HW 1 Basics for CS460

by Noah Caulfield

2/16/2024

italicized text

Prolem 1

```
import pandas as pd
import numpy as np

# Given dictionary
data = {
    'size': ['XL', 'L', 'M', np.nan, 'M', 'M'],
    'color': ['red', 'green', 'blue', 'green', 'red', 'green'],
    'gender': ['female', 'male', np.nan, 'female', 'female', 'male'],
    'price': [199.0, 89.0, np.nan, 129.0, 79.0, 89.0],
    'weight': [500, 450, 300, np.nan, 410, np.nan],
    'bought': ['yes', 'no', 'yes', 'no', 'yes', 'no']
}

df = pd.DataFrame(data)

missing_values_percent = df.isnull().mean()

adjusted_missing_values_percent = missing_values_percent.round(2)

# Print the adjusted values
print(adjusted_missing_values_percent)
```

```
size      0.17
color     0.00
gender    0.17
price     0.17
weight    0.33
bought    0.00
dtype: float64
```

Problem 2

```
from sklearn.datasets import load_iris

iris = load_iris()
data = iris.data
target = iris.target

print(data.shape)
print(target.shape)
```

```
(150, 4)
(150,)
```

Problem 3

```
import numpy as np
from sklearn.datasets import load_breast_cancer

# Load the Breast Cancer Data
raw_data = load_breast_cancer()

# Extract the data and target arrays
data = raw_data['data']
target = raw_data['target']

# Set numpy print options for better readability
np.set_printoptions(precision=2, suppress=True, linewidth=100)

# Print the first three elements of the data array
print(data[:3])
```

```
[[  17.99   10.38  122.8  1001.      0.12    0.28    0.3     0.15    0.24    0.08    1.09    0.91
     8.59  153.4     0.01    0.05    0.05    0.02    0.03    0.01   25.38   17.33  184.6  2019.
     0.16    0.67    0.71    0.27    0.46    0.12]
 [  20.57   17.77  132.9  1326.      0.08    0.08    0.09    0.07    0.18    0.06    0.54    0.73
     3.4    74.08    0.01    0.01    0.02    0.01    0.01    0.     24.99   23.41  158.8  1956.
     0.12    0.19    0.24    0.19    0.28    0.09]
 [  19.69   21.25  130.   1203.      0.11    0.16    0.2     0.13    0.21    0.06    0.75    0.79
     4.58   94.03    0.01    0.04    0.04    0.02    0.02    0.     23.57   25.53  152.5  1709.
     0.14    0.42    0.45    0.24    0.36    0.09]]
```

Problem 4

```
import numpy as np

# Combine the data and target arrays into a single array
all_data = np.c_[data, target]

# Print the first three rows of the combined array
print(all_data[:3])
```

```
[[  17.99   10.38  122.8  1001.      0.12    0.28    0.3     0.15    0.24    0.08    1.09    0.91
     8.59  153.4     0.01    0.05    0.05    0.02    0.03    0.01   25.38   17.33  184.6  2019.
     0.16    0.67    0.71    0.27    0.46    0.12    0.  ]
 [  20.57   17.77  132.9  1326.      0.08    0.08    0.09    0.07    0.18    0.06    0.54    0.73
     3.4    74.08    0.01    0.01    0.02    0.01    0.01    0.     24.99   23.41  158.8  1956.
     0.12    0.19    0.24    0.19    0.28    0.09    0.  ]
 [  19.69   21.25  130.   1203.      0.11    0.16    0.2     0.13    0.21    0.06    0.75    0.79
     4.58   94.03    0.01    0.04    0.04    0.02    0.02    0.     23.57   25.53  152.5  1709.
     0.14    0.42    0.45    0.24    0.36    0.09    0.  ]]
```

Problem 5

```
import pandas as pd

# Create a list of column names, ensuring consistency in types
column_names = list(raw_data['feature_names']) + ['target']

# Create the DataFrame, using list concatenation instead of addition
df = pd.DataFrame(all_data, columns=column_names)

# Print the first five rows of the DataFrame
print(df.head())
```

```
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
```

```
4        20.29         14.34          135.10     1297.0          0.10030

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   mean fractal dimension  ...  worst texture  worst perimeter  worst area  \
0                 0.07871  ...          17.33           184.60      2019.0
1                 0.05667  ...          23.41           158.80      1956.0
2                 0.05999  ...          25.53           152.50      1709.0
3                 0.09744  ...          26.50            98.87       567.7
4                 0.05883  ...          16.67           152.20      1575.0

   worst smoothness  worst compactness  worst concavity  worst concave points  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   worst symmetry  worst fractal dimension  target
0          0.4601                  0.11890     0.0
1          0.2750                  0.08902     0.0
2          0.3613                  0.08758     0.0
3          0.6638                  0.17300     0.0
4          0.2364                  0.07678     0.0

[5 rows x 31 columns]
```

Problem 6

```python
import pandas as pd


data = {
    "products": [
        "bread eggs",
        "bread eggs milk",
        "milk cheese",
        "bread butter cheese",
        "eggs milk",
        "bread milk butter cheese",
    ]
}


df = pd.DataFrame(data)

expanded = df['products'].str.split(' ', expand=True).fillna('None')

row_numbers = pd.Series(range(1, len(df) + 1))

formatted_df = pd.concat([row_numbers, expanded], axis=1)
formatted_df.columns = pd.RangeIndex(start=0, stop=len(formatted_df.columns), step=1)
# Header
final_str = '1.   0       1       2       3\n'
# Data rows
for index, row in formatted_df.iterrows():
    row_str = f"{index + 2}.  " + ' '.join(str(x).ljust(6) for x in row) + '\n'
    final_str += row_str
final_str = final_str.rstrip('\n')

print(final_str)
```

```
1.   0       1       2       3
2.  1      bread  eggs   None   None
3.  2      bread  eggs   milk   None
4.  3      milk   cheese None   None
5.  4      bread  butter cheese None
6.  5      eggs   milk   None   None
7.  6      bread  milk   butter cheese
```

Problem 7

```
import pandas as pd

data = {
    "products": [
        "bread eggs",
        "bread eggs milk",
        "milk cheese",
        "bread butter cheese",
        "eggs milk",
        "bread milk butter cheese",
    ]
}

df = pd.DataFrame(data)

products_list = df['products'].str.split(' ').explode().unique()
products_list.sort()

print(f"1. {list(products_list)}")
```

```
     1. ['bread', 'butter', 'cheese', 'eggs', 'milk']
```