Binary Classification_Sk_NN

```
#import section
import sklearn
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn import metrics
```

```
#load database
data = load_breast_cancer()
```

```
#examine data
x_names = data['feature_names']
y_names = data['target_names']
x = data['data']
y = data['target']
print(y_names)
print(x_names)
```

```
['malignant' 'benign']
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
mlp = MLPClassifier(hidden_layer_sizes=(200,), activation='logistic', alpha = 0.01, solve
```

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

```
Iteration 1, loss = 0.66423476
Iteration 2, loss = 0.62794413
Iteration 3, loss = 0.60989763
Iteration 4, loss = 0.59967550
Iteration 5, loss = 0.59427097
Iteration 6, loss = 0.58521329
Iteration 7, loss = 0.55965895
Iteration 8, loss = 0.55196990
Iteration 9, loss = 0.53932907
Iteration 10, loss = 0.52514628
Iteration 11, loss = 0.51722911
Iteration 12, loss = 0.50711408
Iteration 13, loss = 0.49787412
Iteration 14, loss = 0.49050556
Iteration 15, loss = 0.48752876
Iteration 16, loss = 0.47673948
Iteration 17, loss = 0.47993544
Iteration 18, loss = 0.46797245
Iteration 19, loss = 0.46441103
Iteration 20, loss = 0.45171874
Iteration 21, loss = 0.44151621
Iteration 22, loss = 0.43654456
Iteration 23, loss = 0.42940712
Iteration 24, loss = 0.42571736
Iteration 25, loss = 0.42065474
Iteration 26, loss = 0.41780376
Iteration 27, loss = 0.41532171
Iteration 28, loss = 0.40574464
Iteration 29, loss = 0.40541116
Iteration 30, loss = 0.40079164
Iteration 31, loss = 0.39621897
Iteration 32, loss = 0.38990156
Iteration 33, loss = 0.38724137
Iteration 34, loss = 0.38653433
Iteration 35, loss = 0.38053886
Iteration 36, loss = 0.37144850
Iteration 37, loss = 0.37419820
Iteration 38, loss = 0.36781577
Iteration 39, loss = 0.36358413
Iteration 40, loss = 0.38588840
Iteration 41, loss = 0.35720477
Iteration 42, loss = 0.35326649
Iteration 43, loss = 0.35139777
Iteration 44, loss = 0.35413033
Iteration 45, loss = 0.34662249
Iteration 46, loss = 0.34465118
Iteration 47, loss = 0.35473041
Iteration 48, loss = 0.33575499
Iteration 49, loss = 0.32968173
Iteration 50, loss = 0.33003552
Iteration 51, loss = 0.32563593
Iteration 52, loss = 0.32142269
Iteration 53, loss = 0.33429541
Iteration 54, loss = 0.32404276
Iteration 55, loss = 0.32392734
Iteration 56, loss = 0.32152923
Iteration 57, loss = 0.31658237
Iteration 58, loss = 0.30715736
Iteration 59, loss = 0.30706361
Iteration 60, loss = 0.31783154
```

```
Iteration 61, loss = 0.30688478
Iteration 62, loss = 0.30540081
Iteration 63, loss = 0.30700401
Iteration 64, loss = 0.30176876
Iteration 65, loss = 0.30186032
Iteration 66, loss = 0.30562930
Iteration 67, loss = 0.30408710
Iteration 68, loss = 0.29969768
Iteration 69, loss = 0.29255798
Iteration 70, loss = 0.30184832
Iteration 71, loss = 0.29473848
Iteration 72, loss = 0.30223574
Iteration 73, loss = 0.31478976
Iteration 74, loss = 0.28871198
Iteration 75, loss = 0.32356931
Iteration 76, loss = 0.29239020
Iteration 77, loss = 0.28854308
Iteration 78, loss = 0.31323790
Iteration 79, loss = 0.28649529
Iteration 80, loss = 0.28229284
Iteration 81, loss = 0.28659529
Iteration 82, loss = 0.29547102
Iteration 83, loss = 0.28898333
Iteration 84, loss = 0.27582272
Iteration 85, loss = 0.27657120
Iteration 86, loss = 0.27553380
Iteration 87, loss = 0.27425651
Iteration 88, loss = 0.27275920
Iteration 89, loss = 0.27612397
Iteration 90, loss = 0.27600615
Iteration 91, loss = 0.27009301
Iteration 92, loss = 0.27514190
Iteration 93, loss = 0.26935071
Iteration 94, loss = 0.26879087
Iteration 95, loss = 0.28248209
Iteration 96, loss = 0.27654794
Iteration 97, loss = 0.27641710
Iteration 98, loss = 0.26993826
Iteration 99, loss = 0.27454449
Iteration 100, loss = 0.27336509
Iteration 101, loss = 0.27683239
Iteration 102, loss = 0.26768910
Iteration 103, loss = 0.31394857
Iteration 104, loss = 0.27696649
Iteration 105, loss = 0.28943155
Iteration 106, loss = 0.26700490
Iteration 107, loss = 0.26183720
Iteration 108, loss = 0.26543810
Iteration 109, loss = 0.26998549
Iteration 110, loss = 0.26144614
Iteration 111, loss = 0.26681276
Iteration 112, loss = 0.26515997
Iteration 113, loss = 0.26467451
Iteration 114, loss = 0.25689668
Iteration 115, loss = 0.25912646
Iteration 116, loss = 0.27503798
Iteration 117, loss = 0.33879276
Iteration 118, loss = 0.27424440
Iteration 119, loss = 0.25668529
Iteration 120, loss = 0.29455947
Iteration 121, loss = 0.27781186
```

```
Iteration 121, loss = 0.27781186
Iteration 122, loss = 0.31046773
Iteration 123, loss = 0.27691942
Iteration 124, loss = 0.26822355
Iteration 125, loss = 0.25596092
Iteration 126, loss = 0.25405997
Iteration 127, loss = 0.30208047
Iteration 128, loss = 0.26172731
Iteration 129, loss = 0.25153075
Iteration 130, loss = 0.27497685
Iteration 131, loss = 0.26104151
Iteration 132, loss = 0.26144125
Iteration 133, loss = 0.25332222
Iteration 134, loss = 0.27731017
```

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

```
Iteration 137, loss = 0.25555127
```

```
metrics.accuracy_score(y_test,predictions)
```

```
0.9210526315789473
```

```
MLPClassifier
MLPClassifier(activation='logistic', alpha=0.01, hidden_layer_sizes=(200,),
              random_state=1, solver='sgd', verbose=True)
```