Edit this markdown field to set your home directory

Supervised Learning

Reiterating what we learned last session, supervised learning is the family of algorithms that given a training set are able to find a predictive model.

This predictive model assigns a reference value, y, to a certain feature vector, x.

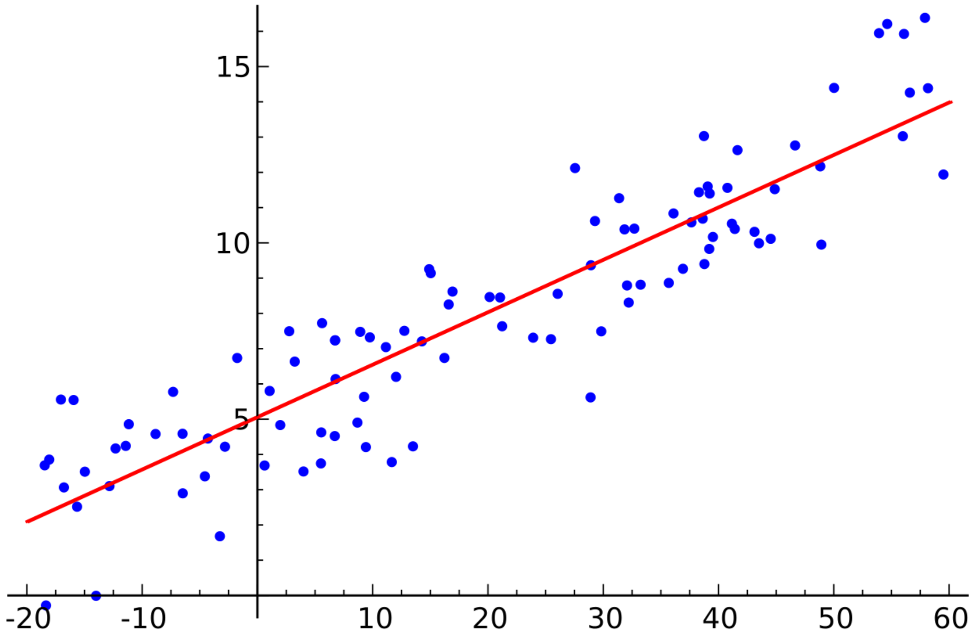
$y = f(x)$

The supervised learning algorithms can be further divided into the class of classification and regression problems.

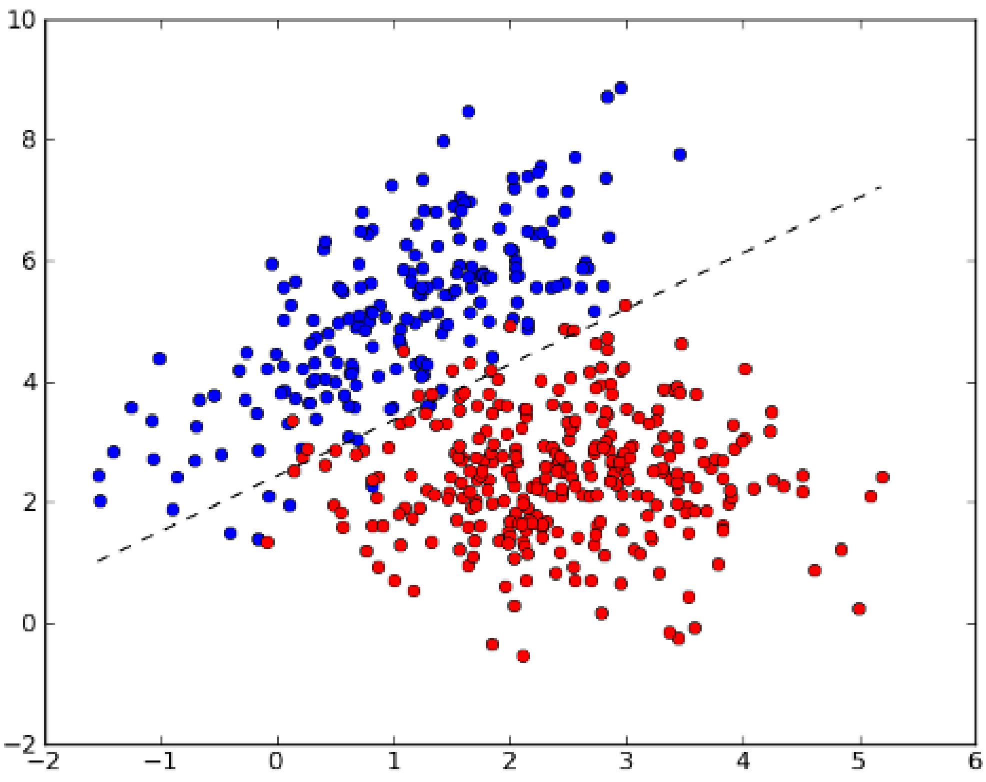
Regression vs Classification

The distinction between regression and classification problems is concerned with the reference value, y.

In regression problems this reference value belongs to a continuous domain. For example, linear regression is concerned with finding a line that best explains the data. This line can be considered a predictive model, given a certain x it predicts a value for y.



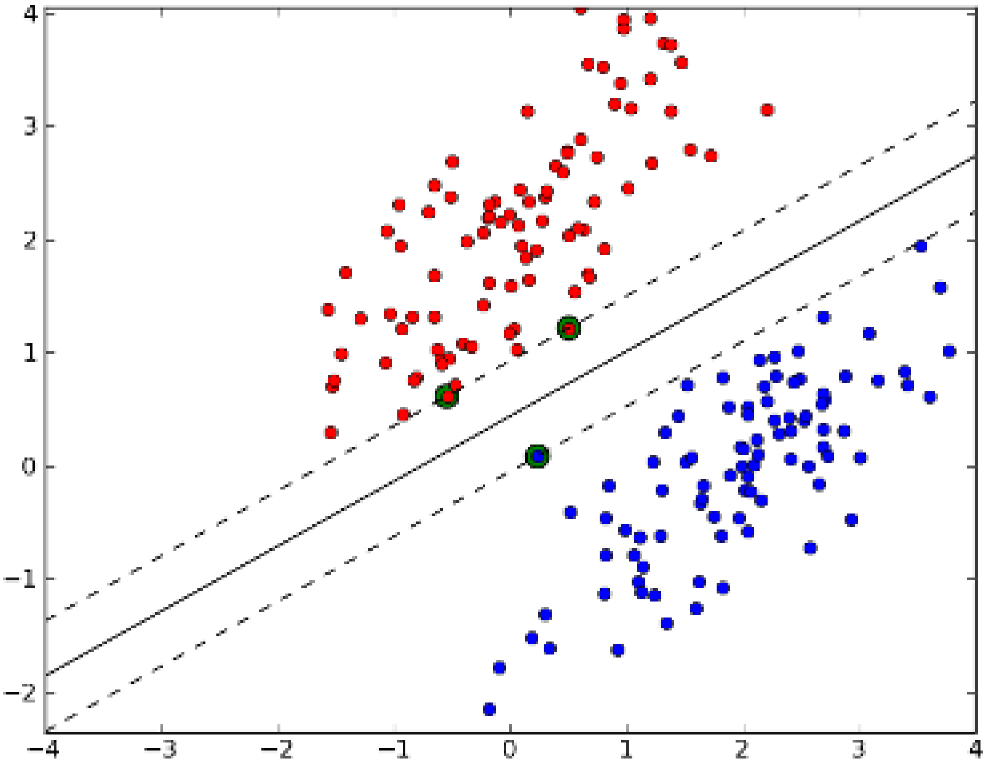
In classification problems the reference value is discrete. For example, given a picture classifying what object is represented. The output will follow a discrete number of possible values, it is either a ship or a horse not a ship-horse.



In this workshop we will look at two classifier algorithms and in the next one we will look into regression models. The first algorithm we will look at is the Support Vector Machine model.

Support Vector Machines

Support vector machines try to divide the space of training data with a hyperplane in such a way that it maximizes the distance between two classes.



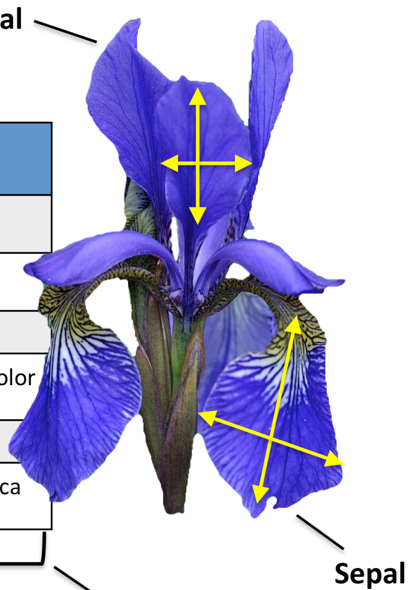
Now that we understand the general idea lets see how we can implement this in Python!

Time for some code!

For this example we will use the Iris dataset. This data sets consists of 3 different types of irises' (Setosa, Versicolour, and Virginica) petal and sepal length, stored in a 150x4 numpy.ndarray.

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

Features
(attributes, measurements, dimensions)

We will start by loading and visualizing the dataset.

We will load the dataset using a handy library called Pandas. Loading the Iris dataset in this way it makes visualization and use much more flexible!

[3]

```python
import pandas as pd

iris_dataframe =
pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/ir
is.csv")
iris_dataframe
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-

Now we can use the Seaborn library to visualize the dataset.

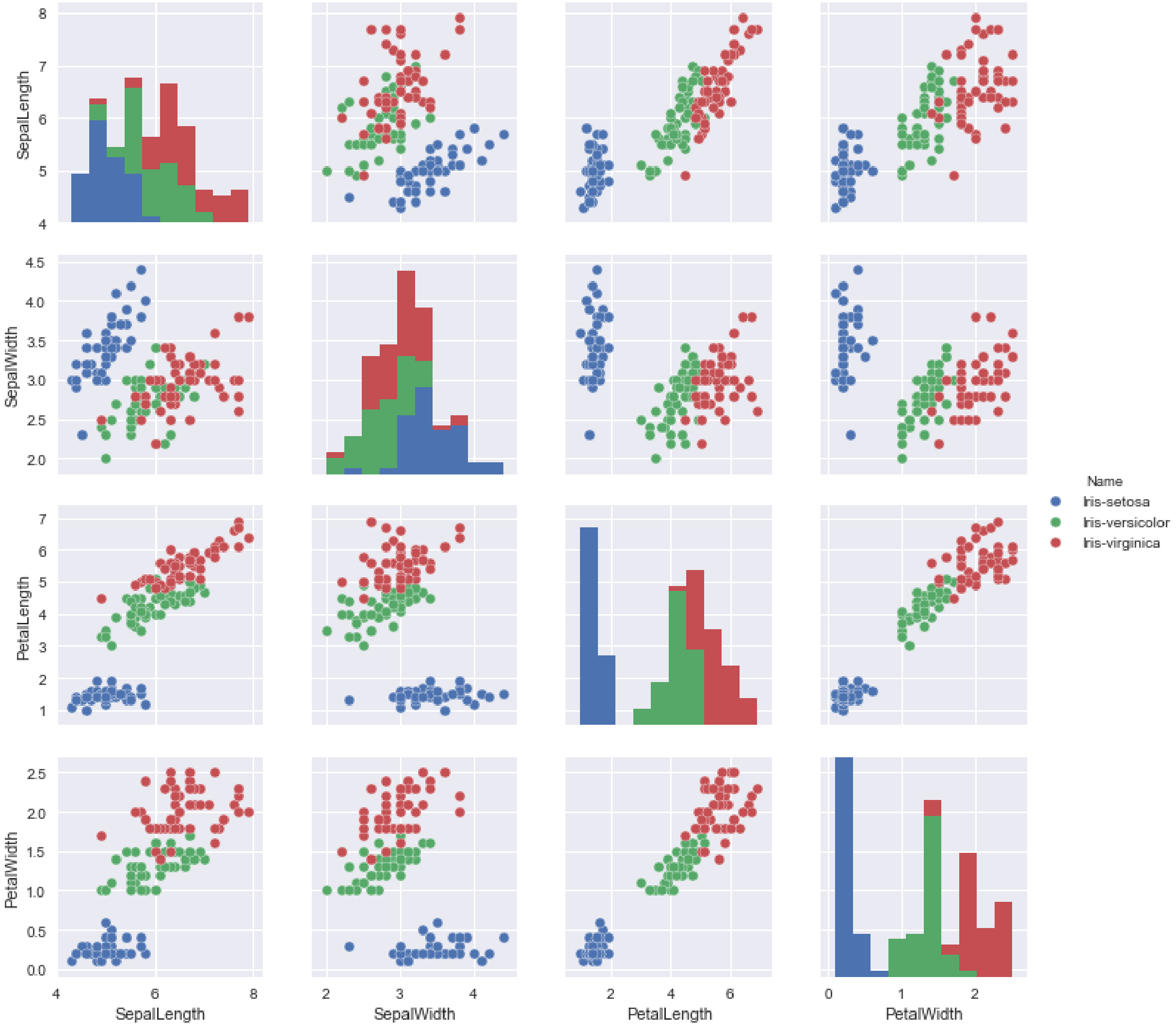
[2]

```python
%matplotlib inline

import seaborn as sns

sns.pairplot(iris_dataframe, hue="Name")
```

<seaborn.axisgrid.PairGrid at 0xb72fb70>



[ ]

[ ]