ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI
SCIENZE STATISTICHE "PAOLO FORTUNATI"

# 2 - Supervised and Unsupervised Models

Giovanni Della Lunga

Halloween Conference in Quantitative Finance

Bologna - October 26-28, 2021

# Outline

1. Supervised Learning
   - What is Supervised Learning?
   - Example: Predicting House Prices

2. Unsupervised Learning
   - What is Unsupervised Learning?
   - Example: k-Means Clustering

3. End Notes
   - Breaking the Dichotomy

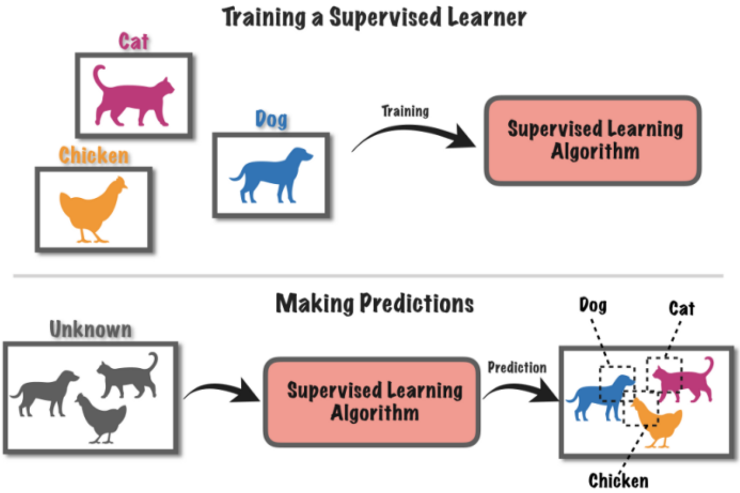
Subsection 1

## What is Supervised Learning?

# Supervised Learning

- When training a machine, supervised learning refers to a category of methods in which we teach or train a machine learning algorithm using data, while guiding the algorithm model with **labels** associated with the data.
- Supervised learning algorithms take a dataset and use its features to learn some relationship with a corresponding set of labels.
- This process is known as **training** and, once complete, we would hope that our algorithm would do a good job of predicting the labels of brand new data in which the algorithm has no explicit knowledge of the true label.

# Supervised Learning


Training a Supervised Learner
Cat
Chicken
Dog
Training
Supervised Learning Algorithm
Making Predictions
Unknown
Supervised Learning Algorithm
Prediction
Dog
Cat
Chicken

# Supervised Learning

- From a formal poin of view, supervised learning process involves input variables, which we call $X$, and an output variable, which we call $Y$.
- We use an algorithm to learn the mapping function from the input to the output.
- In simple mathematics, the output $Y$ is a dependent variable of input $X$ as illustrated by:

$$Y = f(X)$$

Here, our end goal is to try to **approximate the mapping function** $f$, so that we can **predict** the output variables $Y$ when we have new input data $X$.

Subsection 2

## Example: Predicting House Prices

# Iowa House Price Case Study

- The objective is to predict the prices of house in Iowa from features
- 800 observations in training set, 600 in validation set, and 508 in test set
- Here the original competition description: **https://www.kaggle.com/c/house-prices-advanced-regression-techniques**

# Iowa House Price Case Study

- How is this achieved?
- First, we need data about the houses: square footage, number of rooms, **features**, whether a house has a garden or not, and so on.
- We then need to know the prices of these houses, i.e. the corresponding **labels**.
- By leveraging data coming from thousands of houses, their features and prices, we can now train a supervised machine learning model to predict a new house's price based on the examples observed by the model.

# Back to Linear Regression

- Linear regression is a very popular tool because once you have made the assumption that the model is linear you do not need huge amount of data. In ML we refer to the constant term as the *bias* and the coefficients as *weights*
- Assume $n$ observations and $m$ features. Model is

$$Y = a + b_1 X_1 + b_2 X_2 + \cdots + b_m X_m + \epsilon$$

- Standard approach is to choose $a$ and the $b_i$ to minimize the mean square error (mse):

$$mse = \frac{1}{n}\sum_{j=1}^{n}\left[Y_j - (a + b_1 X_{1,j} + b_2 X_{2,j} + \cdots + b_m X_{m,j})\right]^2 \tag{1}$$

- This can be done analytically by inverting a matrix, in practice a numerical (gradient descent) is used.

# Categorical Features

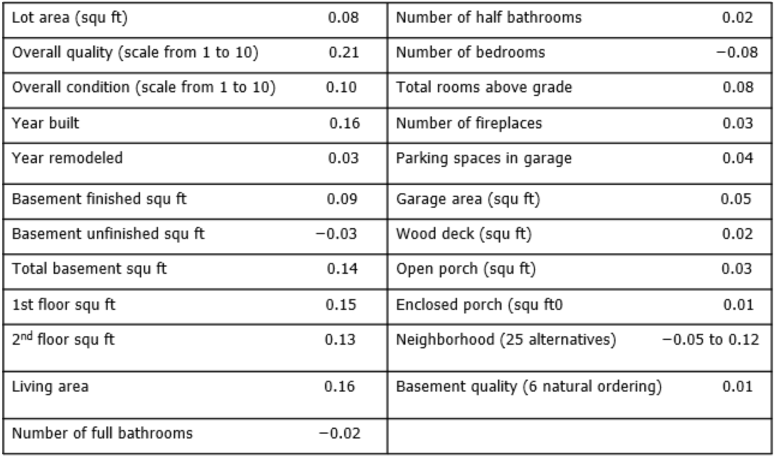
- Categorical features are features where there are a number of non-numerical alternatives
- We can define a dummy variable for each alternative. The variable equals 1 if the alternative is true and zero otherwise. This is known as **one-hot encoding**
- But sometimes we do not have to do this because there is a natural ordering of variables, e.g.:
  - small=1, medium=2, large=3
  - assist. prof=1, assoc. prof=2, full prof =3

# Dummy Variably Trap

- Suppose we have a constant term and a number of dummy variables (equal to 0 or 1)
- There is then no unique solution because, for any C, we can add C to the constant term and subtract C from each of the dummy variables without changing the prediction
- A side effect of regularization is that it solves this problem

# Iowa House Price Results (No regularization)

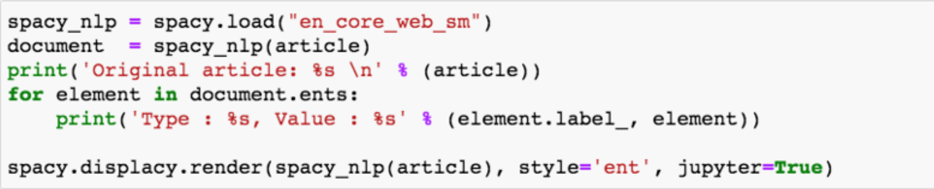
2 categorical variables included. Natural ordering for Basement quality. 25 dummy variables created for neighborhood



| | | | |
|---|---|---|---|
| Lot area (squ ft) | 0.08 | Number of half bathrooms | 0.02 |
| Overall quality (scale from 1 to 10) | 0.21 | Number of bedrooms | −0.08 |
| Overall condition (scale from 1 to 10) | 0.10 | Total rooms above grade | 0.08 |
| Year built | 0.16 | Number of fireplaces | 0.03 |
| Year remodeled | 0.03 | Parking spaces in garage | 0.04 |
| Basement finished squ ft | 0.09 | Garage area (squ ft) | 0.05 |
| Basement unfinished squ ft | −0.03 | Wood deck (squ ft) | 0.02 |
| Total basement squ ft | 0.14 | Open porch (squ ft) | 0.03 |
| 1st floor squ ft | 0.15 | Enclosed porch (squ ft0 | 0.01 |
| 2nd floor squ ft | 0.13 | Neighborhood (25 alternatives) | −0.05 to 0.12 |
| Living area | 0.16 | Basement quality (6 natural ordering) | 0.01 |
| Number of full bathrooms | −0.02 | | |

# Supervised Models

## Kaggle Competition: Iowa House Prices



```
spacy_nlp = spacy.load("en_core_web_sm")
document  = spacy_nlp(article)
print('Original article: %s \n' % (article))
for element in document.ents:
    print('Type : %s, Value : %s' % (element.label_, element))

spacy.displacy.render(spacy_nlp(article), style='ent', jupyter=True)
```

# Outline

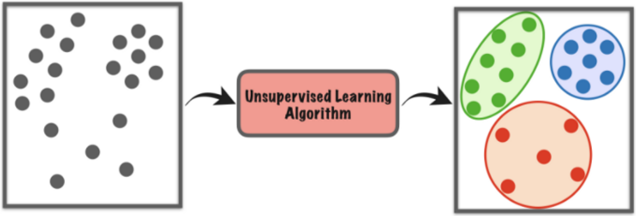
1. Supervised Learning
   - What is Supervised Learning?
   - Example: Predicting House Prices
2. Unsupervised Learning
   - What is Unsupervised Learning?
   - Example: k-Means Clustering
3. End Notes
   - Breaking the Dichotomy

Subsection 1

## What is Unsupervised Learning?

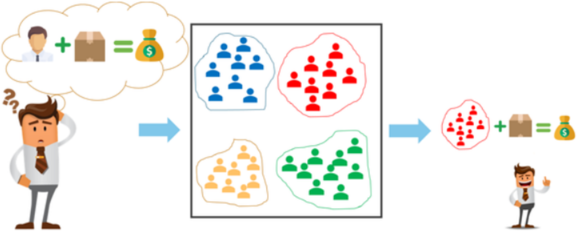
# Unsupervised Learning

- Unsupervised learning algorithms, on the other hand, work with data that isn't explicitly labelled.
- Unsupervised algorithms attempt to find some sort of underlying structure in the data.
- Are some observations clustered into groups? Are there interesting relationships between different features? Which features carry most of the information?



# Unsupervised Learning

- In unsupervised learning we are not trying to predict anything
- The objective is to cluster data to increase our understanding of the environment
- **Example - Clustering Customers**
  - Suppose you are a bank and have hundreds of thousands of customers and 100 features describing each one
  - Unsupervised learning algorithms can be used to divide your customers into clusters so that you can anticipate their needs and communicate with them more effectively



# Unsupervised Learning

- Also in contrast to supervised learning, assessing performance of an unsupervised learning algorithm is somewhat subjective and largely depend on the specific details of the task.
- Unsupervised learning is commonly used in tasks such as text mining and dimensionality reduction.
- K-means is an example of an unsupervised learning algorithm.

Subsection 2

## Example: k-Means Clustering

# The *k*-Means Algorithm

- In this section we explain a simple clustering procedure known as the *k-means algorithm*;
- *k*-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
- Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.
- The objective of K-means is simple: group similar data points together and discover underlying patterns.
- To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.

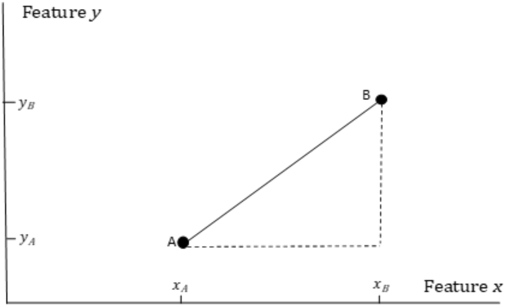
# The *k*-Means Algorithm

- A cluster refers to a collection of data points aggregated together because of certain similarities.
- You'll define a target number k, which refers to the number of centroids you need in the dataset.
- A centroid is the imaginary or real location representing the center of the cluster.
- Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.
- In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the **nearest** cluster, while keeping the centroids as small as possible.
- The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

# The *k*-Means Algorithm

## A Distance Measure

- For clustering we need a distance measure
- The simplest distance measure is the Euclidean distance measure.
  Distance $= \sqrt{(x_B - x_B)^2 + (y_B - y_A)^2}$



# The *k*-Means Algorithm

## Distance Measure

- In general when there are *m* features the distance between P and Q is

$$d = \sqrt{\sum_{j=1}^{m} (\nu_{pj} - \nu_{qj})^2} \tag{2}$$

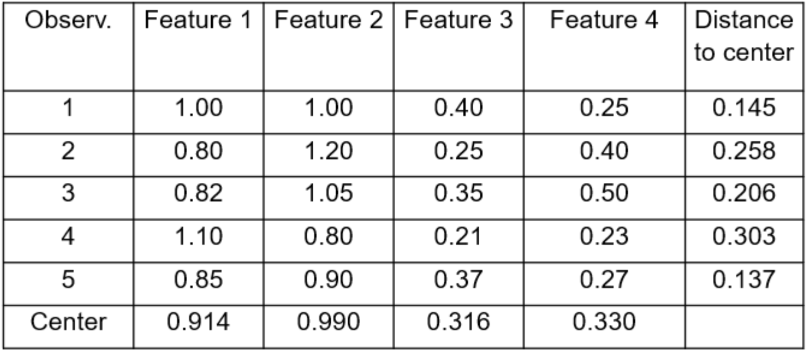
where $\nu_{pj}$ and $\nu_{qj}$ are the values of the $j - th$ feature for $P$ and $Q$

# The *k*-Means Algorithm

## Cluster Centers

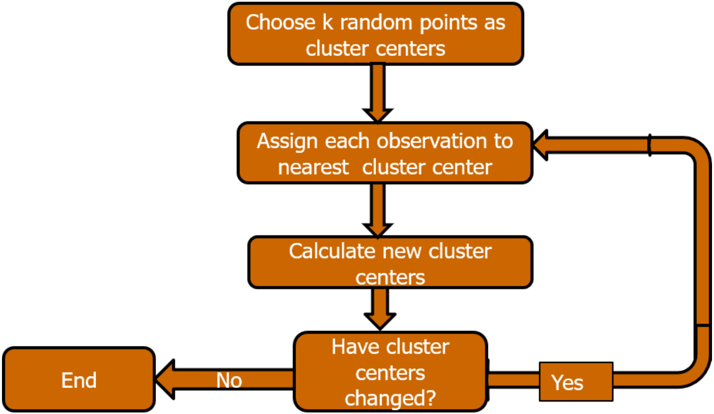
The center of a cluster (sometimes called the **centroid**) is determined by averaging the values of each feature for all points in the cluster.

**Example**:



| Observ. | Feature 1 | Feature 2 | Feature 3 | Feature 4 | Distance to center |
|---|---|---|---|---|---|
| 1 | 1.00 | 1.00 | 0.40 | 0.25 | 0.145 |
| 2 | 0.80 | 1.20 | 0.25 | 0.40 | 0.258 |
| 3 | 0.82 | 1.05 | 0.35 | 0.50 | 0.206 |
| 4 | 1.10 | 0.80 | 0.21 | 0.23 | 0.303 |
| 5 | 0.85 | 0.90 | 0.37 | 0.27 | 0.137 |
| Center | 0.914 | 0.990 | 0.316 | 0.330 | |

# The *k*-Means Algorithm to find k Clusters


Choose k random points as cluster centers
Assign each observation to nearest cluster center
Calculate new cluster centers
Have cluster centers changed?
No
Yes
End

# The *k*-Means Algorithm

## Cluster Centers

Inertia

- A measure of the performance of the algorithm is the within cluster sum of squares also known as *inertia*;
- For any given k the objective is to minimize inertia:

$$Inertia = \sum_{i=1}^{n} d_i^2 \tag{3}$$

  where $d_i$ is the distance of observation $i$ from its cluster center
- In practice we use the k-means algorithm with several different starting points and choose the result that has the smallest inertia

# The *k*-Means Algorithm

## Choosing k

- The elbow approach (see next slide)
- The silhouette method:
    - For each observation $i$ calculate $a(i)$, the average distance from other observations in its cluster, and $b(i)$, the average distance from observations in the closest other cluster. The silhouette score for observation $i$, $s(i)$, is defined as

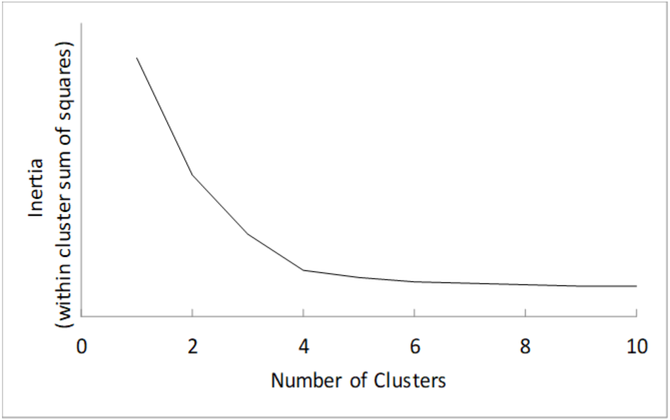
$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]} \tag{4}$$

    - Choose the number of clusters that maximizes the average silhouette score across all observations
- Use the gap statistic which compares the within cluster sum of squares with what would be expected with random data

# The *k*-Means Algorithm

## The elbow method

The **elbow method** (In this example k=4 is suggested)



# The $k$-Means Algorithm

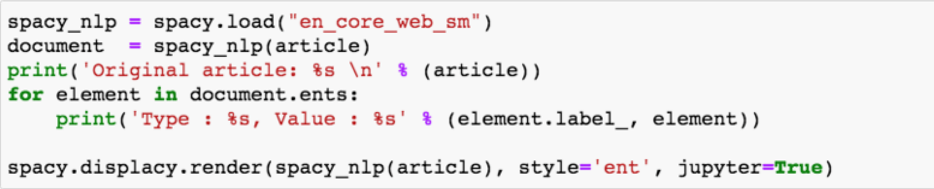
## The Curse of Dimensionality

- The Euclidean distance measure increases as the number of features increase.
- This is referred to as the curse of dimensionality
- Consider two observations that have values for feature $j$ equal to $x_j$ and $y_j$. An alternative distance measure that always lies between 0 and 2 is

$$d = 1 - \frac{\sum_{j=1}^{m} x_j y_j}{\sqrt{\sum_{j=1}^{m} x_j^2 \sum_{j=1}^{m} y_j^2}} \tag{5}$$

# k-Means Clustering

## Country Risk Example



```
spacy_nlp = spacy.load("en_core_web_sm")
document  = spacy_nlp(article)
print('Original article: %s \n' % (article))
for element in document.ents:
    print('Type : %s, Value : %s' % (element.label_, element))

spacy.displacy.render(spacy_nlp(article), style='ent', jupyter=True)
```

# Outline

1. Supervised Learning
   - What is Supervised Learning?
   - Example: Predicting House Prices

2. Unsupervised Learning
   - What is Unsupervised Learning?
   - Example: k-Means Clustering

3. End Notes
   - Breaking the Dichotomy

Subsection 1

## Breaking the Dichotomy

# Breaking the Dichotomy

- In recent years a number of paradigms have appeared that don't quite fit under the supervised and unsupervised labels.
- Semi-Supervised Learning is just what is sounds like, approaches that combine some labelled and some unlabelled data.
- Often labelling is an expensive, time consuming process so there are many situations where we would like to use information from a small amount of labelled data and a larger amount of unlabelled data.
- Also related to this situation is Active Learning where a learning algorithm can query a user to label particular observations which will add the most information.

# Bibliography

- John C. Hull, *Machine Learning in Business: An Introduction to the World of Data Science*, Amazon, 2019.
- Paul Wilmott, *Machine Learning: An Applied Mathematics Introduction*, Panda Ohana Publishing, 2019.