# MLP with Confusion Matrix

```
import pandas as pd
import numpy as np

from sklearn.datasets import load_breast_cancer
```

```
cancer_data = load_breast_cancer()

x = cancer_data.data
y = cancer_data.target
```

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_stat
```

```
##Importing MLPClassifier
from sklearn.neural_network import MLPClassifier
#Initializing the MLPClassifier
#solver is for weight optimization
#max_iter sets limit till convergence or this value
#activation is the activation function for the hidden layer
mlpC = MLPClassifier(hidden_layer_sizes=(150,100,50), max_iter=300, activation='relu'

#training the model
mlpC.fit(x_train, y_train)
```

```
MLPClassifier(hidden_layer_sizes=(150, 100, 50), max_iter=300, random_state=1)
```

```
y_pred=mlpC.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
cf_matrix = confusion_matrix(y_test, y_pred)
print(cf_matrix)
```

```
[[47  5]
 [ 3 88]]
```

# HOMEWORK 01

```
#Logistic Regression on breast_cancer dataset
import numpy as np
import pandas as pd

from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

```
cancer_dataset = load_breast_cancer()
```

```
x=cancer_dataset.data
y=cancer_dataset.target

```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state
```

```
cancerLR = LogisticRegression(max_iter=30000)
cancerLR.fit(x_train, y_train)

```

```
LogisticRegression(max_iter=30000)
```

```
y_predLR = cancerLR.predict(x_test)
```

```
cancerLR.score(x_test, y_test)
```

```
0.9230769230769231
```

```
cf_matrixLR = confusion_matrix(y_test, y_pred)
print(cf_matrixLR)
```

```
[[47  5]
 [ 3 88]]
```

# HOMEWORK 02

```
#Decision Tree Classifier on breast_cancer dataset
import numpy as np
import pandas as pd

from sklearn.datasets import load_breast_cancer
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

```
cancer_dataset = load_breast_cancer()
```

```
x=cancer_dataset.data
y=cancer_dataset.target

```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state
```

```
cancerDTC = DecisionTreeClassifier(criterion="entropy")
cancerDTC.fit(x_train, y_train)
```

```
DecisionTreeClassifier(criterion='entropy')
```

```
cancerDTC.score(x_test, y_test)
```

```
0.9300699300699301
```

```
cf_matrixDTC = confusion_matrix(y_test, y_pred)
print(cf_matrixDTC)
```

```
[[47  5]
 [ 3 88]]
```

```
'''
From the above experiment the confusion matrix of the Logistic Regression
and Decision Tree Classifier is same
MLP Classifier has the least number of FP(False Positives) and FN(False Negatives)
So clearly the MLP classifier would form the best suited for deploying as a
real-time smart cancer diagnosis system.
'''
```