Task

On

Deep Learning

**Course**: Artificial Intelligence

(Machine Learning & Deep Learning)

Name: Abbas Shafi

Father Name: Dr Muhammad Shafi

Week: 13

Submitted to: Sir Syed Nazir Afridi



National Vocational & Technical Training Commission

National Center For Big Data & Cloud Computing

University of Engineering & Technology

Peshawar

[1]:

**import numpy as np import pandas as pd**

[2]:

[2]: satisfaction\_level last\_evaluation number\_project average\_montly\_hours \

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0.38 | 0.53 | 2 | | 157 | |
| 1 | 0.80 | 0.86 | 5 | | 262 | |
| 2 | 0.11 | 0.88 | 7 | | 272 | |
| 3 | 0.72 | 0.87 | 5 | | 223 | |
| 4 | 0.37 | 0.52 | 2 | | 159 | |
| time\_spend\_company Work\_accident left promotion\_last\_5years sales \ | | | | | | |
| 0 | 3 | 0 | 1 | 0 | | sales |
| 1 | 6 | 0 | 1 | 0 | | sales |
| 2 | 4 | 0 | 1 | 0 | | sales |
| 3 | 5 | 0 | 1 | 0 | | sales |
| 4 | 3 | 0 | 1 | 0 | | sales |

[12]:

*# Import LabelEncoder*

**from sklearn.preprocessing import** LabelEncoder

*# Creating labelEncoder*

le =LabelEncoder()

*# Converting string labels into numbers.*

data['salary']=le.fit\_transform(data['salary'])

salary

1. low
2. medium
3. medium
4. low
5. low

[13]:

[14]:

[6]:

[6]: satisfaction\_level last\_evaluation number\_project average\_montly\_hours \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0.38 | 0.53 | 2 | 157 |
| 1 | 0.80 | 0.86 | 5 | 262 |
| 2 | 0.11 | 0.88 | 7 | 272 |
| 3 | 0.72 | 0.87 | 5 | 223 |
| 4 | 0.37 | 0.52 | 2 | 159 |

time\_spend\_company Work\_accident promotion\_last\_5years sales salary \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 3 | 0 | 0 sales | 1 |
| 1 | 6 | 0 | 0 sales | 2 |
| 2 | 4 | 0 | 0 sales | 2 |
| 3 | 5 | 0 | 0 sales | 1 |
| 4 | 3 | 0 | 0 sales | 1 |
| sales | left |  |  |  |

|  |  |  |
| --- | --- | --- |
| 0 | 7 | 1 |
| 1 | 7 | 1 |
| 2 | 7 | 1 |
| 3 | 7 | 1 |
| 4 | 7 | 1 |

[15]:

[16]:

[16]: satisfaction\_level last\_evaluation number\_project average\_montly\_hours \

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0.38 | 0.53 | 2 | 157 | | |
| 1 | 0.80 | 0.86 | 5 | 262 | | |
| 2 | 0.11 | 0.88 | 7 | 272 | | |
| 3 | 0.72 | 0.87 | 5 | 223 | | |
| 4 | 0.37 | 0.52 | 2 | 159 | | |
| time\_spend\_company Work\_accident promotion\_last\_5years salary sales | | | | | | |
| 0 | 3 | 0 | 0 | | 1 | 7 |
| 1 | 6 | 0 | 0 | | 2 | 7 |
| 2 | 4 | 0 | 0 | | 2 | 7 |
| 3 | 5 | 0 | 0 | | 1 | 7 |

\

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 |  | 3 | 0 | 0 | 1 | 7 |
| sales\_enc |  |  |  |  |  |  |
| 0 | 7 |  |  |  |  |  |
| 1 | 7 |  |  |  |  |  |
| 2 | 7 |  |  |  |  |  |
| 3 | 7 |  |  |  |  |  |
| 4 | 7 |  |  |  |  |  |

[17]:

*# Spliting data into Feature and*

*#X=data[['satisfaction\_level', 'last\_evaluation', 'number\_project',*␣

*‹→'average\_montly\_hours', 'time\_spend\_company', 'Work\_accident',*␣

*‹→'promotion\_last\_5years', 'sales ', 'salary']]*

*#y=data['left']*

X=dataset.iloc[:,:-1].values y=dataset.iloc[:,-1].values

*# Import train\_test\_split function*

**from sklearn.model\_selection import** train\_test\_split

*# Split dataset into training set and test set*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3,␣

*‹→*random\_state=42) *# 70% training and 30% test*

[29]:

*# Import MLPClassifer*

**from sklearn.neural\_network import** MLPClassifier

*# Create model object*

clf = MLPClassifier(hidden\_layer\_sizes=(7,5),

random\_state=7, verbose=**True**, learning\_rate\_init=0.01)

*# Fit data onto the model*

clf.fit(X\_train,y\_train)

Iteration 1, loss = 6.90918511

Iteration 2, loss = 2.15414132

Iteration 3, loss = 2.10590869

Iteration 4, loss = 2.09205923

Iteration 5, loss = 2.05215255

Iteration 6, loss = 2.02961262

Iteration 7, loss = 1.99408831

Iteration 8, loss = 1.97456292

Iteration 9, loss = 1.91849519

Iteration 10, loss = 1.85098936

Iteration 11, loss = 1.77045050

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | 12, | loss | = | 1.69458421 |
| Iteration | 13, | loss | = | 1.61406081 |
| Iteration | 14, | loss | = | 1.53471636 |
| Iteration | 15, | loss | = | 1.46095445 |
| Iteration | 16, | loss | = | 1.40015214 |
| Iteration | 17, | loss | = | 1.33898878 |
| Iteration | 18, | loss | = | 1.30118095 |
| Iteration | 19, | loss | = | 1.22413549 |
| Iteration | 20, | loss | = | 1.17645563 |
| Iteration | 21, | loss | = | 1.15265163 |
| Iteration | 22, | loss | = | 1.09213759 |
| Iteration | 23, | loss | = | 1.05569270 |
| Iteration | 24, | loss | = | 1.02473567 |
| Iteration | 25, | loss | = | 0.99291737 |
| Iteration | 26, | loss | = | 0.96860890 |
| Iteration | 27, | loss | = | 0.94798228 |
| Iteration | 28, | loss | = | 0.92197414 |
| Iteration | 29, | loss | = | 0.90261463 |
| Iteration | 30, | loss | = | 0.87421134 |
| Iteration | 31, | loss | = | 0.87418987 |
| Iteration | 32, | loss | = | 0.84334465 |
| Iteration | 33, | loss | = | 0.83188184 |
| Iteration | 34, | loss | = | 0.80104762 |
| Iteration | 35, | loss | = | 0.77809077 |
| Iteration | 36, | loss | = | 0.76207688 |
| Iteration | 37, | loss | = | 0.75378267 |
| Iteration | 38, | loss | = | 0.72752123 |
| Iteration | 39, | loss | = | 0.72428788 |
| Iteration | 40, | loss | = | 0.69021661 |
| Iteration | 41, | loss | = | 0.66565330 |
| Iteration | 42, | loss | = | 0.66986149 |
| Iteration | 43, | loss | = | 0.63476558 |
| Iteration | 44, | loss | = | 0.61739820 |
| Iteration | 45, | loss | = | 0.59360172 |
| Iteration | 46, | loss | = | 0.57837448 |
| Iteration | 47, | loss | = | 0.56510462 |
| Iteration | 48, | loss | = | 0.54926129 |
| Iteration | 49, | loss | = | 0.53787826 |
| Iteration | 50, | loss | = | 0.50101724 |
| Iteration | 51, | loss | = | 0.49391076 |
| Iteration | 52, | loss | = | 0.46411008 |
| Iteration | 53, | loss | = | 0.45482539 |
| Iteration | 54, | loss | = | 0.43455856 |
| Iteration | 55, | loss | = | 0.41863234 |
| Iteration | 56, | loss | = | 0.41713389 |
| Iteration | 57, | loss | = | 0.39444515 |
| Iteration | 58, | loss | = | 0.36456493 |
| Iteration | 59, | loss | = | 0.35167850 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | 60, | loss | = | 0.33745141 |
| Iteration | 61, | loss | = | 0.31046966 |
| Iteration | 62, | loss | = | 0.28627818 |
| Iteration | 63, | loss | = | 0.26681626 |
| Iteration | 64, | loss | = | 0.25768704 |
| Iteration | 65, | loss | = | 0.24620103 |
| Iteration | 66, | loss | = | 0.22315434 |
| Iteration | 67, | loss | = | 0.22039342 |
| Iteration | 68, | loss | = | 0.22023882 |
| Iteration | 69, | loss | = | 0.18319734 |
| Iteration | 70, | loss | = | 0.16585613 |
| Iteration | 71, | loss | = | 0.15567930 |
| Iteration | 72, | loss | = | 0.13905990 |
| Iteration | 73, | loss | = | 0.12716110 |
| Iteration | 74, | loss | = | 0.11967931 |
| Iteration | 75, | loss | = | 0.10862967 |
| Iteration | 76, | loss | = | 0.09768636 |
| Iteration | 77, | loss | = | 0.09074990 |
| Iteration | 78, | loss | = | 0.08304390 |
| Iteration | 79, | loss | = | 0.07550714 |
| Iteration | 80, | loss | = | 0.07022761 |
| Iteration | 81, | loss | = | 0.06458544 |
| Iteration | 82, | loss | = | 0.06098632 |
| Iteration | 83, | loss | = | 0.05504479 |
| Iteration | 84, | loss | = | 0.05267300 |
| Iteration | 85, | loss | = | 0.04562058 |
| Iteration | 86, | loss | = | 0.04282473 |
| Iteration | 87, | loss | = | 0.04289638 |
| Iteration | 88, | loss | = | 0.03587114 |
| Iteration | 89, | loss | = | 0.03294347 |
| Iteration | 90, | loss | = | 0.03076797 |
| Iteration | 91, | loss | = | 0.02948031 |
| Iteration | 92, | loss | = | 0.02655500 |
| Iteration | 93, | loss | = | 0.93331627 |
| Iteration | 94, | loss | = | 0.66269485 |
| Iteration | 95, | loss | = | 0.09911942 |
| Iteration | 96, | loss | = | 0.08455427 |
| Iteration | 97, | loss | = | 0.07757799 |
| Iteration | 98, | loss | = | 0.07021018 |
| Iteration | 99, | loss | = | 0.06526856 |

Iteration 100, loss = 0.06227956

Iteration 101, loss = 0.05861075

Iteration 102, loss = 0.05585657

Iteration 103, loss = 0.05368814

Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.

[29]: 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=(7, 5), learning\_rate='constant', learning\_rate\_init=0.01, max\_iter=200, momentum=0.9, n\_iter\_no\_change=10, nesterovs\_momentum=True, power\_t=0.5, random\_state=7, shuffle=True, solver='adam', tol=0.0001, validation\_fraction=0.1, verbose=True, warm\_start=False)

[31]:

*# Make prediction on test dataset*

y\_pred=clf.predict(X\_test)

*# Import accuracy score*

**from sklearn.metrics import** accuracy\_score

*# Calcuate accuracy*

accuracy\_score(y\_test,y\_pred)

[31]: 0.9991111111111111 [ ]:

[32]:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [[ | 371 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0] |
| [ | 0 | 243 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0] |
| [ | 0 | 0 | 228 | 0 | 0 | 0 | 0 | 0 | 0 | 0] |
| [ | 0 | 0 | 0 | 245 | 0 | 0 | 0 | 0 | 0 | 0] |
| [ | 0 | 0 | 0 | 0 | 152 | 0 | 0 | 0 | 0 | 0] |
| [ | 0 | 0 | 0 | 0 | 0 | 281 | 0 | 0 | 0 | 0] |
| [ | 0 | 0 | 0 | 0 | 0 | 0 | 240 | 0 | 0 | 0] |
| [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1207 | 0 | 0] |
| [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 689 | 4] |
| [ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 840]] |

[32]: 0.9991111111111111 [ ]: