Dimensionality Reduction

Principal Component Analysis

Out of the m independent variables, PCA selects $n < m$ new independent variables that explain the most variance **regardless of the dependent variable.** Since dependent variable is not considered, PCA is an unsupervised model.

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv('datasets/diabetes.csv')

df.head()
```

Out[1]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [2]:

```
df.shape
```

Out[2]:

```
(768, 9)
```

In [3]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [4]:

```
import seaborn as sns

sns.pairplot(df,hue = 'Outcome')
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py:
488: RuntimeWarning: invalid value encountered in true_divide
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
C:\Users\Jesus\Anaconda3\lib\site-packages\statsmodels\nonparametric\kdetool
s.py:34: RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```

Out[4]:

```
<seaborn.axisgrid.PairGrid at 0x1c588ebb240>
```

In [5]:

```
X = df.drop('Outcome',axis = 1)
y = df['Outcome']
```

In [6]:

```
# Split in training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,random_state =
```

In [7]:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train)
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
```

Out[7]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [8]:

```
y_pred = clf.predict(X_test)
```

In [9]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[9]:

```
array([[130,  17],
       [ 38,  46]], dtype=int64)
```

In [10]:

```
clf.score(X_test, y_test)
```

Out[10]:

0.7619047619047619

In [11]:

```
# Split in training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=3,random_state = 4
```

In [12]:

```
X_train.head()
```

Out[12]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
690	8	107	80	0	0	24.6	0.
473	7	136	90	0	0	29.9	0.
204	6	103	72	32	190	37.7	0.
97	1	71	48	18	76	20.4	0.
336	0	117	0	0	0	33.8	0.

In [13]:

```
# Always scale data for good results on PCA
from sklearn.preprocessing import StandardScaler
X_sca = StandardScaler()
X_train = X_sca.fit_transform(X_train)
X_test = X_sca.transform(X_test)
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\preprocessing\data.py:64
5: DataConversionWarning: Data with input dtype int64, float64 were all conv
erted to float64 by StandardScaler.
  return self.partial_fit(X, y)
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\base.py:464: DataConversi
onWarning: Data with input dtype int64, float64 were all converted to float6
4 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
C:\Users\Jesus\Anaconda3\lib\site-packages\ipykernel_launcher.py:5: DataConv
ersionWarning: Data with input dtype int64, float64 were all converted to fl
oat64 by StandardScaler.
  """
```

In [14]:

```
X_train[0:5,:]
```

Out[14]:

```
array([[ 1.23168349, -0.43604825,  0.56160593, -1.28775566, -0.69294238,
        -0.93586599,  1.15656737,  0.06294811],
       [ 0.93510248,  0.47024091,  1.07773825, -1.28775566, -0.69294238,
        -0.26437369, -0.79299869,  1.4242287 ],
       [ 0.63852147, -0.56105365,  0.14870008,  0.71926405,  0.95559722,
         0.72386024, -0.44895762,  1.84962888],
       [-0.84438358, -1.56109686, -1.09001747, -0.15880707, -0.03352654,
        -1.46799195, -0.45197552, -0.95801234],
       [-1.14096459, -0.12353475, -3.56745259, -1.28775566, -0.69294238,
         0.22974327,  1.38592809,  0.91374848]])
```

In [15]:

```
from sklearn.decomposition import PCA
pca = PCA()
pca.fit_transform(X_train)
explained_variance_ratio = pca.explained_variance_ratio_
print(explained_variance_ratio)
```

```
[0.26171944 0.21677862 0.128557   0.10958539 0.09494264 0.08534802
 0.05253743 0.05053145]
```

In [16]:

```
# Extract top 2 principal components
pca = PCA(n_components=2)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
print(explained_variance)
```

```
[0.26171944 0.21677862]
```

In [17]:

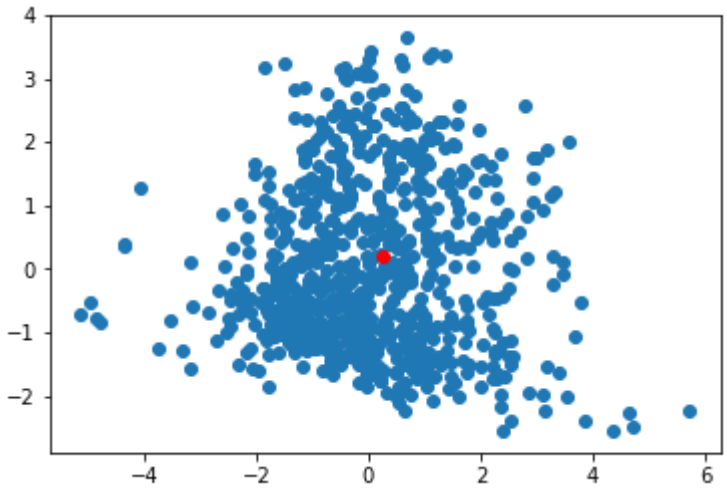
```
X_train
```

Out[17]:

```
array([[-0.78397591,  1.34563612],
       [-0.23714761,  2.44484824],
       [ 1.21249591,  0.96614483],
       ...,
       [ 1.85090101,  0.80663508],
       [-1.74219774, -0.89044531],
       [-1.38214951, -0.25137068]])
```

In [18]:

```
plt.scatter(X_train[:,0],X_train[:,1])
plt.scatter(explained_variance[0],explained_variance[1],c = 'r')
plt.show()
```



In [19]:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train)
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
```

Out[19]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

In [20]:

```
y_pred = clf.predict(X_test)
```

In [21]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[21]:

```
array([[3]], dtype=int64)
```

In [22]:

```
from sklearn.metrics import accuracy_score

accuracy_score(y_test, y_pred)
```

Out[22]:

1.0

In [23]:

```
clf.score(X_test, y_test)
```

Out[23]:

1.0

In [24]:

```
# Split in training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=3,random_state = 7
```

In [25]:

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis(n_components=2)
X_train = lda.fit_transform(X_train, y_train)
X_test = lda.transform(X_test)

```

In [26]:

```
X_train
```

```
[ 8.93942725e-01],
[ 3.91899748e-01],
[-4.06773438e-01],
[-6.04159603e-03],
[ 8.65074171e-02],
[-8.62645714e-01],
[ 3.73737798e-01],
[-1.16915125e+00],
[ 5.38354761e-02],
[ 4.39553771e-02],
[ 1.73063543e+00],
[-1.23637148e+00],
[ 1.72140836e+00],
[-9.21589191e-01],
[-9.85956059e-01],
[ 1.25719080e+00],
[ 1.18119464e-01],
[-1.18018629e+00],
[-1.42486210e-01],
[ 7.17493164e-01].
```

In [27]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
clf = LogisticRegression(random_state=0)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
cm
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:
433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Speci
fy a solver to silence this warning.
  FutureWarning)
```

Out[27]:

```
array([[1, 0],
       [0, 2]], dtype=int64)
```

In [28]:

```
clf.score(X_test, y_test)
```

Out[28]:

```
1.0
```