Label Encoding

July 11, 2024

Converting the labels into numeric form.

classification machine learning proble; Is predicting if a datapoint belongs to one class or another. predicting if a person is diabetic or non diabetic.

[1]:
```
#Importing the dependencies

import pandas as pd
from sklearn.preprocessing import LabelEncoder  #labelEncoder function
```

Label Encoding of Breast CAncer Dataset

[2]:
```
cancer = pd.read_csv('breast_cancer_data.csv')
```

[3]:
```
#first 5 rows
cancer.head()
```

[3]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  …          17.33           184.60      2019.0            0.1622
1  …          23.41           158.80      1956.0            0.1238
2  …          25.53           152.50      1709.0            0.1444
3  …          26.50            98.87       567.7            0.2098
4  …          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
```

```
0            0.6656           0.7119            0.2654          0.4601
1            0.1866           0.2416            0.1860          0.2750
2            0.4245           0.4504            0.2430          0.3613
3            0.8663           0.6869            0.2575          0.6638
4            0.2050           0.4000            0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

[4]:
```
#finding the count of differrnt labels

cancer['diagnosis'].value_counts()
```

[4]:
```
diagnosis
B    357
M    212
Name: count, dtype: int64
```

Converting the 'B' and 'M to corresponing lables with numerical values

[5]:
```
#loading the label encoder function
#It labels with values between 0 and 1

label_encoder = LabelEncoder()
```

[6]:
```
#transforming values of diagnosis to 0 nad 1
labels = label_encoder.fit_transform(cancer.diagnosis)
```

[7]:
```
#Appending the labels to the data frame.
#It will create a new column call target for the label encoding

cancer['target'] = labels
```

[8]:
```
cancer.head()
```

[8]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0
```

```
   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  perimeter_worst  area_worst  smoothness_worst  compactness_worst  \
0  …           184.60      2019.0            0.1622             0.6656
1  …           158.80      1956.0            0.1238             0.1866
2  …           152.50      1709.0            0.1444             0.4245
3  …            98.87       567.7            0.2098             0.8663
4  …           152.20      1575.0            0.1374             0.2050

   concavity_worst  concave points_worst  symmetry_worst  \
0           0.7119                0.2654          0.4601
1           0.2416                0.1860          0.2750
2           0.4504                0.2430          0.3613
3           0.6869                0.2575          0.6638
4           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32  target
0                  0.11890          NaN       1
1                  0.08902          NaN       1
2                  0.08758          NaN       1
3                  0.17300          NaN       1
4                  0.07678          NaN       1

[5 rows x 34 columns]
```

0 = Benign 1 = Malig nant

[9]:
```
cancer['target'].value_counts()
```

[9]:
```
target
0    357
1    212
Name: count, dtype: int64
```

[ ]:
```
#Droping the diagnosiscolumn since we have a new column representing it
```

[ ]:
```

```

NOW USING IRIS DATASET, and it contains three label

Label Encoding of Iris Data

[12]:
```
iris = pd.read_csv('iris_data.csv')
```

[13]:
```
iris.head()
```

[13]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Transforming the labels to numerical value

[27]:
```
iris['Species'].value_counts()
```

[27]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

Loading the Label Encoder

[33]:
```
#Loading the Label Encoder
label_encode = LabelEncoder()
```

Creating the lables and Storing it in the Variable iris_label

[34]:
```
#Creating the lables and Storing it in the Variable iris_label

iris_labels = label_encode.fit_transform(iris.Species)
```

[36]:
```
#Appending the labels(new column called target) to our dataset
iris['target'] = iris_labels
```

[37]:
```
iris.head()
```

[37]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	target
0	1	5.1	3.5	1.4	0.2	Iris-setosa	0
1	2	4.9	3.0	1.4	0.2	Iris-setosa	0
2	3	4.7	3.2	1.3	0.2	Iris-setosa	0
3	4	4.6	3.1	1.5	0.2	Iris-setosa	0
4	5	5.0	3.6	1.4	0.2	Iris-setosa	0

[42]:
```
iris['target'].value_counts()
```

[42]:
```
target
0    50
1    50
2    50
Name: count, dtype: int64
```

[ ]:
```
Iris-satosa ---> 0
Iris-versicolor ---> 1
Iris-virginica --->2
```

[ ]:
```

```