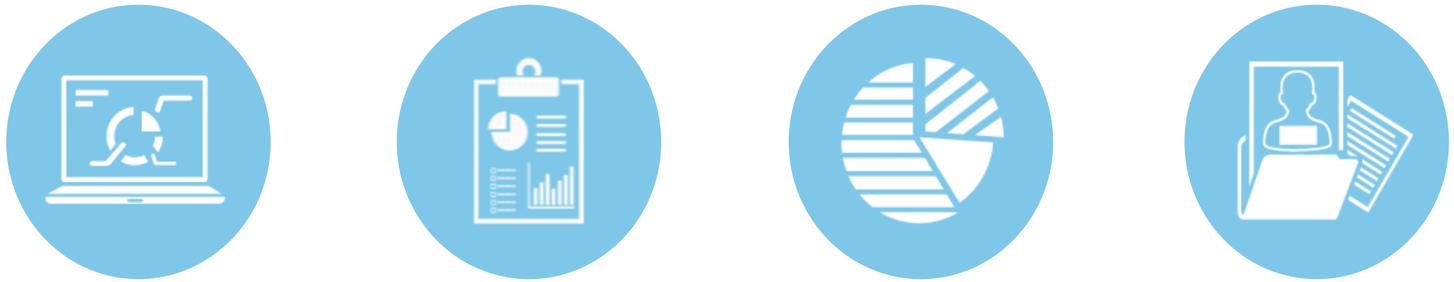
# Machine Learning

Lesson 4: Feature Engineering



# Concepts Covered

- Factor Analysis
- PCA
- LDA
- PCA, LDA in Python

# Learning Objectives

By the end of this lesson, you will be able to:

- Demonstrate feature engineering and its significance using python
- Practice different feature selection techniques

# Feature Engineering

## Topic 1: Feature Selection

# Problem Statement

Given a multidimension dataset, extract information based on interrelations among variables.

| Temperature | Time | Weight | Location | Weekday | Weekend | Sales |
|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | .. |
| ... | ... | ... | ... | ... | ... | .. |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |

# Probable Solutions

Techniques to extract feature-based information

| Temperature | Time | Weight | Location | Weekday | Weekend | Sales |
|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | .. |
| ... | ... | ... | ... | ... | ... | .. |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |

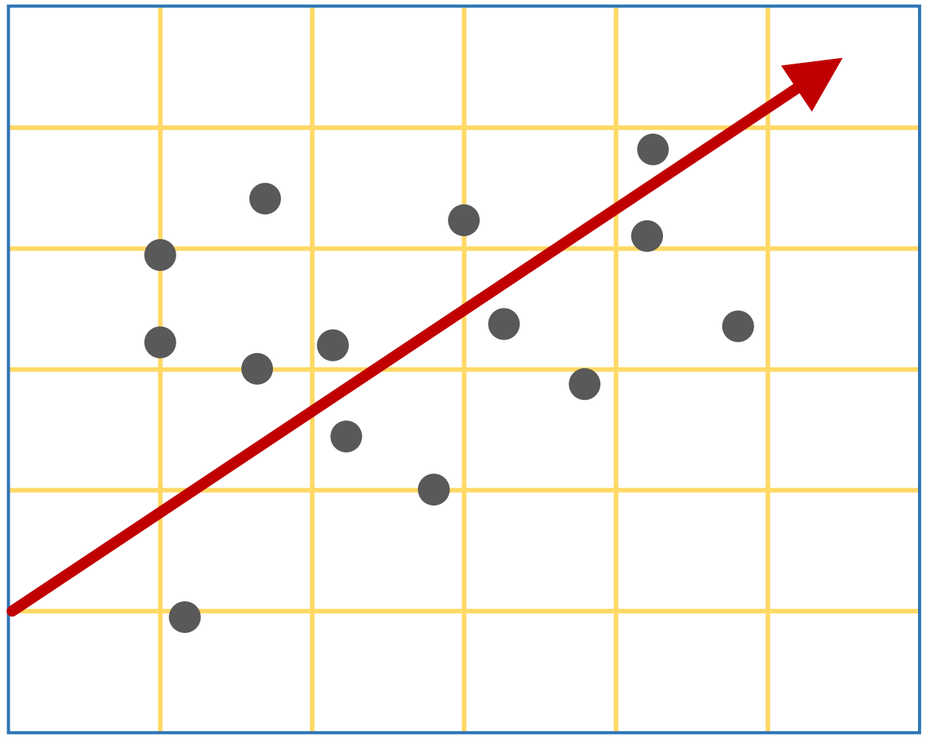
Regression

Factor Analysis

# Regression

**Regression**

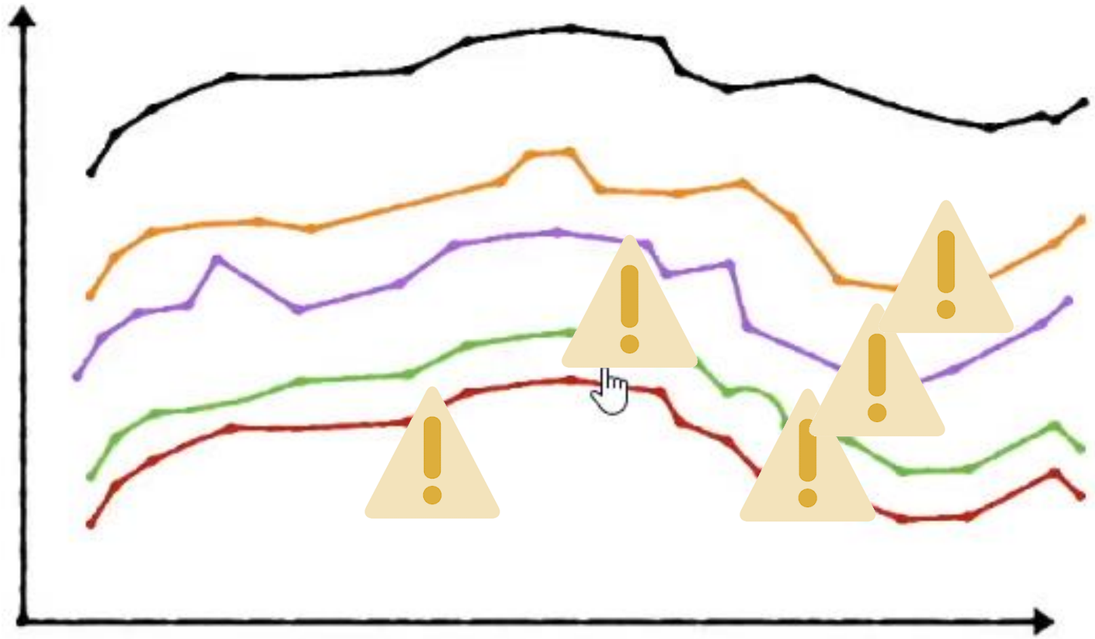
Factor Analysis



Regression tells the relationship among variables and quantifies the relationship using set of equations

# Multicollinearity Problem

Interdependence among two or more explanatory variables may lead to an unreliable model.

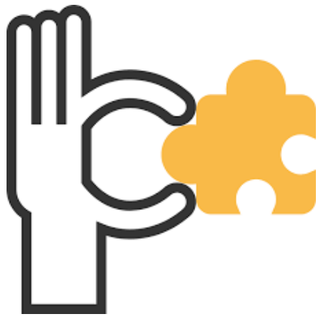


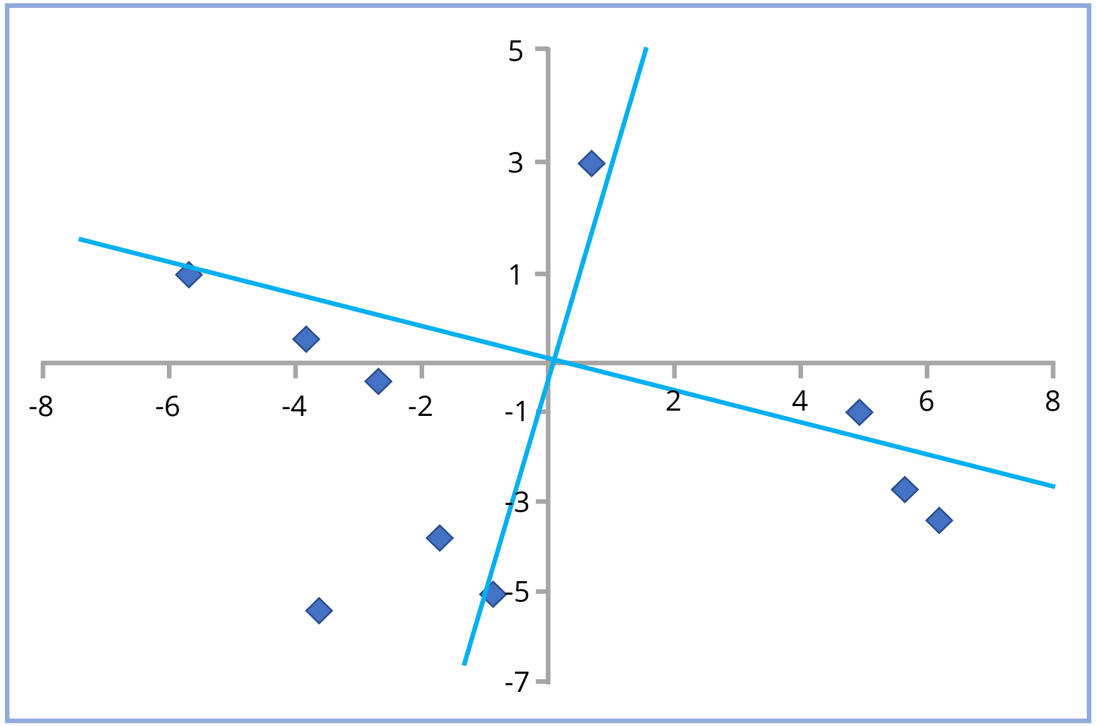
**Solution** : Perform *Factor Analysis* to extract underlying causes leading to this behaviour

# Factor Analysis

Regression

**Factor Analysis**

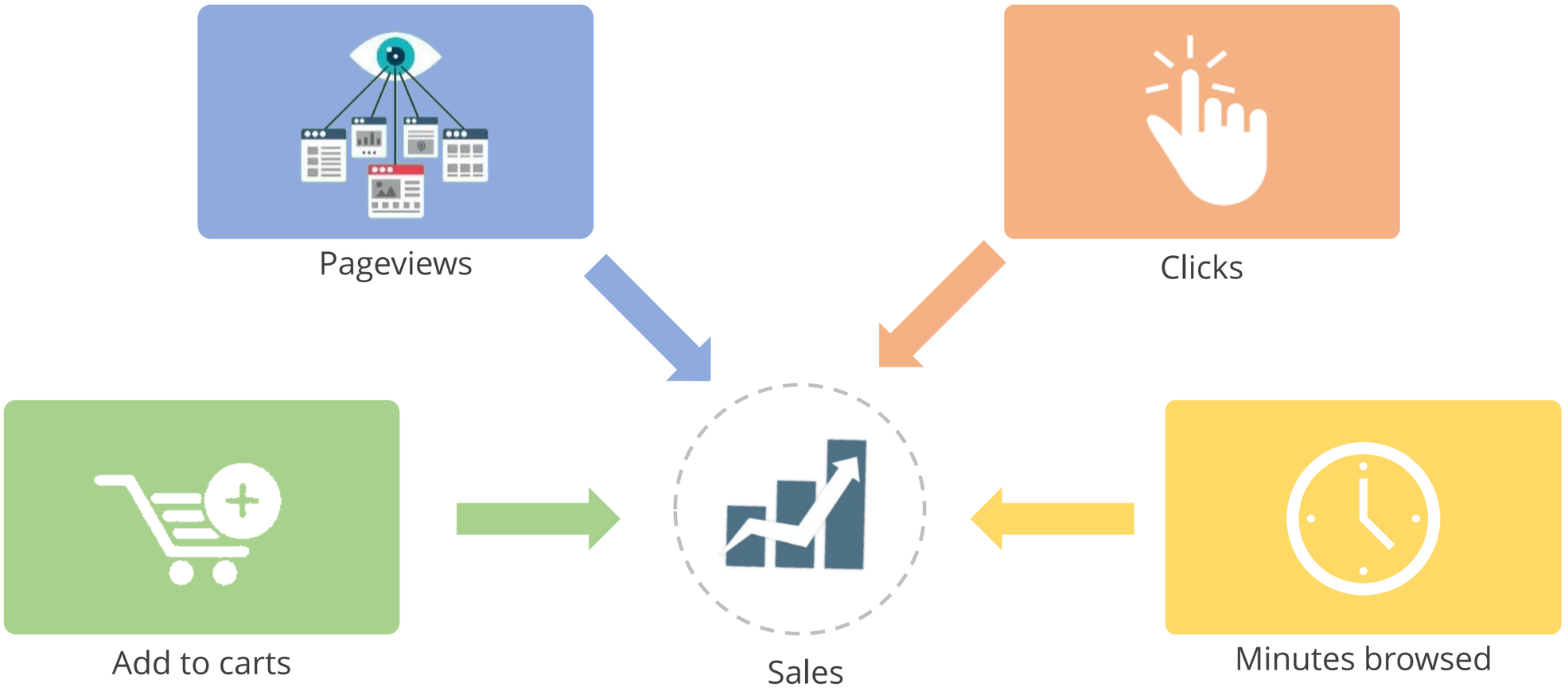





Common factors of the observations explain the variable interdependence

# Example

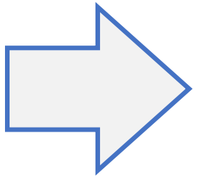
The relationship between observable variables and observable outcome: sales



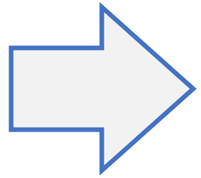
# Example (Contd.)

There may be few underlying causes which can affect the output apart from the observed ones:

- Pageviews
- Clicks
- Add to carts
- Minutes Browsed
- Sessions



- Selection
- Marketing spend
- Pricing



- Sales

*Observed causes*

*Underlying causes*

*Effect*

**Note**: Identifying underlying causes helps in accurate sales prediction

# Feature Engineering

## Topic 2: Factor Analysis

# Factor Analysis Process

There may be few underlying causes which can affect the output, apart from the observed ones:

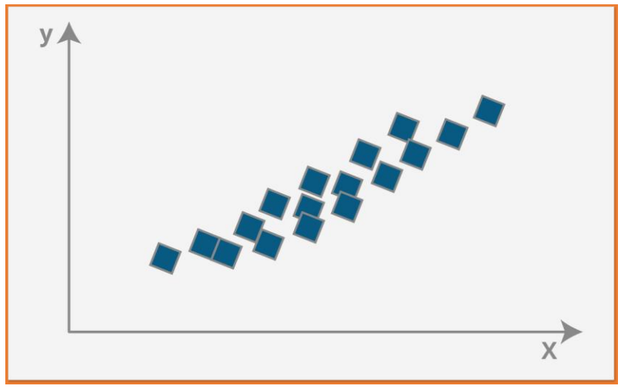
## Principal Component Analysis (PCA)

- Extracts hidden factors from the dataset
- Defines your data using less number of components, explaining the variance in your data
- Reduces the computational complexity
- Determines whether a new data point is part of the group of data points from your training set

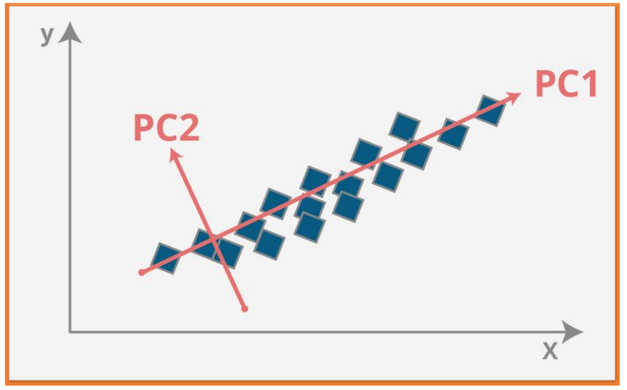
## Linear Discriminant Analysis (LDA)

- Reduces Dimensions
- Searches for a linear combination of variables that best separates 2 classes
- Reduces the degree of overfitting
- Determines how to classify a new observation out of a group of classes

# Principal Component Analysis




You have data on the x and y axis

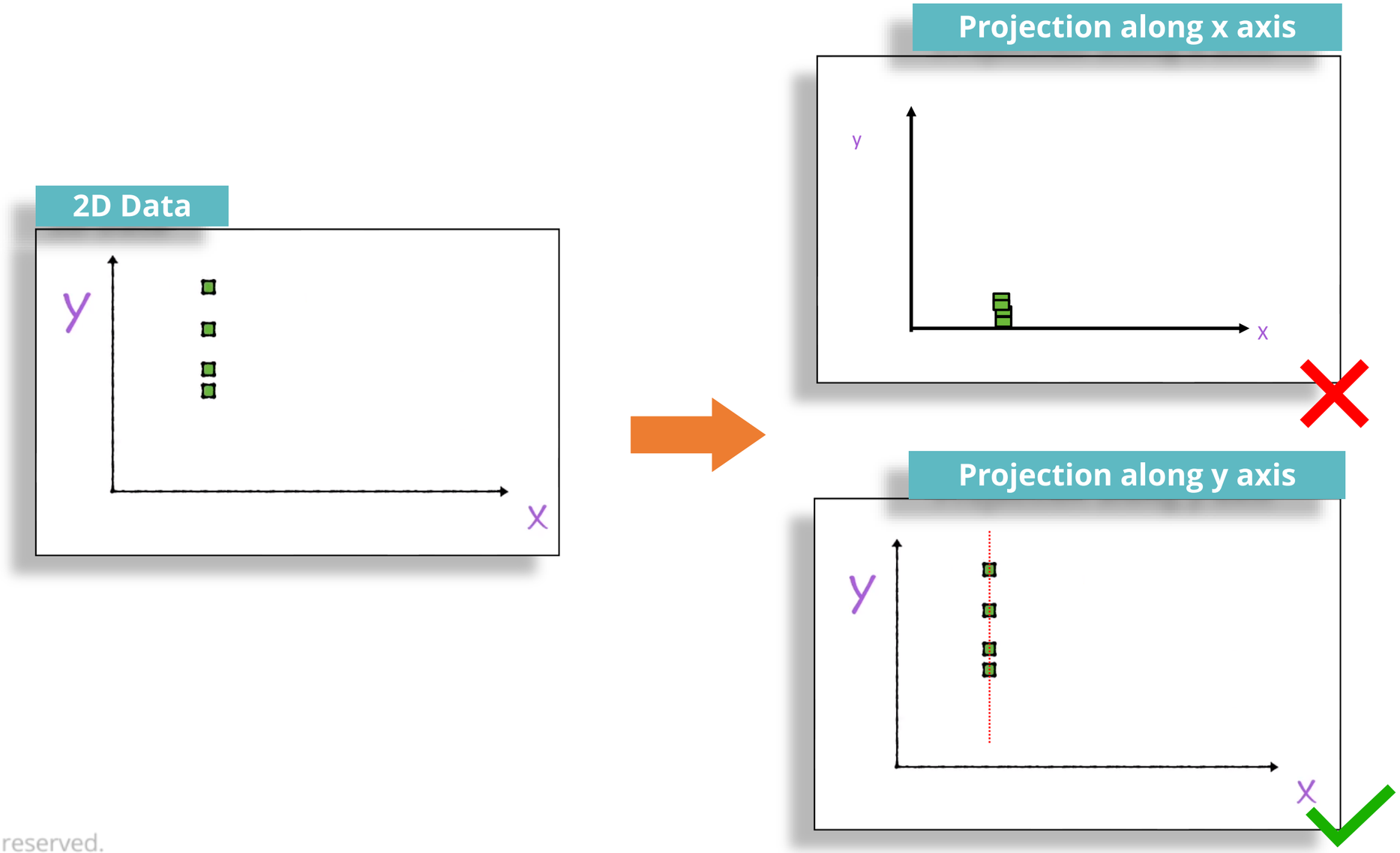



*Applying PCA*: New set of axes are achieved, denoted as PC1 and PC2


Data around PC2 projected along PC1 to ensure no variance is lost

# Direction of Maximum Variance

New set of axes are found to be orthogonal to each other



# Finding PC1

The first principal component is the direction of maximum variance and is obtained by solving eigen vectors

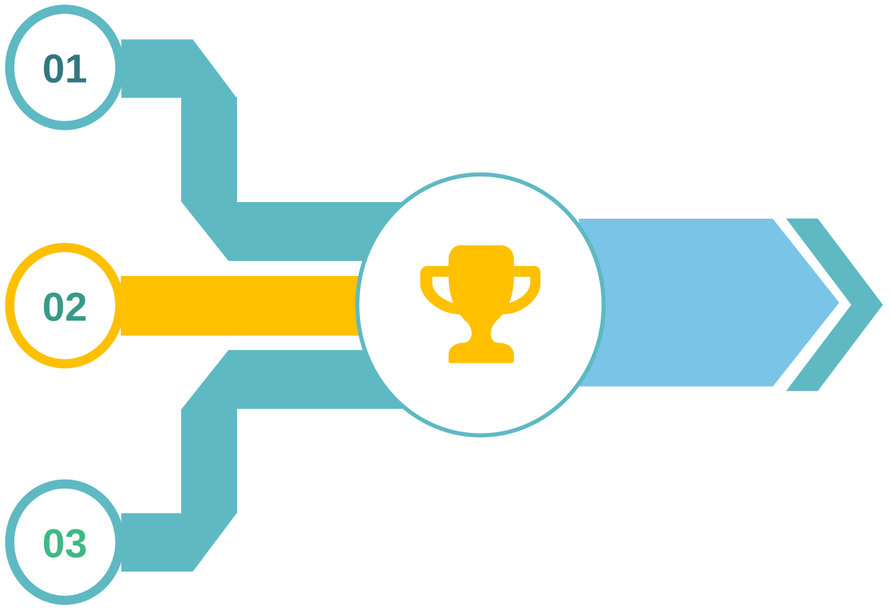
PC1: Mathematically,

$a_1x_1 + a_2x_2 + a_3x_3.. + a_kx_k$

Constraint:

$a_1{}^2 + a_2{}^2 + .. + a_k{}^2 = 1$

Eigen Decomposition is used to solve the above equation




**Note**: Solution of Eigen Decomposition is out of scope of this course. For more details visit: https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix

# Results of Eigen Decomposition

Solution of eigen vectors results into eigen values

## Eigen Vector

Values from $a_1$ to $a_k$ form Eigen Vector

$V_1 = [a_1, a_2, a_3, .., a_K]$

One eigen vector corresponds to each principal component

## Principal Component

- Derived by multiplying the original data matrix with the eigen vector $V_1$: $F_1 = a_1x_1 + a_2x_2 + .. + a_k$

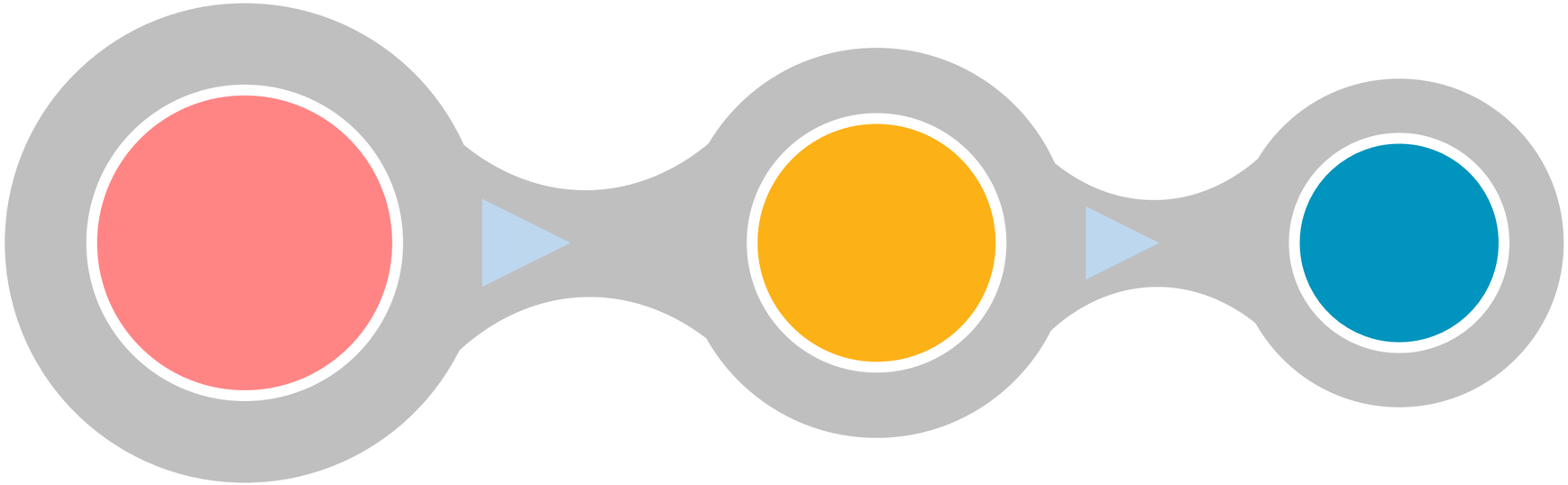
## Eigen Value

Variance of $F_1$:

e = variance($F_1$)

# Eigen Values and PCA

Eigen values are the variances of principal components arranged in descending order



$F_1, F_2, F_3, ...., F_k$

$\text{Var}(F_1) > \text{Var}(F_2) > \text{Var}(F_3) > \text{Var}(F_k)$

Eigenvalue 1 > Eigenvalue 2 > Eigenvalue 3 > Eigenvalue k

# Assisted Practice

## Factor Analysis

Duration: 15 mins.

**Problem Statement:** The dataset you are going to use in this practice is the famous Iris data set. The dataset consists of 150 records of Iris plant with four features: “sepal-length”, “sepal-width”, “petal-length”, and “petal-width”. All the features are numeric. The records have been classified into one of the three classes, that is, “Iris-setosa”, “Iris-versicolor”, or “Iris-verginica”.

**Objective:**

- Train the models on original number of features
- Reduce the number of variables by merging correlated variables
- Extract the most important features from the dataset that are responsible for maximum variance in the output.

**Access:** Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

# Unassisted Practice

## Factor Analysis

Duration: 15 mins.

**Problem Statement:** Scikit learn comes with pre-loaded datasets, load the digits dataset from that collection (**http://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html**). Using Scikit learn perform a PCA transformation such that the transformed dataset can explain 95% of the variance in the original dataset. Find out the number of components in the projected subspace.

**Objective:** Understand and practice principal component analysis using Scikit learn.

**Note:** This practice is not graded. It is only intended for you to apply the knowledge you have gained to solve real-world problems.

**Access:** Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

# Data Import and Split

Below is the code for importing and splitting the dataset:

Code

```
import sklearn
from sklearn.datasets import  load_digits
digits = load_digits()
X = digits.data
y = digits.target

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
print(X_train.shape)
```

# PCA Transformation

Transforming the train and test sets such that they explain 95% of variance

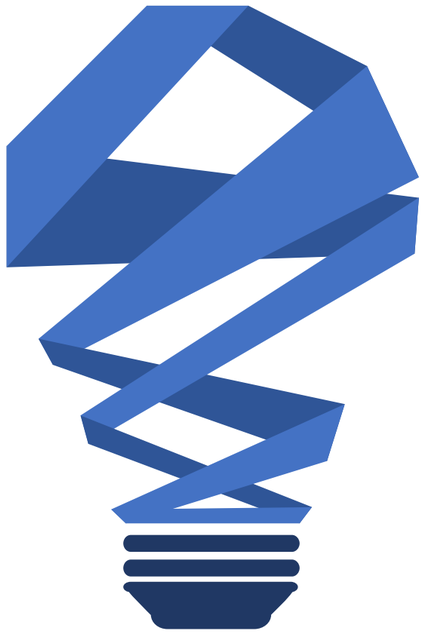
Code

```
from sklearn.decomposition import PCA
sklearn_pca = PCA(n_components=0.95)
sklearn_pca.fit(X_train)
X_train_transformed = sklearn_pca.transform(X_train)

print(X_train_transformed.shape)
print(X_test.shape)
X_test_transformed =sklearn_pca.transform(X_test)
print(X_test_transformed.shape)
```

```
(1437, 28)
(360, 64)
(360, 28)
```

# Linear Discriminant Analysis (LDA)



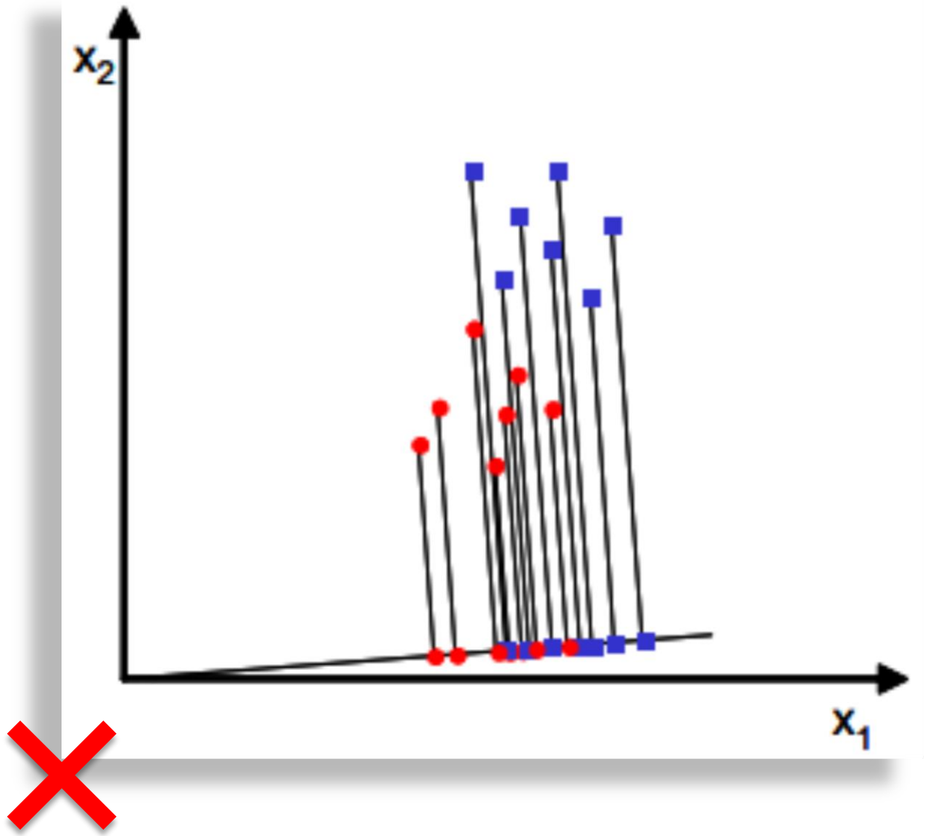
Assume a set of D - dimensional samples {X(1, X(2, ..., X(N}, N1 of which belong to class ω1 and N2 to class ω2

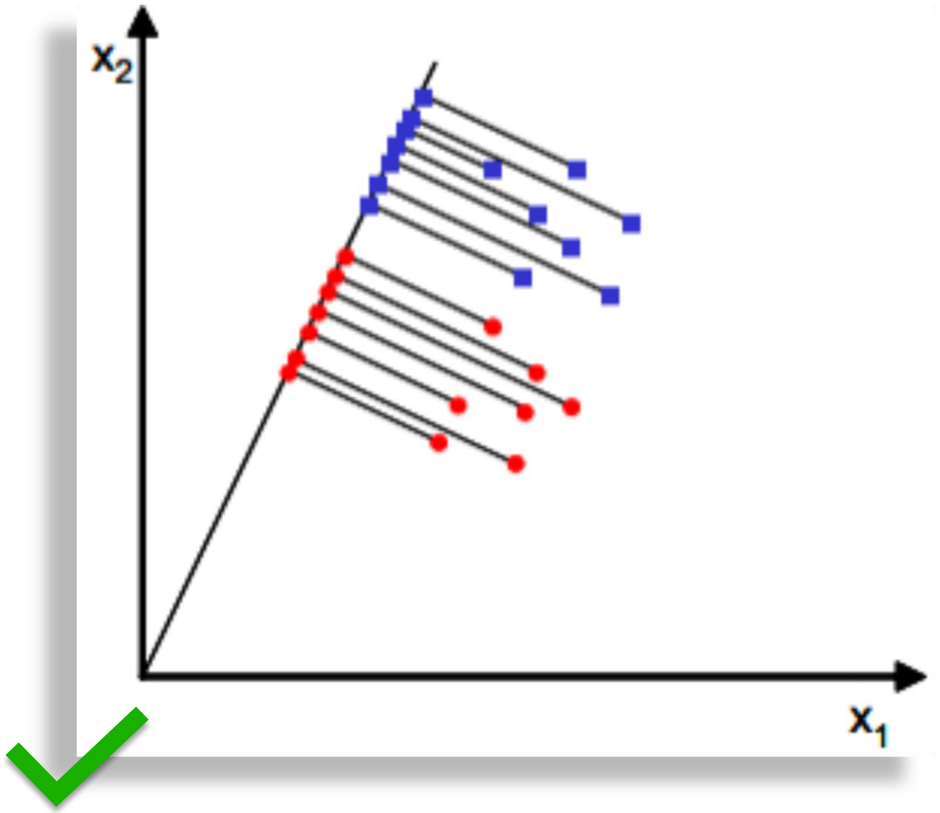
Obtain a scalar y by projecting the samples x onto a line: Y = $W^T$X

Of all the possible lines, select the one that maximizes the separability of the scalars

# Maximum Separable Line

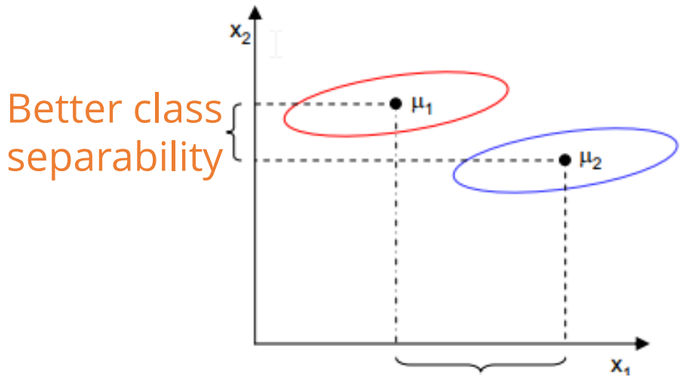
The maximum separable line finds out the feature subspace such that class separability is also optimized



# Finding Maximum Separable Line

- Measure of separation
- Linear Discriminant
- Optimum projection
- Finding the maxima




- Mean vector within each class of x and y is:

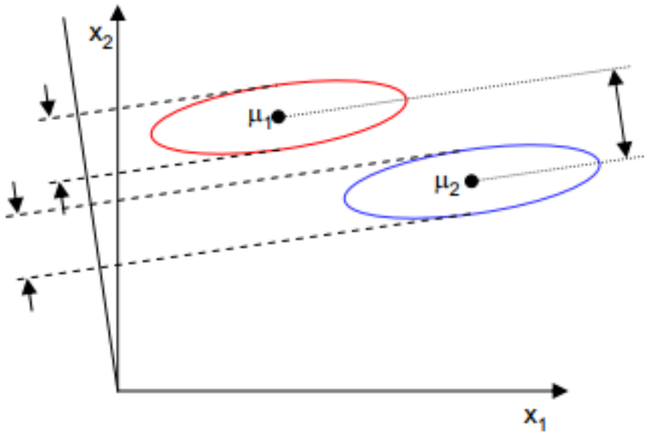
$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y = \frac{1}{N_i}\sum_{x\in\omega_i} w^T x = w^T \mu_i$$

- Objective function is the distance between the projected means:

$$J(w) = \left|\tilde{\mu}_1 - \tilde{\mu}_2\right| = \left|w^T(\mu_1 - \mu_2)\right|$$

# Finding Maximum Separable Line (Contd.)

- Measure of separation
- Linear Discriminant
- Optimum projection
- Obtain the maxima

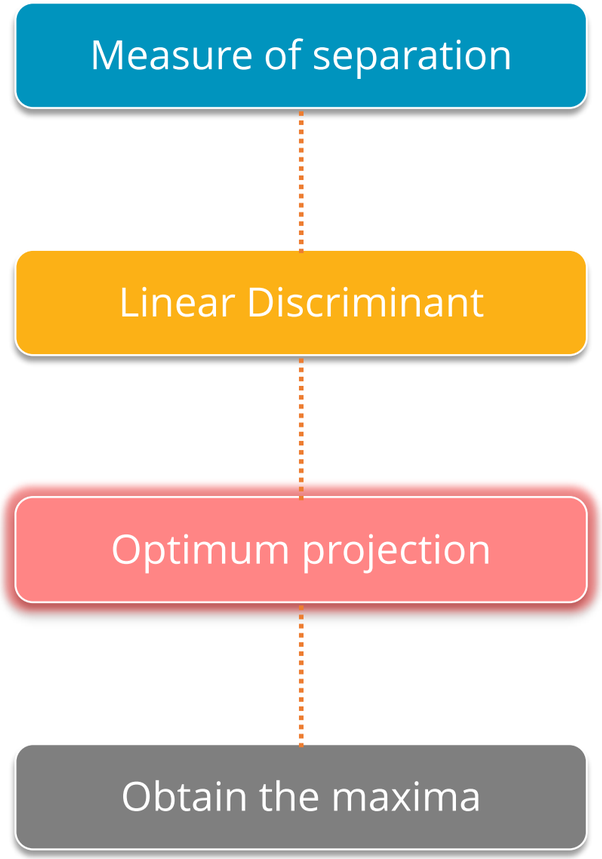



- Function of difference between the means normalized by measure of scatter(an equivalent variance):

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

***Variables from same class are projected very close to each other and at the same time, the projected means are as farther apart as possible***

# Finding Maximum Separable Line (Contd.)




- A measure of the scatter in multivariate feature space x:

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_1 + S_2 = S_W$$ , $S_w$ is the within class scatter matrix

- Scatter of the projection y can be expressed as a function of the scatter matrix in feature space x

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 = \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w = w^T S_i w$$

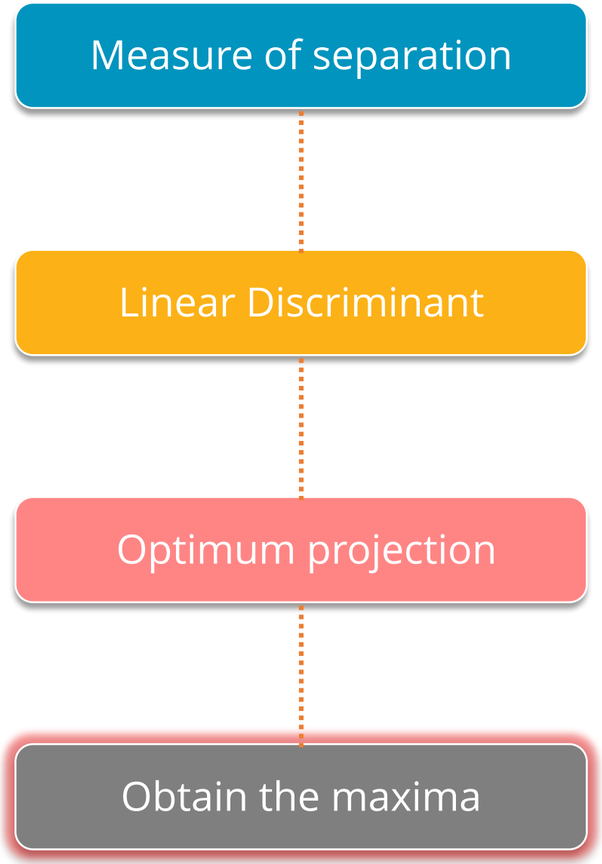
$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w$$

$S_w$ is the within class scatter

- Difference between the projected means

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w$$

# Finding Maximum Separable Line (Contd.)




- Express the linear discriminant in terms of $S_w$ and $S_B$:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

- Find the maxima of J(w) by differentiating and equating to zero

- Solving the above differentiation yields:

$$w^* = \underset{w}{\operatorname{argmax}} \left\{ \frac{w^T S_B w}{w^T S_W w} \right\} = S_W^{-1}(\mu_1 - \mu_2)$$

***Above is the optimal direction for projection***

# Assisted Practice

## Factor Analysis

Duration: 10 min.

**Problem Statement:** Consider the iris dataset preloaded within the mlxtend library. Segregate the data and labels accordingly and transform the data to two linear discriminants.

**Objective:** Perform LDA as a dimensionality reduction technique.

**Access:** Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

# Unassisted Practice

## Factor Analysis

Duration: 15 min.

**Problem Statement:** Scikit learn comes with pre-loaded datasets, load the digits dataset from that collection (**http://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html**). Using Scikit learn perform LDA on the dataset. Find out the number of components in the projected subspace. Transform the dataset and fit a logistic regression and observe the accuracy. Compare it with the previous model based on PCA in terms of accuracy and model complexity.

**Objective:** Understand and practice LDA using Scikit learn.

**Note:** This practice is not graded. It is only intended for you to apply the knowledge you have gained to solve real-world problems.

**Access: :** Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

# Import and Split

Import the load_digits dataset from scikit learn and split it accordingly

Code

```
import sklearn
from sklearn.datasets import  load_digits
digits = load_digits()
X = digits.data
y = digits.target
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
print(X_train.shape)
```

```
(1437, 64)
```

# Number of Components in Transformed Shape

Fit LDA on the training data

Code

```
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as lda
sklearn_lda = lda()
sklearn_lda.fit(X_train,y_train)
X_train = sklearn_lda.transform(X_train)
print("Number of components in transformed shape {}".format(X_train.shape[1]))
```

```
Number of components in transformed shape 9
```

# Dataset Transformation

Transforming the test data

Code

```
#sklearn_lda.transform(X_test)

X_test =sklearn_lda.transform(X_test)
print(X_test.shape)
```

```
(360, 9)
```

# Fitting a Logistic Regression Model

Code

```
lr = LogisticRegression(penalty='l1')
lr.fit(X_train,y_train)
y_predict = lr.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_predict,y_test)
print(accuracy)
```

```
0.963888888889
```

# Key Takeaways

Now, you are able to:

- Demonstrate feature engineering and its significance using python
- Practice different feature selection techniques

# Knowledge Check

| Knowledge Check<br><br>1 | **Dimensionality reduction algorithms are one of the possible ways to reduce the computation time required to build a model.** |
|---|---|

a. True

b. False

| Knowledge Check 1 | Dimensionality reduction algorithms are one of the possible ways to reduce the computation time required to build a model. |
|---|---|

a. True

b. False

The correct answer is **a. True**

**By reducing the dimensions of data, it will take less time to train a model.**

**Knowledge Check**

**2**

## Which of the following is/are true about PCA?

1. PCA is an unsupervised method
2. It searches for the directions with the data having the largest variance
3. Maximum number of principal components <= number of features
4. All principal components are orthogonal to each other

a. 1 and 2

b. 1, 2, and 4

c. 1, 2, and 3

d. All the above

| Knowledge Check 2 | **Which of the following is/are true about PCA?**<br>1. PCA is an unsupervised method<br>2. It searches for the directions with the data having the largest variance<br>3. Maximum number of principal components <= number of features<br>4. All principal components are orthogonal to each other |
|---|---|

a. **1 and 2**

b. **1, 2, and 4**

c. **1, 2, and 3**

d. All the above

The correct answer is **d. All the above**

**All the above options are true.**

# Lesson-End Project

Duration: 20 mins.

**Problem Statement:** John Cancer Hospital (JCH) is a leading cancer hospital in USA. It specializes in preventing breast cancer.  Over the last few years, JCH has collected breast cancer data from patients who came for screening/treatment. However, this data has almost 30 attributes and is difficult to run and interpret the result. You as an ML expert, has to reduce the no. of attributes (Dimensionality Reduction) so that results are meaningful and accurate.

**Objective:** Reduce the number of attributes/features in data to make the analysis of the results comprehensible to doctors.

**Access:** Click the Labs tab in the left side panel of the LMS. Copy or note the username and password that are generated. Click the Launch Lab button. On the page that appears, enter the username and password in the respective fields and click Login.

# Thank You