Week 9 Quiz

Bryan Gibson - brg2130

Due Sat. Nov. 16, 11:59pm

Load Standard Libraries

In [1]:
```
# Import numpy, pandas, matplotlib.pyplot and seaborn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set matplotlib to display inline
%matplotlib inline
```

Load the Dataset

In [2]:
```
# Import the datasets submodule from sklearn.
from sklearn import datasets

# Load the breast cancer dataset using the load_breast_cancer functio
n.
# Store in the variable 'cancer'.
cancer = datasets.load_breast_cancer()

# Create a new dataframe df with values from cancer.data and with col
umns named using cancer.feature_names.
# Print information about the dataframe using the info function.
df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
mean radius                569 non-null float64
mean texture               569 non-null float64
mean perimeter             569 non-null float64
mean area                  569 non-null float64
mean smoothness            569 non-null float64
mean compactness           569 non-null float64
mean concavity             569 non-null float64
mean concave points        569 non-null float64
mean symmetry              569 non-null float64
mean fractal dimension     569 non-null float64
radius error               569 non-null float64
texture error              569 non-null float64
perimeter error            569 non-null float64
area error                 569 non-null float64
smoothness error           569 non-null float64
compactness error          569 non-null float64
concavity error            569 non-null float64
concave points error       569 non-null float64
symmetry error             569 non-null float64
fractal dimension error    569 non-null float64
worst radius               569 non-null float64
worst texture              569 non-null float64
worst perimeter            569 non-null float64
worst area                 569 non-null float64
worst smoothness           569 non-null float64
worst compactness          569 non-null float64
worst concavity            569 non-null float64
worst concave points       569 non-null float64
worst symmetry             569 non-null float64
worst fractal dimension    569 non-null float64
dtypes: float64(30)
memory usage: 133.5 KB
```

In [3]:
```
# call print(cancer.DESCR) to get a desciption of this dataset.
print(cancer.DESCR)
```

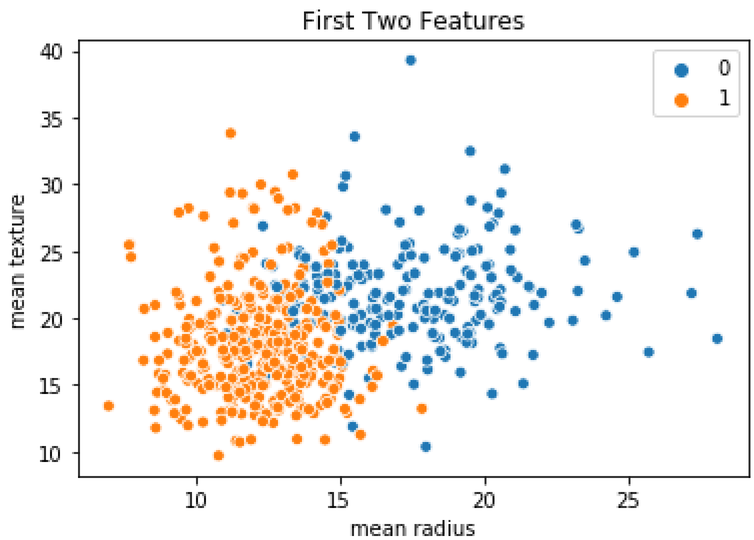
```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the
class

    :Attribute Information:
        - radius (mean of distances from center to points on the peri
meter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the
three
        largest values) of these features were computed for each imag
e,
        resulting in 30 features.  For instance, field 3 is Mean Radi
us, field
        13 is Radius SE, field 23 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign

    :Summary Statistics:

    ===================================== ====== ======
                                           Min    Max
    ===================================== ====== ======
    radius (mean):                        6.981  28.11
    texture (mean):                       9.71   39.28
    perimeter (mean):                     43.79  188.5
    area (mean):                          143.5  2501.0
    smoothness (mean):                    0.053  0.163
    compactness (mean):                   0.019  0.345
    concavity (mean):                     0.0    0.427
    concave points (mean):                0.0    0.201
    symmetry (mean):                      0.106  0.304
    fractal dimension (mean):             0.05   0.097
    radius (standard error):              0.112  2.873
    texture (standard error):             0.36   4.885
    perimeter (standard error):           0.757  21.98
    area (standard error):                6.802  542.2
    smoothness (standard error):          0.002  0.031
```

```
    compactness (standard error):         0.002  0.135
    concavity (standard error):           0.0    0.396
    concave points (standard error):      0.0    0.053
    symmetry (standard error):            0.008  0.079
    fractal dimension (standard error):   0.001  0.03
    radius (worst):                       7.93   36.04
    texture (worst):                      12.02  49.54
    perimeter (worst):                    50.41  251.2
    area (worst):                         185.2  4254.0
    smoothness (worst):                   0.071  0.223
    compactness (worst):                  0.027  1.058
    concavity (worst):                    0.0    1.252
    concave points (worst):               0.0    0.291
    symmetry (worst):                     0.156  0.664
    fractal dimension (worst):            0.055  0.208
    ===================================== ====== ======

    :Missing Attribute Values: None

    :Class Distribution: 212 - Malignant, 357 - Benign

    :Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangas
arian

    :Donor: Nick Street

    :Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) dataset
s.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References
```

```
   - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature 
extraction
     for breast tumor diagnosis. IS&T/SPIE 1993 International Symposi
um on
     Electronic Imaging: Science and Technology, volume 1905, pages 8
61-870,
     San Jose, CA, 1993.
   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer di
agnosis and
     prognosis via linear programming. Operations Research, 43(4), pa
ges 570-577,
     July-August 1995.
   - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learnin
g techniques
     to diagnose breast cancer from fine-needle aspirates. Cancer Let
ters 77 (1994)
     163-171.
```

Plot the First 2 Features From the Dataset

In [4]:
```
# Using seaborn, create a scatterplot with 'mean radius' on the x-axi
s and 'mean texture' on the y-axis.
# Color the points by their class assignment by setting hue as cance
r.target.
sns.scatterplot(x='mean radius',y='mean texture',hue=cancer.target,da
ta=df);

# Using matplotlib.pyplot set the title to 'First Two Features'.
plt.title('First Two Features');
```



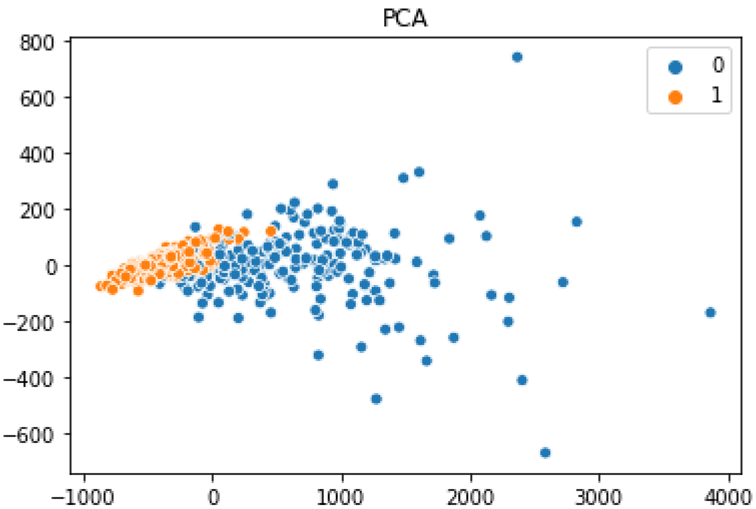
Reduce Data to 2 Dimensions Using PCA

In [5]:
```
# Import PCA from sklearn.
from sklearn.decomposition import PCA

# Create a 2D transformation of the dataframe df using PCA and fit_tr
ansform and store in X_pca.
X_pca = PCA(n_components=2).fit_transform(df)

# Print the shape of X_pca.
# Note: it should have 2 columns.
X_pca.shape
```

Out[5]:
```
(569, 2)
```

Plot the Reduced Representation

In [6]:
```
# Using seaborn, create a scatterplot with the first column of X_pca
 on the x-axis
#    and the second column pf X_pca on the y-axis.
# Color the points by their class assignment by setting hue as cance
r.target.
sns.scatterplot(X_pca[:,0],X_pca[:,1],hue=cancer.target);

# Using matplotlib.pyplot set the title to 'PCA'.
plt.title('PCA');
```



Calculate Feature Ranges

In [7]:
```
# The scale of features in this dataset varies quite a bit, affecting PCA performance.
# To get a sense of the difference, print the range of each feature by subracting df.min() from df.max().
df.max() - df.min()
```

Out[7]:
```
mean radius                   21.129000
mean texture                  29.570000
mean perimeter               144.710000
mean area                   2357.500000
mean smoothness                0.110770
mean compactness               0.326020
mean concavity                 0.426800
mean concave points            0.201200
mean symmetry                  0.198000
mean fractal dimension         0.047480
radius error                   2.761500
texture error                  4.524800
perimeter error               21.223000
area error                   535.398000
smoothness error               0.029417
compactness error              0.133148
concavity error                0.396000
concave points error           0.052790
symmetry error                 0.071068
fractal dimension error        0.028945
worst radius                  28.110000
worst texture                 37.520000
worst perimeter              200.790000
worst area                  4068.800000
worst smoothness               0.151430
worst compactness              1.030710
worst concavity                1.252000
worst concave points           0.291000
worst symmetry                 0.507300
worst fractal dimension        0.152460
dtype: float64
```

Scale the Data

In [8]:
```
#Import StandardScaler from sklearn
from sklearn.preprocessing import StandardScaler

# Using StandardScaler with default settings create a new matrix X_scaled that is a scaled version of df.
X_scaled = StandardScaler().fit_transform(df)

# Print the shape of X_scaled
X_scaled.shape
```

Out[8]:
```
(569, 30)
```

Reduce Scaled Data to 2 Dimensions Using PCA

In [9]:
```
# Reduce X_scaled to 2-D using PCA and fit_transform and store in X_s
caled_pca
X_scaled_pca = PCA(n_components=2).fit_transform(X_scaled)

# Print the shape of X_scaled_pca
X_scaled_pca.shape
```

Out[9]:
```
(569, 2)
```

Plot Reduced Representation of Scaled Data

In [10]:
```
# Using seaborn, create a scatterplot with the first column of X_scal
ed_pca on the x-axis
#    and the second column pf X_scaled_pca on the y-axis.
# Color the points by their class assignment by setting hue as cance
r.target.
sns.scatterplot(X_scaled_pca[:,0],X_scaled_pca[:,1],hue=cancer.target
);

# Using matplotlib.pyplot set the title to 'Scaled PCA'.
plt.title('Scaled PCA');
```