857143, 0.55614973, 0.45810056, 0.35869565,
             0.61034483, 0.5443038 , 0.38      , 0.6214511 , 0.41979522,
             0.44827586, 0.54212454, 0.55777461],
            [0.71315789, 0.15714286, 0.47593583, 0.32402235, 0.52173913,
             0.55862069, 0.54008439, 0.16      , 0.38170347, 0.38993174,
             0.31896552, 0.70695971, 0.55777461],
            [0.35263158, 0.00816327, 0.        , 0.        , 0.19565217,
             0.34482759, 0.04852321, 0.3       , 0.00315457, 0.05716724,
             0.43103448, 0.2014652 , 0.17261056],
            [0.71052632, 0.70612245, 0.48128342, 0.66480447, 0.19565217,
             0.10344828, 0.02742616, 0.78      , 0.23343849, 0.4556314 ,
             0.19827586, 0.17582418, 0.17261056],
            [0.83947368, 0.63061224, 0.61497326, 0.1452514 , 0.63043478,
             0.69655172, 0.56962025, 0.14      , 0.52681388, 0.32593857,
             0.29310345, 0.82783883, 0.34379458],
            [0.83947368, 0.16326531, 0.5026738 , 0.31843575, 0.52173913,
             0.76551724, 0.56118143, 0.26      , 0.51104101, 0.43515358,
             0.3362069 , 0.74725275, 0.4935806 ],
            [0.53157895, 0.17755102, 0.39572193, 0.3575419 , 0.40217391,
             0.69655172, 0.56118143, 0.3       , 0.51104101, 0.32081911,
             0.28448276, 0.76190476, 0.43295292]])
```

```
from sklearn.neural_network import MLPClassifier
```

```
xtest_transform[0:10]
```

```
array([[0.76767677, 0.65116279, 0.67521368, 0.44444444, 0.34328358,
        0.234375  , 0.02531646, 0.69230769, 0.        , 0.39422085,
        0.57831325, 0.23043478, 0.29797571],
       [0.34006734, 0.2248062 , 0.37606838, 0.58333333, 0.35820896,
        0.32291667, 0.50949367, 0.38461538, 0.375     , 0.        ,
        0.69879518, 0.72608696, 0.02672065],
       [0.75084175, 0.60981912, 0.69230769, 0.17777778, 0.47761194,
        0.6875    , 0.80696203, 0.05769231, 0.39814815, 0.31372549,
        0.57831325, 0.9       , 0.43157895],
       [0.26936027, 0.11627907, 0.        , 0.26666667, 1.        ,
        0.24479167, 0.25316456, 0.        , 0.78703704, 0.08152735,
        0.96385542, 0.77391304, 0.32874494],
       [0.79461279, 0.2997416 , 0.79487179, 0.28333333, 0.46268657,
        0.84375   , 0.73101266, 0.48076923, 0.40740741, 0.4375645 ,
        0.78313253, 0.71304348, 0.86072874],
       [0.7003367 , 0.73643411, 0.37606838, 0.41666667, 0.05970149,
        0.125     , 0.        , 0.84615385, 0.03703704, 0.37564499,
        0.39759036, 0.23043478, 0.21700405],
       [0.11784512, 0.50129199, 1.        , 0.44444444, 0.28358209,
        0.19270833, 0.49050633, 0.88461538, 0.11574074, 0.17956656,
        0.90361446, 0.52608696, 0.2388664 ],
       [1.        , 0.29198966, 0.53846154, 0.        , 0.26865672,
        1.        , 1.        , 0.28846154, 1.        , 0.56140351,
        0.86746988, 0.74347826, 1.        ],
       [0.31986532, 0.79844961, 0.53846154, 0.5       , 0.05970149,
        0.47916667, 0.13924051, 0.69230769, 0.11111111, 0.57688338,
        0.09638554, 0.12608696, 0.16842105],
       [0.44107744, 0.26098191, 0.45299145, 0.58333333, 0.        ,
        0.        , 0.40506329, 0.65384615, 0.38425926, 0.12796698,
        0.48192771, 0.49130435, 0.14251012]])
```

```
model=MLPClassifier(hidden_layer_sizes=(300,200,100),alpha=0.0001,activation='relu',max_iter=
```

```
model.fit(xtrain_transform,ytrain)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py
  y = column_or_1d(y, warn=True)
MLPClassifier(hidden_layer_sizes=(300, 200, 100), max_iter=300)
```

```
ypred=model.predict(xtest_transform)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

```
accuracy=accuracy_score(ytest,ypred)
print("accuracy is",accuracy)
```

```
accuracy is 0.9722222222222222
```

```
cm=confusion_matrix(ytest,ypred)
cm
```

```
array([[14,  0,  0],
       [ 0, 13,  0],
       [ 0,  1,  8]])
```

```
from sklearn.metrics import classification_report
cr=classification_report(ytest,ypred)
```

```
print(cr)
```

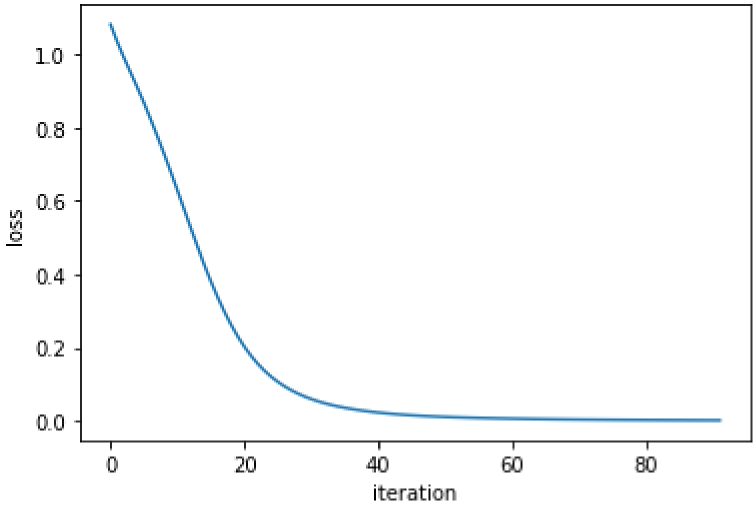
```
              precision    recall  f1-score   support

     class_0       1.00      1.00      1.00        14
     class_1       0.93      1.00      0.96        13
     class_2       1.00      0.89      0.94         9

    accuracy                           0.97        36
   macro avg       0.98      0.96      0.97        36
weighted avg       0.97      0.97      0.97        36
```

```
plt.plot(model.loss_curve_)
plt.xlabel("iteration")
plt.ylabel("loss")
```

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



✓ 0s completed at 10:46 AM ● ✕