Data science

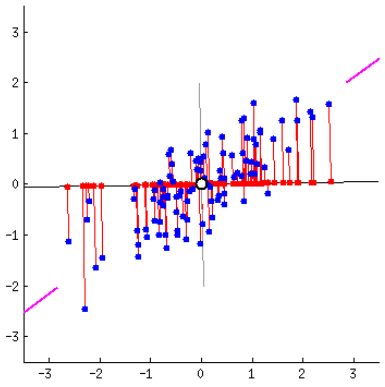
"The best way to learn data science is to apply data science."

Activity 2

Bochra CHEMAM

About PCA :

In summary, we can define **principal component analysis (PCA)** as the transformation of any high number of variables into a smaller number of uncorrelated variables called principal components (PCs), developed to capture as much of the data's variance as possible.



- We can use PCA to reduce the number of variables, avoid multicollinearity, or have too many predictors relative to the number of observations.
- PCA is a linear combination of the *p* features, and taking these linear combinations of the measurements is essential to reduce the number of plots necessary for visual analysis while retaining most of the information present in the data

Steps involved in PCA

- Standardize the PCA.
- Calculate the covariance matrix.
- Find the eigenvalues and eigenvectors for the covariance matrix.
- Plot the vectors on the scaled data.

Data science

"The best way to learn data science is to apply data science."

For this Activity session we will:

work through simple data set to visualize PCA And learn how we interpret it

1. **Problem definition: For this activity, we will investigate the (breast_cancer) dataset.**

2. **Import libraries: We will import the important python libraries required for this algorithm**

   - **import matplotlib.pyplot as plt**
   - **import pandas as pd**
   - **import numpy as np**
   - **import seaborn as sns**
   - **%matplotlib inline**

3. **Import the dataset from the python library sci-kit-learn.**

   - **from sklearn.datasets import load_breast_cancer**
   - **cancer = load_breast_cancer()**

The dataset is in a form of a dictionary. So we will check what all key values are there in the dataset.

- **cancer.keys()**

4. **Now, let's make the Dataframe for the given data and check its head value.**

5. **Analyze your dataset: use Describe method, shape, dtypes,**

6. **Use .corr() to calculate correlation Matrix between different variables**

7. **we need to scale the data such that each feature has unit variance and has not a greater impact than the other one.**

   from sklearn.preprocessing import StandardScaler

   scaler = StandardScaler()

```
scaler.fit(df)

scaled_data = scaler.transform(df)
```

8. **Let's check whether the normalized data has a mean of zero and a standard deviation of one.**

```
np.mean(scaled_data),np.std(scaled_data)
```

9. **PCA with Scikit Learn uses a very similar process to other preprocessing functions that come with SciKit Learn. We instantiate a PCA object, find the principal components using the fit method, then apply the rotation and dimensionality reduction by calling transform().**

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

pca.fit(scaled_data)
```

10. **Now we can transform this data into its first 2 principal components.**

```
x_pca = pca.transform(scaled_data)
```

11. **Now let us check the shape of data before and after PCA using .shape**
12. **We've reduced 30 dimensions to just 2! Let's plot these two dimensions out!**
13. **Plot a scater :**

```
plt.figure(figsize=(8,6))

plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')

plt.xlabel('First principal component')

plt.ylabel('Second Principal Component')
```

Interpret results