DEMO AND WORKSHOPS

PCA Workshop using Python

- You will need to following libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from factor_analyzer import FactorAnalyzer
from sklearn.preprocessing import StandardScaler
```

PCA Coding 1

```
# Get a summary of the dataset:
df.describe()

# plot a histogram
df.hist()

# Generate a correlation matrix
df.corr()

# Bartlett's test
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
Bartlett = calculate_bartlett_sphericity(df)

# KMO test
from factor_analyzer.factor_analyzer import calculate_kmo
kmo = calculate_kmo(df)
```

PCA Coding 2

```
# Standardize the data:
StandardScaler().fit_transform(df)

# Run (fit) the PCA procedure:
pca = PCA(n_components).fit(df)

# Get the Component Loading Matrix:
Loadings = np.sqrt(eigenvalues)*eigenvectors

# Get the Eigenvectors:
eigenvectors = pca.components_

# Get the Eigenvalues:
eigenvalues = pca.explained_variance_

# Generate the component scores (transform df using fitted PCA model)
PC_scores = pca.transform(df)

# other output: (1) Loading Plot (2) Scree Plot (3) Communalities
```

Try out you PCA code

1. Run your PCA with the dataset from the example in the course notes: *LoanApplicant.csv*
2. Check that your code is working fine.
3. For you hands-on practice, run PCA on the `house_data.csv`
4. Finally, do the following workshop for submission:
    1. Run your PCA on the `diabetes.csv` dataset
    2. Decide on how many components you want to keep
    3. Then use these components as input to your
        - logistic regression model
        - KNN
        - Naïve Bayes
    4. Observe the results and compare it with the non-pca results
    5. Document your observations as comments in the code file.
    6. Name your ipynb file as follows: *yourname.ipynb*
    7. Submit this ONE FILE ONLY to the luminous.