# Project Plan

1. Investigation of the problem and available solutions… References
2. Writing the Python code to implement the solution

**Linear discriminant analysis** (**LDA**) is a generalization of **Fisher's linear discriminant**, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. [ https://en.wikipedia.org/wiki/Linear\_discriminant\_analysis ] = [1]

LDA is closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data [1].LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made.

# Statistical classification

In machine learning and statistics, **classification** is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Classification is an example of pattern recognition.

In the terminology of machine learning,[*Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9.* [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number)*978-0-262-01243-0.]* classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or *features*. These properties may variously be ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

An algorithm that implements classification, especially in a concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

The ***Iris* flower data set** or **Fisher's *Iris* data set** is a multivariate data set introduced by the British statistician and [biologist](https://en.wikipedia.org/wiki/Biologist) Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis [1] https://en.wikipedia.org/wiki/Iris\_flower\_data\_set

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor) [2] R. A. Fisher (1936). "The use of multiple measurements in taxonomic problems". [Annals of Eugenics](https://en.wikipedia.org/wiki/Annals_of_Eugenics). 7 (2): 179–188. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1111/j.1469-1809.1936.tb02137.x](https://doi.org/10.1111%2Fj.1469-1809.1936.tb02137.x)

Four features were measured from each sample: the length and the width of the [sepals](https://en.wikipedia.org/wiki/Sepal) and [petals](https://en.wikipedia.org/wiki/Petal), in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

Based on Fisher's linear discriminant model, this data set became a typical test case for many statistical classification techniques in machine learning such as support vector machines[3] ["UCI Machine Learning Repository: Iris Data Set"](https://archive.ics.uci.edu/ml/datasets/iris). archive.ics.uci.edu. Retrieved 2017-12-01.

The use of this data set in cluster analysis however is not common, since the data set only contains two clusters with rather obvious separation. One of the clusters contains Iris setosa, while the other cluster contains both Iris virginica and Iris versicolor and is not separable without the species information Fisher used. This makes the data set a good example to explain the difference between supervised and unsupervised techniques in data mining: Fisher's linear discriminant model can only be obtained when the object species are known

**Picture of the 3 iris Flower Species**

|  |  |  |
| --- | --- | --- |
| **Iris Virginica** | **Iris Versicolor** | **Iris Setosa** |

The Iris dataset employed in this project is the same used in R.A. Fisher's classic 1936 paper, “The Use of Multiple Measurements in Taxonomic Problems”, and can also be found on the UCI Machine Learning Repository. [https://www.kaggle.com/danalexandru/simple-analysis-of-iris-dataset/data]

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. R. A Fisher observed that one flower species is linearly separable from the other two samples, but the other two are not linearly separable from each other. This observation will be validated in this project, together with other statistical insights, using Python3.

The columns in the dataset shown in Appendix A are:

• Sepal Length (cm)

• Sepal Width (cm)

• Petal Length (cm)

• Petal Width (cm

• Species Class:

-- Iris Virginica

-- Iris Versicolour

-- Iris Setosa

Sepal is a part of the flowering plants. Usually green, sepals typically function as protection for the flower in bud, and often as support for the petals when in bloom.

Petals are modified leaves that surround the reproductive parts of flowers. They are often brightly colored or unusually shaped to attract pollinators. Together, all of the petals of a flower are called a corolla. As mentioned above, Petals are usually accompanied by another set of special leaves called sepals. Collectively, Sepals and Petals are called calyx. [ https://en.wikipedia.org/wiki/Petal ]

P1--- Sample input/Output(Appendix 1)

# Francis Adepoju. March 31 - May 15 2018

# End of Module Project

# Investigating the Iris\_flower\_data\_set

# http://archive.ics.uci.edu/ml/machine-learning-databases/iris/

"""

#

# A Python script that reads in the Iris data set stored in the data subdirectory under

# the current project and then prints the four numerical values on

# each row in a nice format on the screen. The printed data are for:

# petal length, petal width, sepal length and sepal width.

# These values are listed with decimal places aligned, and

# with a space between the columns.

"""

# Open the input iris data file using the with keyword

with open("data/iris.csv") as inputFile:

# Loop through the file and split the data, summing up the data in the process

samplesTotal = 0

for myData in inputFile: #

# Data are forced to be seen as type float

col1 = float(myData.split(',')[0])

col2 = float(myData.split(',')[1])

col3 = float(myData.split(',')[2])

col4 = float(myData.split(',')[3])

col5 = myData.split(',')[4] # Not used in this program...

samplesTotal = samplesTotal + 1

# Using the python f-Strings to print

print(f'{col1} {col2} {col3} {col4}')

print("Total Samples = " + str(samplesTotal))

# End of processing...

print("End of file processing")

Data samples and traits

Data analysis - distributions

Data analysis - correlations

Naturally, we find:

* a **high positive correlation** between PetalWidth and PetalLength (**0.96**)
* a **high positive correlation** between PetalLength and SepalLength (**0.87**)
* a **high positive correlation** between PetalWidth and SepalLength (**0.81**)

As such, we observe correlations between these main attributes: **PetalWidth**, **PetalLength** and **SepalLength**.

### **Theory**

PCC is:

* 1 is total positive linear correlation
* 0 is no linear correlation
* −1 is total negative linear correlation

Data analysis – clusterization

Scatterplots for the correlating pairs

import matplotlib.pyplot as plt

# Open the input iris data file using the with keyword

myList1 = []

myList2 = []

myList3 = []

myList4 = []

with open("data/iris.csv") as f:

# Loop through the file and split the data, summing up the data in the process

samplesTotal = 0

for myData in f: #

# Data are forced to be seen as type float

col1 = float(myData.split(',')[0])

col2 = float(myData.split(',')[1])

col3 = float(myData.split(',')[2])

col4 = float(myData.split(',')[3])

col5 = myData.split(',')[4] # Not used in this program...

samplesTotal = samplesTotal + 1

myList1.append(col1)

myList2.append(col2)

myList3.append(col3)

myList4.append(col4)

# Data are formatted and printed on the screen as required.

# print('{0:.1f} {1:.1f} {2:.1f} {3:.1f}'.format(col1, col2, col3, col4))

# Using the python f-Strings to print same

#print(f'{col1} {col2} {col3} {col4}')

#print(myList1)

#print(myList2)

#print(myList3)

#print(myList4)

plt.title('Relationship Between Sepalwidth & Sepal whatever')

plt.xlabel('Sepal Height(cm)')

plt.ylabel('Sepal Width(cm)')

# plt.scatter(X, Y, s=60, c='red', marker='^')

# plt.scatter(myList1, myList2, myList3, myList4)

# plt.scatter(myList1, c='red', marker='\*', myList2, c='blue', myList3, c='green', myList4, c='black')

plt.scatter(myList1, myList2, myList3, myList4, marker='\*')

plt.show()

#print("Total Samples = " + str(samplesTotal))

#

print("End of file processing") # End of processing.

