MLP Classifier

April 24, 2019

In this project, I have used MLP classifer from neural networks to classify MLB data set

```
In [1]: import sklearn
        import numpy as np
        from sklearn.preprocessing import OneHotEncoder

In [2]: from sklearn.neural_network import MLPClassifier

In [3]: import pandas as pd
```

The above lines of code are ran to import the modules required to classify the prediction data.

The followig lines will create the data frame from the test data csv file that we have, then we are popping out the Position column to create of dependent variable dataframe, then popping out the not required columns and keeping the only required columns in our initial dataframe to use it as or dependent variables dataframe.

```
In [9]: test_dfa = pd.read_csv("mlb_test_data.csv")
        test_dfa.head()
```

Out[9]:

	Name	Team	Position	Height(inches)	Weight(pounds)	Age	BMI	LenFirstName	LenLastName
0	Marcus_Thames	DET	Outfielder	74	205	29.98	26.318	6	6
1	Mark_DeRosa	CHC	Outfielder	73	205	32.01	27.044	4	6
2	Mark_Ellis	OAK	Infielder	71	180	29.73	25.102	4	5
3	Mark_Kiger	OAK	Infielder	71	180	26.75	25.102	4	5
4	Mark_Kotsay	OAK	Outfielder	72	201	31.24	27.258	4	6

```
In [10]: test_dfa = pd.read_csv("mlb_test_data.csv")
         test_dfa = test_dfa.dropna()
         test_y = test_dfa.pop("Position")
         test_dfa.pop("Name")

         test_dfa.pop("Team")
```

```
test_dfa.pop("LenFirstName")

test_dfa.pop("LenLastName")

test_x = test_dfa
```

In the following lines, we are doing the same for the training data as we did for the test data.

```
In [11]: train_dfa = pd.read_csv("mlb_train_data1.csv")

         train_dfa.pop("Name")

         train_dfa.pop("Team")

         train_dfa.pop("LenFirstName")

         train_dfa.pop("LenLastName")

         train_y = train_dfa.pop("Position")



         train_x = train_dfa
```

In the following lines we are using the MLPClassifier function in the neural networks module of the Sklearn module to fit the training data.

Then the fitted data is tested to get the score.

```
In [16]: clf = MLPClassifier(hidden_layer_sizes=100)
         clf.fit(train_x,train_y)
         print(clf)
         score = clf.score(test_x,test_y)
         print(score)
```

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=100, learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
       random_state=None, shuffle=True, solver='adam', tol=0.0001,
       validation_fraction=0.1, verbose=False, warm_start=False)
0.6263736263736264
```

In the below line we are finding the predictions of position based on inputted test data on the model developed.

```
In [17]: from sklearn.metrics import confusion_matrix
         predict_y = clf.predict(test_x)
```

In the following lines we are creating a new data frame, to show the predicted position and the actual position along with the independent variables

```
In [18]: b = test_x
         b["Predicted Postion"] = predict_y.reshape((-1,1))
         b["Actual Position"] = test_y
         b
```

Out[18]:

	Height(inches)	Weight(pounds)	Age	BMI	Predicted Postion	\
0	74	205	29.98	26.318	Catcher	
1	73	205	32.01	27.044	Outfielder	
2	71	180	29.73	25.102	Infielder	
3	71	180	26.75	25.102	Infielder	
4	72	201	31.24	27.258	Outfielder	
5	73	170	23.34	22.426	Infielder	
6	77	230	32.34	27.271	Pitcher	
7	75	238	23.49	29.745	Pitcher	
8	75	245	27.14	30.620	Pitcher	
9	74	220	30.99	28.243	Pitcher	
10	75	210	30.35	26.245	Pitcher	
11	71	197	23.64	27.473	Catcher	
12	74	190	32.08	24.392	Infielder	
13	72	215	28.63	29.156	Catcher	
14	74	200	34.14	25.676	Infielder	
15	74	180	26.33	23.108	Infielder	
16	72	190	35.12	25.766	Infielder	
17	72	205	25.33	27.800	Catcher	
18	75	215	38.49	26.870	Outfielder	
19	71	200	35.82	27.891	Outfielder	
20	72	210	30.48	28.478	Catcher	
21	75	220	40.66	27.495	Outfielder	
22	69	176	29.31	25.988	Infielder	
23	72	176	25.18	23.867	Infielder	
24	69	175	39.85	25.840	Infielder	
25	72	185	29.22	25.088	Infielder	
26	71	180	38.11	25.102	Infielder	
27	72	150	22.41	20.341	Infielder	
28	70	186	32.51	26.685	Infielder	
29	76	223	30.39	27.141	Pitcher	
..	...	...	...	...	...	
61	72	220	38.30	29.834	Outfielder	
62	80	240	31.02	26.363	Pitcher	
63	74	180	23.29	23.108	Infielder	
64	78	240	26.98	27.732	Pitcher	
65	72	188	30.95	25.495	Infielder	
66	78	230	27.86	26.576	Pitcher	
67	78	235	26.06	27.154	Pitcher	
68	71	200	33.01	27.891	Outfielder	

69	76	200	34.30	24.342	Infielder
70	74	190	26.60	24.392	Infielder
71	70	163	37.66	23.386	Infielder
72	69	185	32.24	27.317	Outfielder
73	72	185	29.95	25.088	Infielder
74	75	240	31.36	29.995	Pitcher
75	77	230	26.47	27.271	Pitcher
76	75	230	38.85	28.745	Pitcher
77	72	185	33.77	25.088	Infielder
78	72	180	24.25	24.410	Infielder
79	73	180	25.94	23.746	Infielder
80	74	211	31.62	27.088	Outfielder
81	76	250	26.21	30.428	Pitcher
82	74	211	32.89	27.088	Outfielder
83	71	190	33.95	26.497	Infielder
84	74	190	28.19	24.392	Infielder
85	75	220	31.06	27.495	Pitcher
86	71	190	29.43	26.497	Infielder
87	72	175	23.98	23.732	Infielder
88	71	225	24.63	31.378	Pitcher
89	71	190	28.62	26.497	Catcher
90	72	230	22.27	31.190	Pitcher

	Actual Position
0	Outfielder
1	Outfielder
2	Infielder
3	Infielder
4	Outfielder
5	Infielder
6	Pitcher
7	Pitcher
8	Pitcher
9	Catcher
10	Catcher
11	Catcher
12	Catcher
13	Catcher
14	Outfielder
15	Infielder
16	Catcher
17	Catcher
18	Catcher
19	Catcher
20	Catcher
21	Outfielder
22	Infielder
23	Infielder

```
24       Infielder
25       Infielder
26      Outfielder
27       Infielder
28      Outfielder
29      Outfielder
..             ...
61      Outfielder
62         Pitcher
63       Infielder
64         Pitcher
65      Outfielder
66         Pitcher
67         Pitcher
68      Outfielder
69      Outfielder
70         Catcher
71      Outfielder
72       Infielder
73      Outfielder
74         Catcher
75         Pitcher
76         Pitcher
77       Infielder
78       Infielder
79       Infielder
80      Outfielder
81         Pitcher
82      Outfielder
83         Catcher
84         Catcher
85      Outfielder
86         Catcher
87       Infielder
88         Catcher
89         Catcher
90         Pitcher

[91 rows x 6 columns]
```

Through the following lines of code we are generating a confusion matrix to the get the details of how many times did it actual predict correct value for a given position.

```
In [19]: labelist = ["Pitcher","Outfielder","Infielder","Catcher"]
         cm =confusion_matrix(test_y,predict_y,labels = labelist)
         cm

Out[19]: array([[15,  0,  0,  0],
                [ 2, 10, 12,  1],
```

```
       [ 0,  1, 23,  0],
       [ 8,  3,  7,  9]], dtype=int64)
```

The below lines of code is used to get the accuracy of the model that is used to predict the positions, and also append it to the data frame we created to show the values

```
In [20]: #print('Observed_Pos          Predicted_Pos')
         num_correct = 0
         correct = []
         for i in range(0, len(test_y)):
             Yo = test_y[i]; Ys= predict_y[i]
             if(Yo==Ys):
                 correct.append('Yes')
                 num_correct += 1
             else:
                 correct.append('No')

         b["True/False"] = correct

In [21]: Accuracy = float(num_correct)/float(len(test_y))
         print("AccuracY = {:.2f}".format(Accuracy))

AccuracY = 0.63


In [22]: from sklearn.metrics import accuracy_score

In [23]: accuracy = accuracy_score(test_y,predict_y)

In [24]: accuracy

Out[24]: 0.6263736263736264
```

Through the mlp classification model developed, we have achieved an accuracy of 0.62, we might have got better accuracy with a decision tree or a logistic regression model.