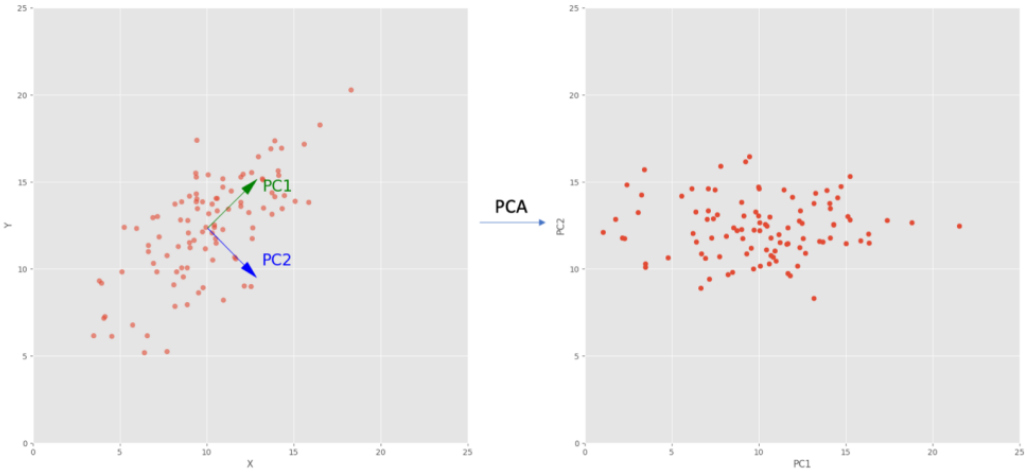
Aim :

Implementation of Principal Component Analysis (PCA)

Theory:

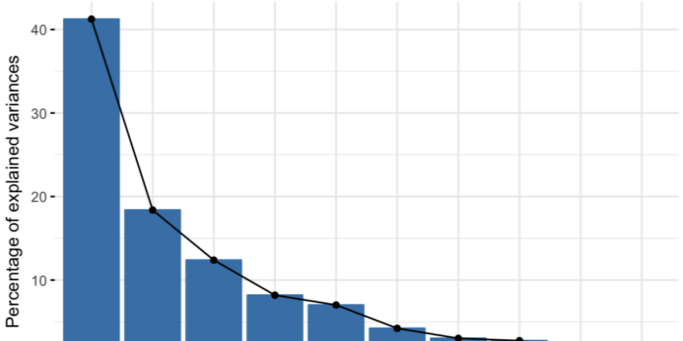
What is PCA?

- Principal Component Analysis or PCA is a *dimensionality reduction technique* for data sets with many features or dimensions.
- PCA allows us to compress our data and reduce the number of dimensions while retaining a lot of the information.
- Instead of expressing our data in terms of arbitrary dimensions like x,y , and z (or the features in our dataset), we can express it in terms of the principal components. These principal components are the orthogonal (or perpendicular) directions in which our data varies. The process of PCA is the process of finding these principal components and using them to transform our data. It's just a method of summarizing some data.



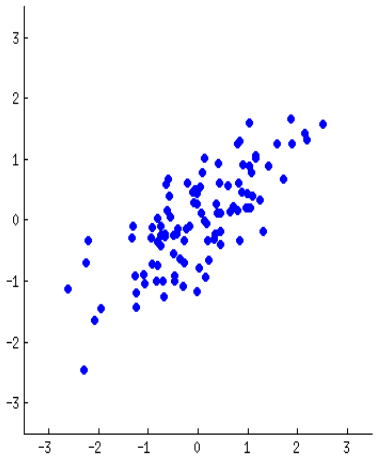
The above images are two dimensional PCA examples. When we don't eliminate components we simply rotate our coordinates to match the orthogonal directions of variance

The first principal component is always the direction that captures the most variance in the data. The following components capture the next highest variance and so on, under the constraint that all components are always orthogonal. The idea is 10-dimensional data gives you 10 principal

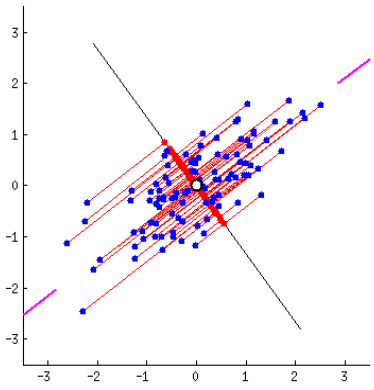


components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.It constructs some new characteristics that turn out to summarize our dataset. Of course, these new characteristics are constructed using the old ones.

You see that the two properties (x and y on this figure) are correlated. A new property can be constructed by drawing a line through the center of this cloud and projecting all points onto this line. This new property will be given by a linear combination$w1x$+$w2y$, where each line corresponds to some particular values of$w1$and$w2$.



Here is what these projections look like for different lines (red dots are projections of the blue dots):



First, the variation of values along this line should be maximal. We can see that the "spread" (we call it "variance") of the red dots changes while the line rotates,we can see when it reaches maximum.
Second, if we reconstruct the original two characteristics (position of a blue dot) from the new one (position of a red dot), the reconstruction error will be given by the length of the connecting red line. We have to observe how the length of these red lines changes while the line rotates, we can see that the total length reaches minimum.

In the given graph "the maximum variance" and "the minimum error" are reached at the same time, This line corresponds to the new property that will be constructed by PCA.

The spread of the red dots is measured as the average squared distance from the center of the cloud to each red dot, it is known as the Variance.

Steps Involved in the PCA

1. Standardize the dataset.
2. Calculate the covariance matrix for the features in the dataset.
3. Calculate the eigenvalues and eigenvectors for the covariance matrix.
4. Sort eigenvalues and their corresponding eigenvectors.
5. Pick k eigenvalues and form a matrix of eigenvectors.
6. Transform the original matrix.

1. Standardize data

- Standardization is the process of translating and scaling our features so that they are all distributed around a mean of zero with a standard deviation of one.
- We want to standardize our data so that the covariances are easily comparable for each pair of features. If we don't do it, features with larger ranges of numbers will have higher covariances. This part is not strictly necessary but can be very helpful and is required for some models.
- Normalization is one of the most frequently used data preparation techniques, which helps us to change the values of numeric columns in the dataset to use a common scale.

Formula :

$$z = \frac{value - mean}{standard\ deviation}$$

$$z = \frac{x - \mu}{\sigma}$$

2. Construct covariance matrix

- Building the covariance matrix is the actual first step of PCA.
- Covariance tells us how the two features vary with respect to one another.
- In this step, we will be building a square matrix p*p where each row represents a feature and each column also represents a feature. Each entry represents the covariance between the row feature and the column feature at that position.
- In order to talk about covariance, we will first look at variance. Variance tells us about the spread of data, just like standard deviation does.
- Covariance tells us how the two features vary with respect to one another.
- If a given sample has a value above the mean (which we've set to zero through standardization) for both features then we get a positive result as the product.

  The same is true if the values for both features are below the mean.

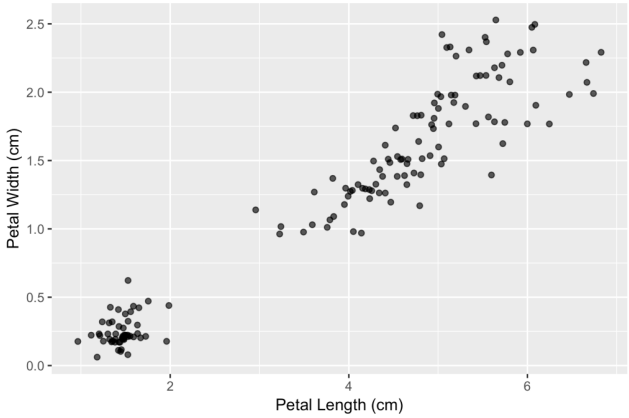
  Conversely, if one value is below and one is above we will get a negative value.

- If we take the example of the given graph,then we can observe tendencies in our data.

  If the covariance for a given pair of features is positive, both of these features tend to increase together.

  If the covariance is negative, one feature tends to increase as the other decreases.

  If the covariance is zero then the two features are unrelated



3. Compute the Eigenvectors and Eigenvalues of Covariance Matrix to Identify the Principal Components

- Eigenvectors and eigenvalues always come in pairs, so that every eigenvector has an eigenvalue. And their number is equal to the number of dimensions of the data.

For example, for a 3-dimensional data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

- The eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance(most information) and that is Principal Components.
- And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.
- By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.
- After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues.

Implementation:

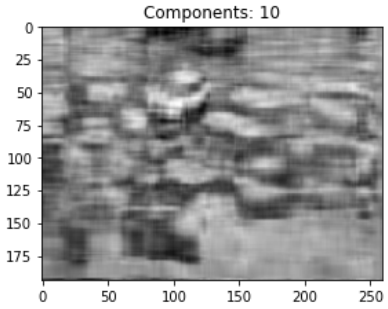
Python Program

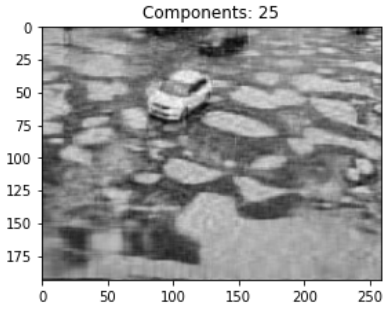
```
import numpy as np
import matplotlib.pyplot as plt
#define the dataset
X=[[0.1,0.2,0.5],
[0.2,0.1,0.3],
[0.3,0.3,0.3],
[0.4,0.4,0.3],
[0.7,0.8,0.4],
[0.8,0.6,0.6]]

#Step-1- We standardize data to bring all values in the range (-1,1), we don't need to divide here by std. deviation
print("Mean of each column is")
print(np.mean(X, axis=0))

X_standard = (X - np.mean(X , axis = 0))
print("Standardized data=",X_standard)

#Step-2
cov_mat = np.cov(X_standard , rowvar = False)
print("Covariance Matrix=",cov_mat)
#Step-3
eigen_values , eigen_vectors = np.linalg.eig(cov_mat)

#Step-4
sorted_index = np.argsort(eigen_values)[::-1]
```

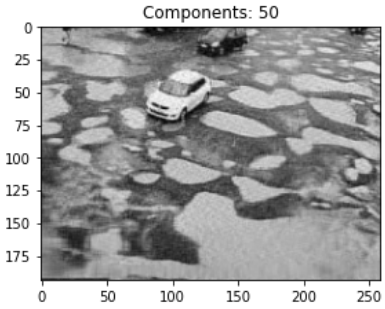
```
sorted_eigenvalue = eigen_values[sorted_index]
print("Eigenvalues sorted")
print(sorted_eigenvalue)
sorted_eigenvectors = eigen_vectors[:,sorted_index]
print("Eigenvectors sorted")
print(sorted_eigenvectors)

#Step-5
eigenvector_subset = sorted_eigenvectors[:,0:2]
print("Top 2 Eigenvectors")
print(eigenvector_subset)

#Step-6
X_reduced = np.dot(X_standard,eigenvector_subset)
print("New transformed data with two columns pc1 and pc2")
print(X_reduced)
#Plot the reduced data
for pc1, pc2, lab, col in zip(X_reduced.transpose()[0],X_reduced.transpose()[1],('A', 'B',
'C', 'D', 'E', 'F'),
('blue', 'red', 'green', 'orange', 'purple', 'brown')):
plt.scatter(pc1,pc2,c=col)
plt.annotate(lab,(pc1,pc2))
plt.xlabel("Principal Component 1")

plt.ylabel("Principal Component 2")
plt.show()
```

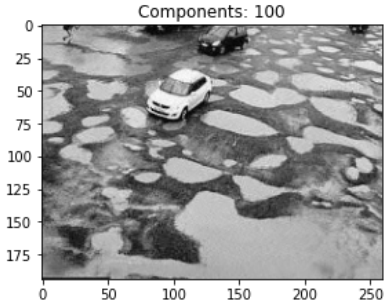
Results:

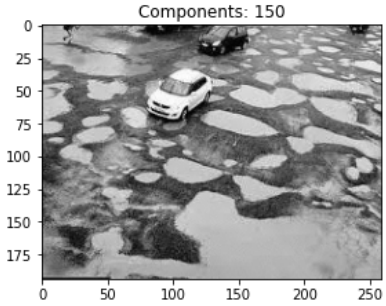
- PCA on images



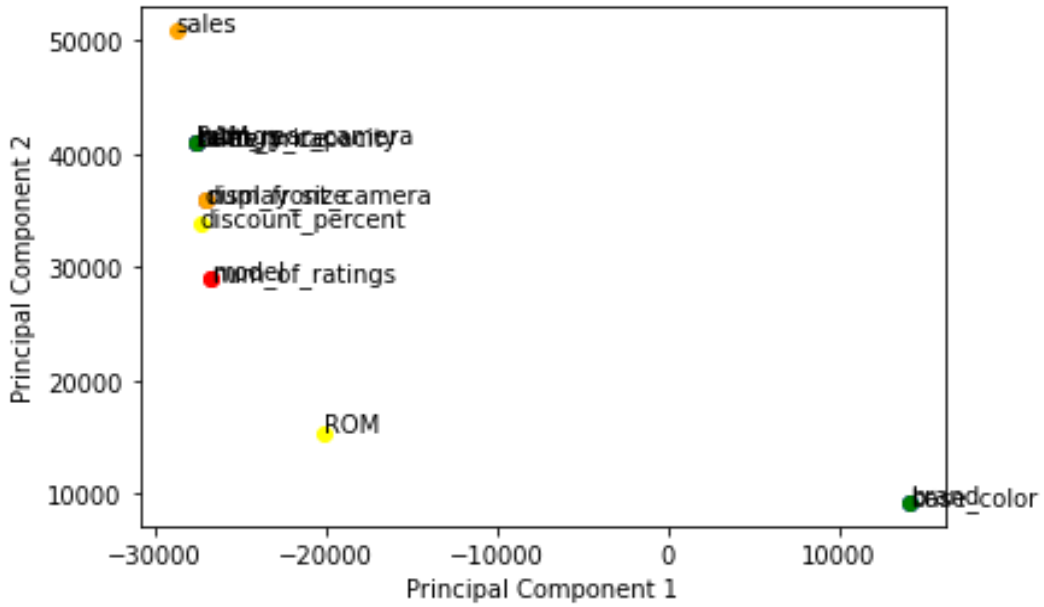




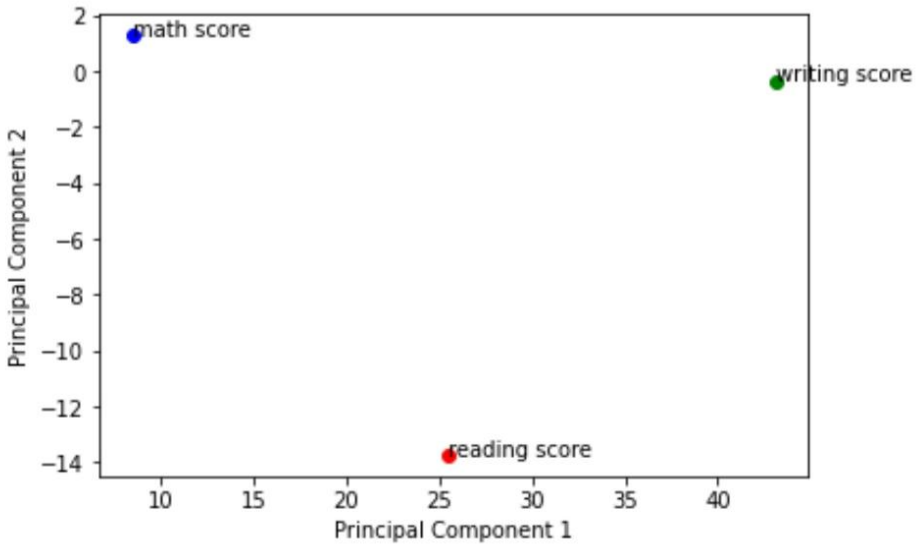




- PCA on flipkart dataset



- PCA on student marks dataset



Conclusion:

In this practical we learned about Eigenvectors and Eigenvalues. Also implemented Principal Component Analysis (PCA).