```
#Implement MLP classifier using sklearn library


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from  sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.neural_network import MLPClassifier


import warnings
warnings.filterwarnings("ignore")
df = pd.read_csv("drug200.csv")
df
```

| | Age | Sex | BP | Cholesterol | Na_to_K | Drug |
|---|---|---|---|---|---|---|
| **0** | 23 | F | HIGH | HIGH | 25.355 | DrugY |
| **1** | 47 | M | LOW | HIGH | 13.093 | drugC |
| **2** | 47 | M | LOW | HIGH | 10.114 | drugC |
| **3** | 28 | F | NORMAL | HIGH | 7.798 | drugX |
| **4** | 61 | F | LOW | HIGH | 18.043 | DrugY |
| **...** | ... | ... | ... | ... | ... | ... |
| **195** | 56 | F | LOW | HIGH | 11.567 | drugC |
| **196** | 16 | M | LOW | HIGH | 12.006 | drugC |
| **197** | 52 | M | NORMAL | HIGH | 9.894 | drugX |
| **198** | 23 | M | NORMAL | NORMAL | 14.020 | drugX |
| **199** | 40 | F | LOW | NORMAL | 11.349 | drugX |

200 rows × 6 columns

```
from sklearn.preprocessing import LabelEncoder
le_class = LabelEncoder()
df['sex_code'] = le_class.fit_transform(df['Sex'])
df['BP_code']=le_class.fit_transform(df['BP'])
df['Cholesterol_code']=le_class.fit_transform(df['Cholesterol'])
df['Drug_code']=le_class.fit_transform(df['Drug'])


selected_columns = df[["Age","Na_to_K","sex_code","BP_code","Cholesterol_code","Drug_code"
```

```
x = selected_columns.iloc[:,:-1]
x
```

| | Age | Na_to_K | sex_code | BP_code | Cholesterol_code |
|---|---|---|---|---|---|
| **0** | 23 | 25.355 | 0 | 0 | 0 |
| **1** | 47 | 13.093 | 1 | 1 | 0 |
| **2** | 47 | 10.114 | 1 | 1 | 0 |
| **3** | 28 | 7.798 | 0 | 2 | 0 |
| **4** | 61 | 18.043 | 0 | 1 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **195** | 56 | 11.567 | 0 | 1 | 0 |
| **196** | 16 | 12.006 | 1 | 1 | 0 |
| **197** | 52 | 9.894 | 1 | 2 | 0 |
| **198** | 23 | 14.020 | 1 | 2 | 1 |
| **199** | 40 | 11.349 | 0 | 1 | 1 |

200 rows × 5 columns

```
y = selected_columns.iloc[:,5:]
```

```
mlp = MLPClassifier(hidden_layer_sizes=(100,100,100,100,100),activation="tanh",max_iter=30
```

```
mlp
```

```
MLPClassifier(activation='tanh', hidden_layer_sizes=(100, 100, 100, 100, 100),
              max_iter=300)
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

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

```
(160, 5)
(160, 1)
(40, 5)
(40, 1)
```

```
mlp.fit(x_train,y_train)
```

```
MLPClassifier(activation='tanh', hidden_layer_sizes=(100, 100, 100, 100, 100),
              max_iter=300)
```

```
prediction = mlp.predict(x_test)
prediction
```

```
array([0, 1, 4, 4, 0, 0, 0, 4, 0, 2, 0, 0, 0, 4, 2, 4, 3, 0, 4, 0, 1, 0,
       0, 0, 0, 0, 4, 4, 0, 4, 0, 0, 4, 4, 0, 4, 3, 3, 1, 4])
```

```
confusion_matrix(y_test,prediction)
```

```
array([[18,  0,  0,  0,  0],
       [ 0,  3,  0,  0,  0],
       [ 1,  0,  2,  0,  0],
       [ 0,  0,  0,  3,  1],
       [ 0,  0,  0,  0, 12]])
```

```
print(accuracy_score(prediction,y_test))
```

```
0.95
```

```
#NOW BY Using Relu
```

```
mlp = MLPClassifier(hidden_layer_sizes=(100,100,100,100,100),max_iter=300,activation='relu
```

```
mlp.fit(x_train,y_train)
```

```
MLPClassifier(hidden_layer_sizes=(100, 100, 100, 100, 100), max_iter=300)
```

```
prediction = mlp.predict(x_test)
prediction
```

```
array([0, 1, 4, 4, 0, 0, 0, 0, 0, 2, 0, 0, 0, 4, 1, 4, 3, 0, 4, 0, 1, 0,
       0, 0, 0, 0, 4, 4, 0, 4, 0, 0, 4, 3, 0, 4, 3, 3, 1, 4])
```

```
confusion_matrix(y_test,prediction)
```

```
array([[18,  0,  0,  0,  0],
       [ 0,  3,  0,  0,  0],
       [ 1,  1,  1,  0,  0],
       [ 0,  0,  0,  4,  0],
       [ 1,  0,  0,  0, 11]])
```

```
print(accuracy_score(prediction,y_test))
```

```
0.925
```