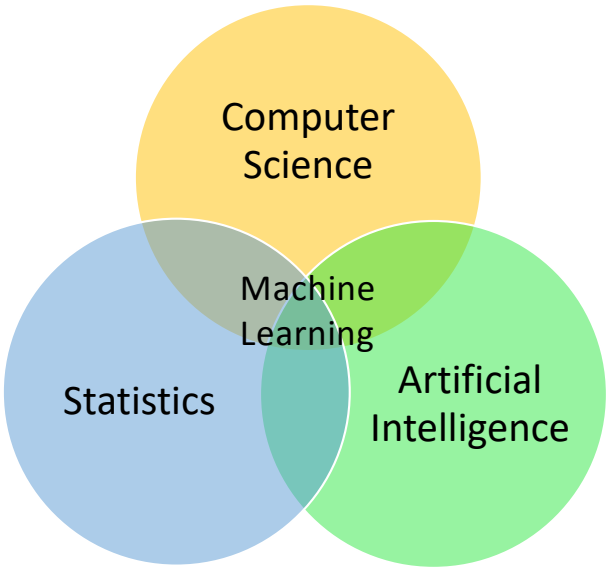
MET CS 688 Fundamentals of Machine Learning

Leila Ghaedi – Fall 2024

Creator: cemagraphics | Credit: Getty Images/iStockphoto

Machine Learning

- Machine learning is about extracting knowledge from data.
- It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning.



Machine Learning

- The application of machine learning methods has in recent years become ubiquitous in everyday life.
- From automatic recommendations of which movies to watch, to what food to order or which products to buy, to personalized online radio and recognizing your friends in your photos, many websites and devices have machine learning algorithms at their core.
- When you look at a website/app like Amazon, Netflix or Instagram, it is very likely that every part of the site contains multiple machine learning models.
- Machine Learning tools have been applied to diverse scientific problems such as understanding stars, finding distant planets, discovering new particles, analyzing DNA sequences, and providing personalized cancer treatments.

Outline of Machine Learning for Web Mining

- Web Mining is a subset of Machine Learning, focused on dealing with the data collected from web.

- Most of traditional web mining works focused on text mining. However, other disciplines and data are rising as well, e.g. searching Instagram photos for some specific information, using Youtube videos to extract human behavioral patterns, using twitch to quantify hate speech, etc.

Why ML is very important

Because every live creature is able to learn and adapt themselves to a new situation. **Computers were not able to learn,** they operated based on a given set of instruction (algorithm) to do a specific job. Machine learning, data mining and artificial intelligence is the science of **enabling computers to *learn* and *adapt* their reaction** [Chapmann '17] (Page 68).

- Scientific process is operated based on reproducibility and repeatability.
- Reproducibility is the ability of independent investigators to draw the same conclusions from an experiment by following the documentation shared by the original investigators.
- ML, AI, DM, … operate based on the notion of repeatability.

What is ML, DM, AI?

- Data mining: It is the process of extracting (mining) knowledge from the data.
- Machine Learning:
  - Mitchell [Mitchell '97] states machine learning is the study of "*how to construct computer programs that automatically improve with experience*".
  - Géron [Géron '17] defines machine learning as a "*science of programming computers so they learn from data*"

- Machine learning is a technique in which knowledge is extracted from data. The process of applying a machine learning technique on the data to extract knowledge called data mining.

A.I. What can we do with data?

In general, we can use three techniques with the data include:

- Describing or diagnosing a phenomenon (classification, clustering)
- Predicting an event or changes based on the available data (prediction)
- Creating a system that use data and mimic a cognitive capability of a human behavior, e.g. finding a cat in a picture or understanding a handwritten text. (Artificial Intelligence). This includes generating artifacts similar to human, such as composing a poem, drawing a picture, etc.

Artificial Intelligence: Thinking Rationally, Acting Rationally, Thinking Humanly, Acting Humanly. [Russel ’09] (Page 2.)

Machine learning, data mining and artificial intelligence is the science of enabling computers to learn and adapt their reaction [Chapmann ’17] (Page 68).

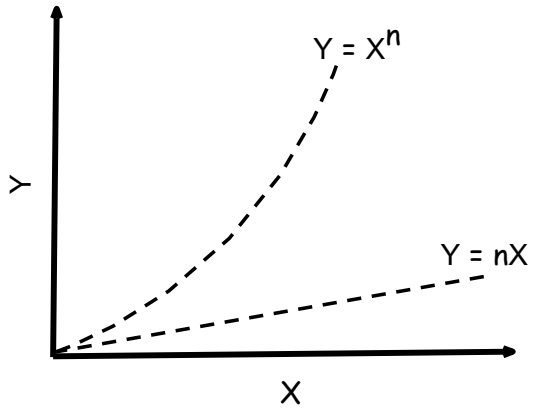
Algorithm

- All computer science is based on algorithm. We give a one or more than one input into the algorithm, system, machine, … and it produces one or more output.

f(x)=y

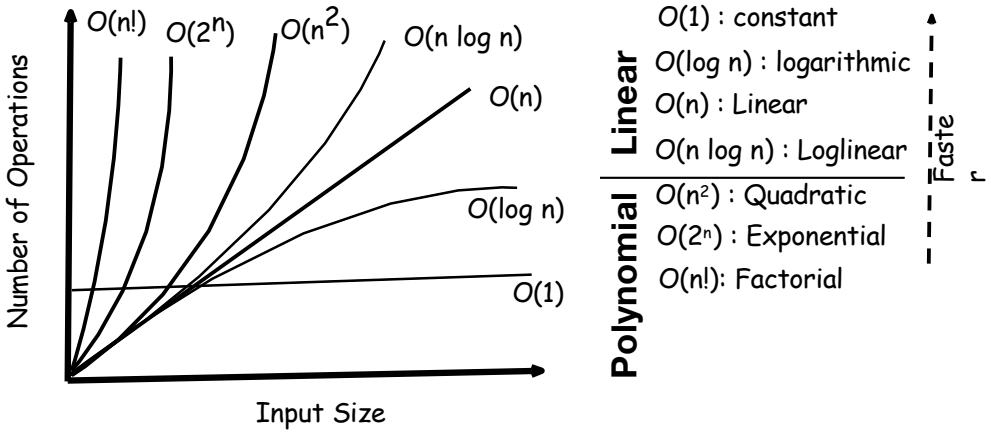
How to Evaluate ML Algorithm?

1. Efficiency and Computational Complexity (Space, Time). Along with efficiency we report resource utilization including response time, memory use, … as well.
2. Accuracy

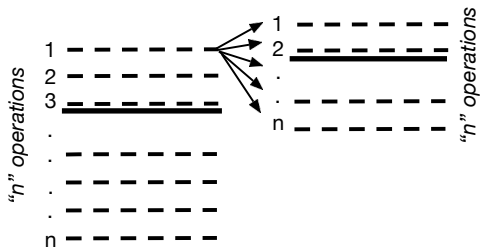


Linear and exponential growth example.

Algorithm Efficiency



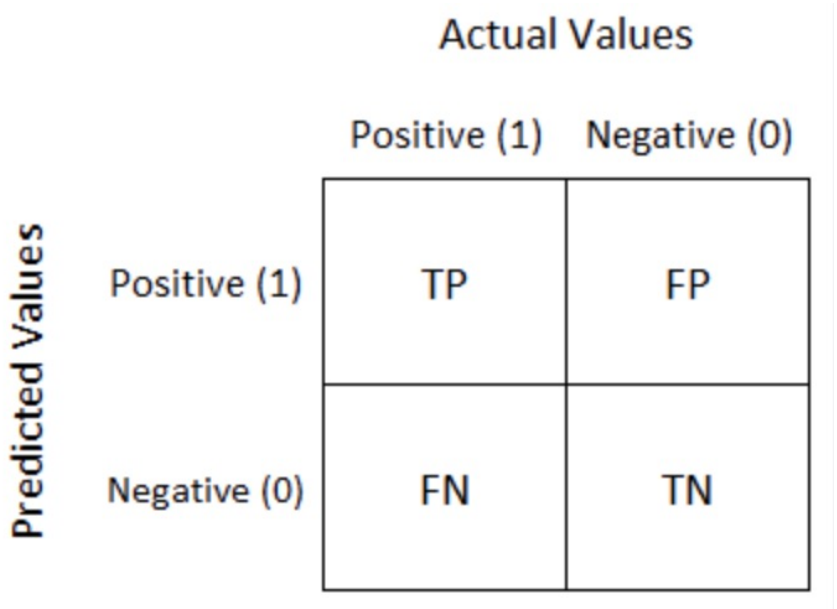
batch learning
vs
online learning




```
for (1 to n)
    for(1 to n)
```

Toy example of n^2 computational complexity.

Confusion Matrix



Accuracy Calculations

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F-Score} = \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

ML Input

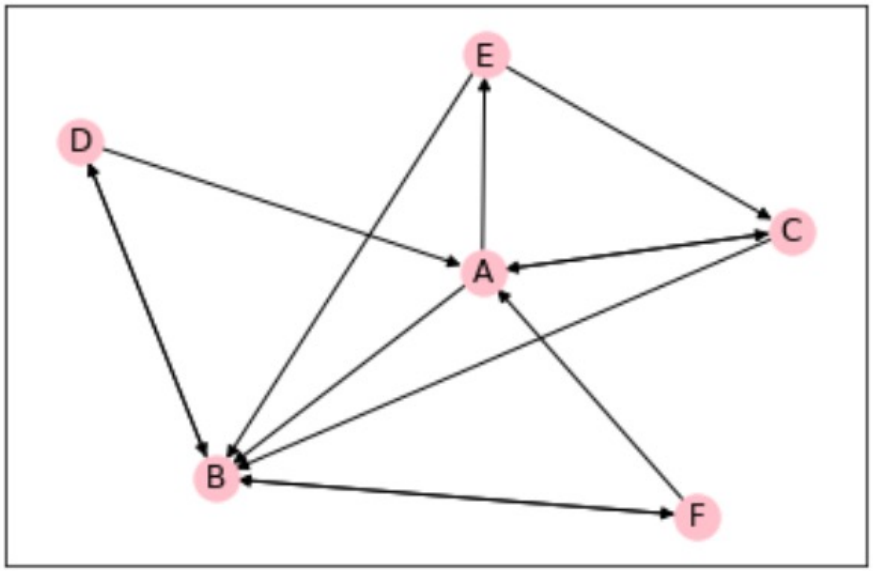
- CSV, JSON, XML, ARFF, …
- Temporal Data and Timestamped Data (regular time series)
- Data Stream (sequence)
- Graph Dataset (use nodes to store data entities and edges to store relationship between entities)
- HTML, PDF, Word documents or any other form of text document

Tabular Data Examples

sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
5.4	3.9	1.7	0.4	Setosa
4.6	3.4	1.4	0.3	Setosa
5	3.4	1.5	0.2	Setosa
4.4	2.9	1.4	0.2	Setosa
4.9	3.1	1.5	0.1	Setosa

The characteristics of tabular data are: They consists of rows and columns.

Graph Data Example



Input Variables

Scalar

20

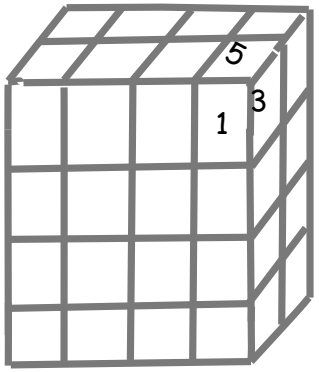
Vector

$[59, 22, 33]$

Matrix

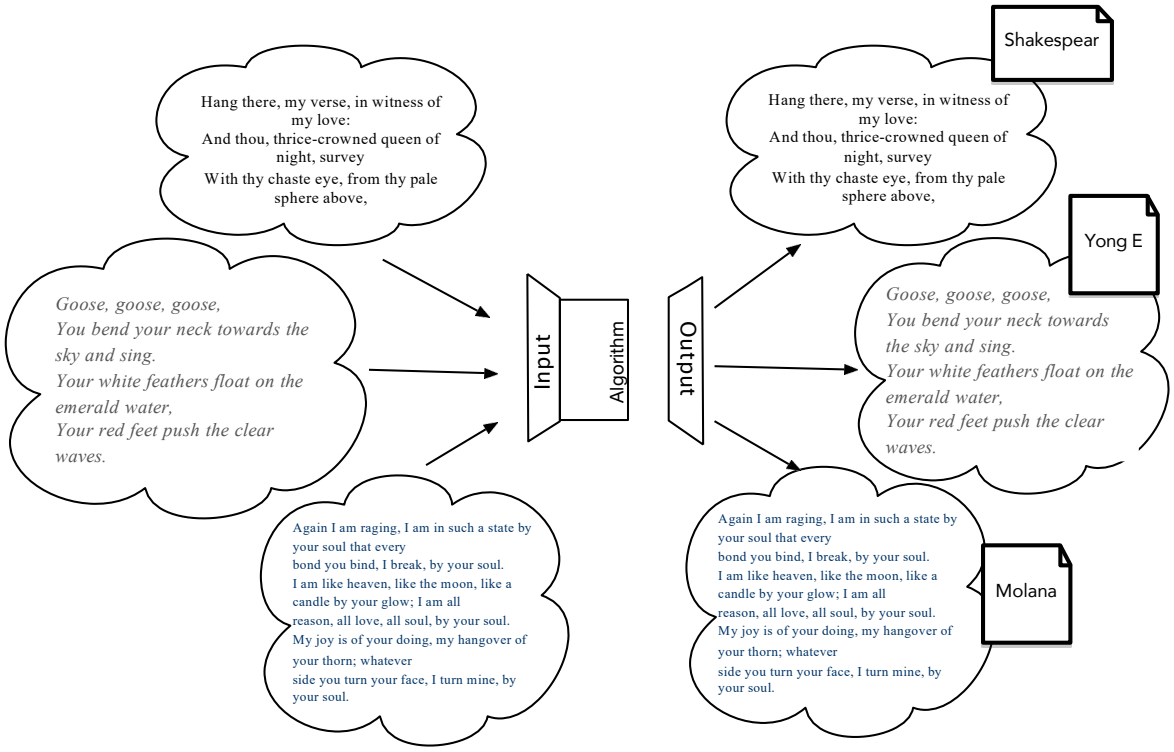
$$\begin{bmatrix} 1 & 5 \\ 2 & 3 \\ 3 & 90 \end{bmatrix}$$

Tensor



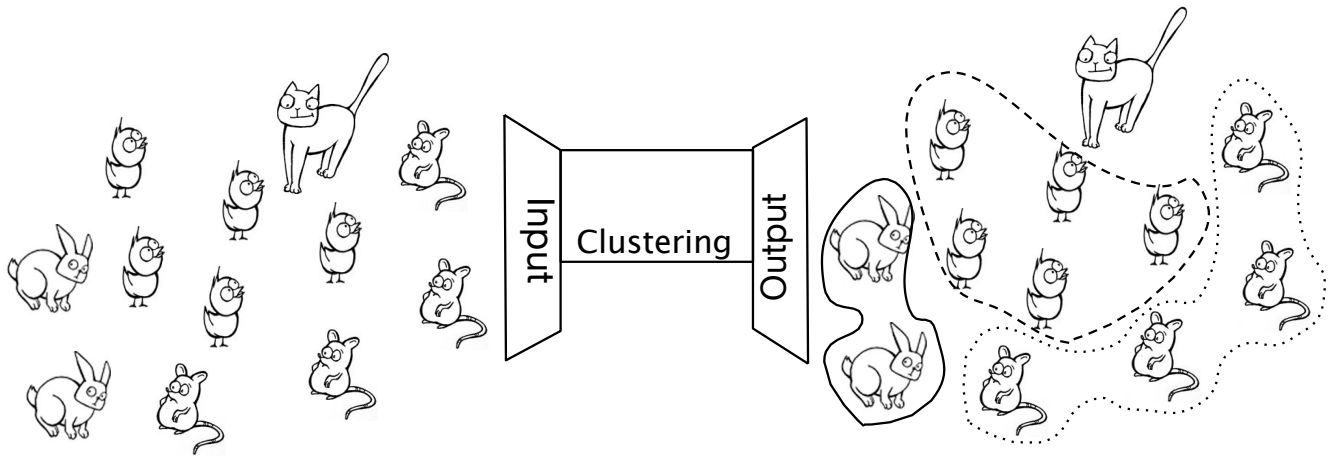
- A tensor is a geometrical object that describes a multilinear relationship between sets of objects related to a vector space. Tensors may map between different objects such as vectors, scalars, and even other tensors.
- Think of matrix as a two_dimensional tensor.

Output (speech/text recognition)

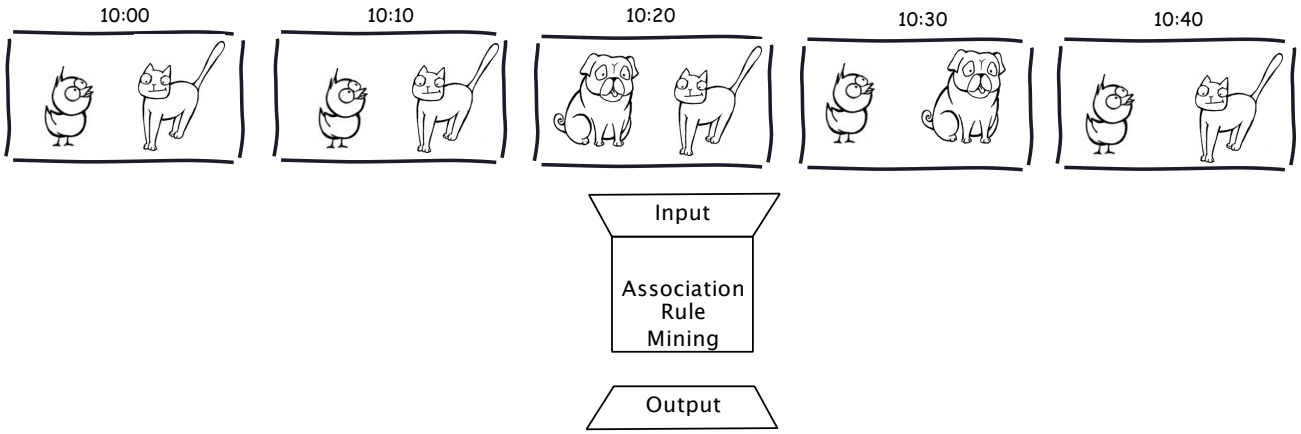


Output (Image Recognition)

Clustering



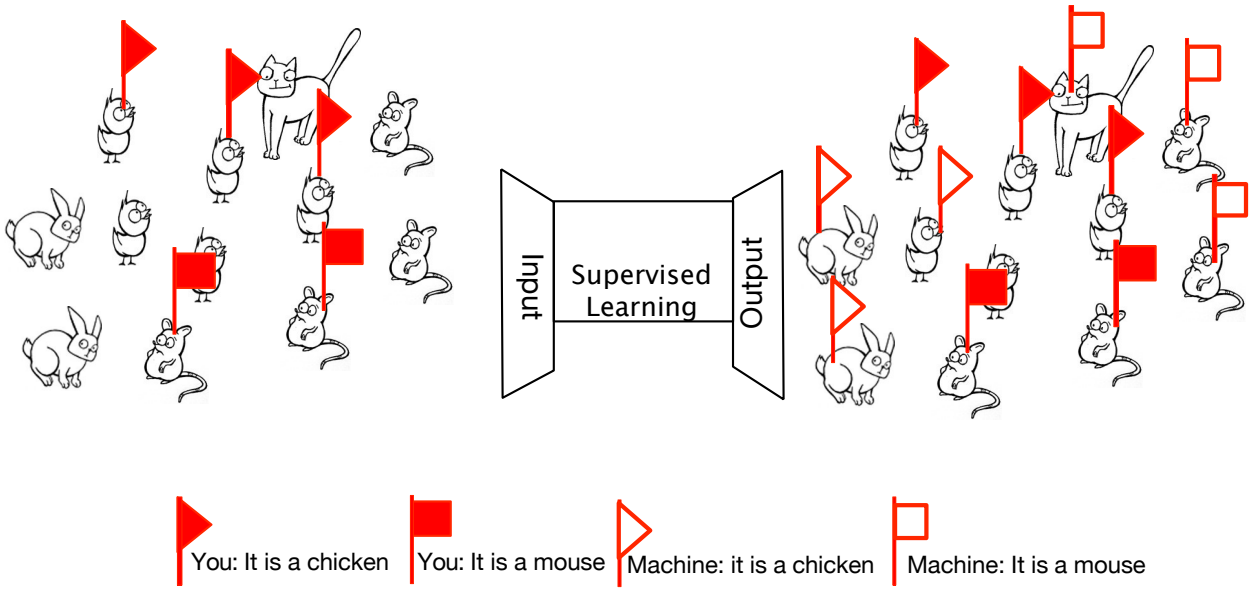
Co-occurrences and Association Rule, Sequence Mining



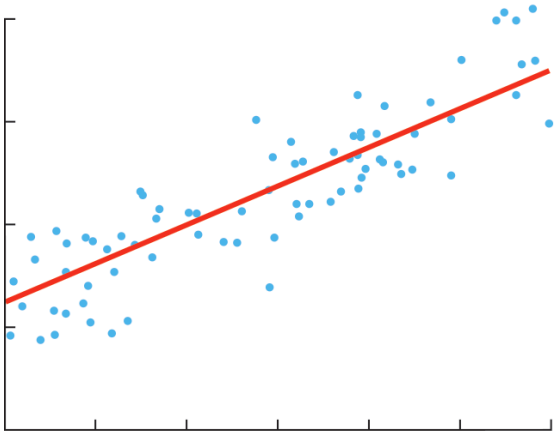
{Chicken, Cat} Confidence = 60%

Association rule mining finds interesting associations and relationships among large sets of data items.

Classification



Regression & Correlation



SOURCE HBR.ORG

- Regression is measuring a changes of one variable along another variable changes.
- Correlation is used for understanding the strength of relationship between two variables.

Process of Data Mining

- Defining the question and collecting data
- Explore the data with visualization (if possible) and statistics
- Data cleaning and wrangling
- Preparing data for the algorithm (feature engineering, dimensionality reduction)
- Running the algorithm
- Evaluation

Supervised vs Unsupervised Learning

- Supervised Learning:
  - Algorithms that automate decision-making processes by generalizing from known examples. The user provides the algorithm with pairs of inputs and desired outputs, and the algorithm finds a way to produce the desired output given an input.
- Unsupervised Learning:
  - In unsupervised learning, only the input data is known, and no known output data is given to the algorithm. These algorithms discover hidden patterns or data groupings without the need for labeling. Unsupervised learning methods are usually harder to understand and evaluate.

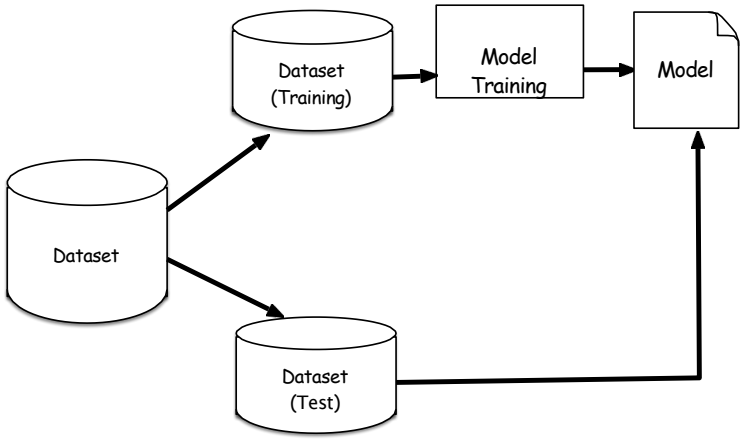
Supervised Learning Examples

- Determining benign or malignant tumor based on a medical image:
  - The image of tumor is the input of the algorithm. The output is whether the tumor is benign or malignant evaluated by experts. It might even be necessary to do additional diagnosis beyond the content of the image to determine whether the tumor in the image is cancerous or not.
- Detecting fraudulent activity in credit card transactions:
  - Here the input is a record of the credit card transaction, and the output is whether it is likely to be fraudulent or not. The label is if a user reports any transaction as fraudulent.

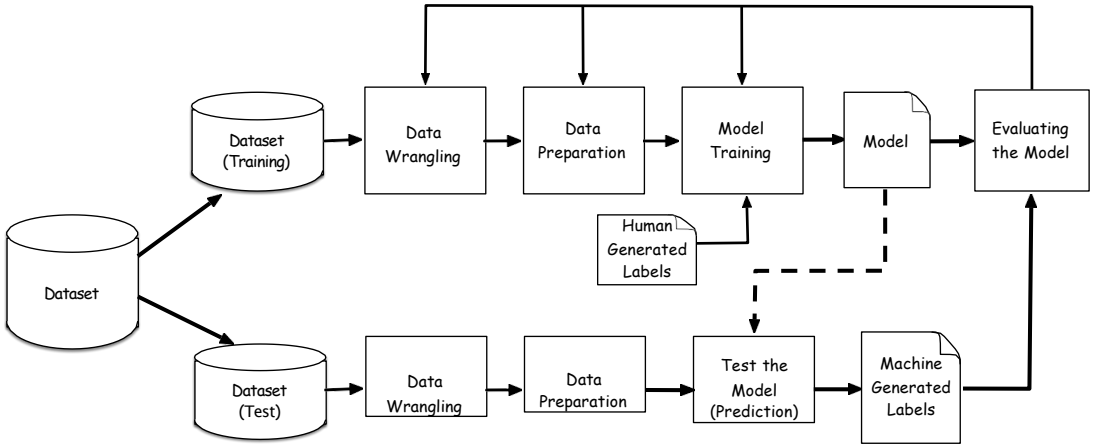
Unsupervised Learning Examples

- Identifying topics in a set of blog posts:
  - Imagine trying to summarize and categorize a large collection of text data. There is no prior information about the number of topics and themes. Therefore, there are no known outputs.
- Identifying behavioral personas for customers
  - Identifying groups of people who have similar customer behavior. Behavioral personas describe your customers based on what they do: their interactions with a specific product (imagine using an app or going to a retail location) rather than demographics or market data.

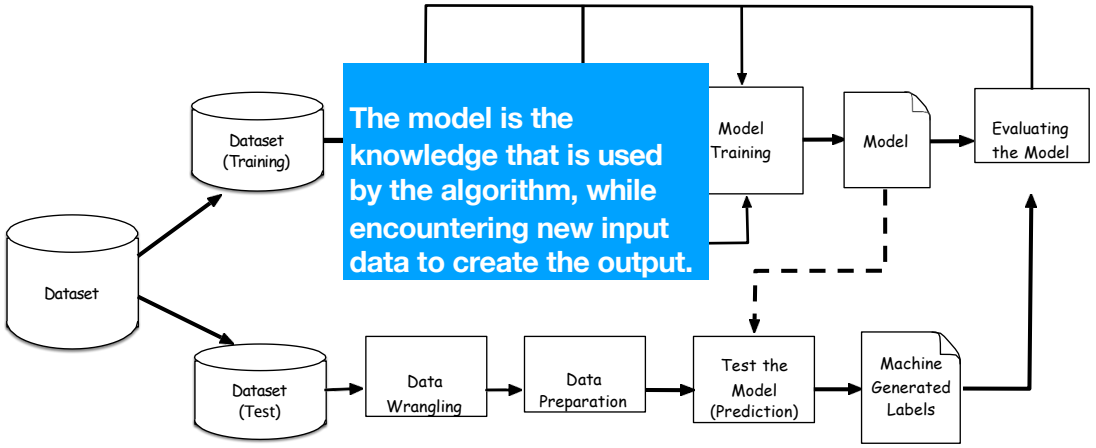
Supervised Learning Simplified



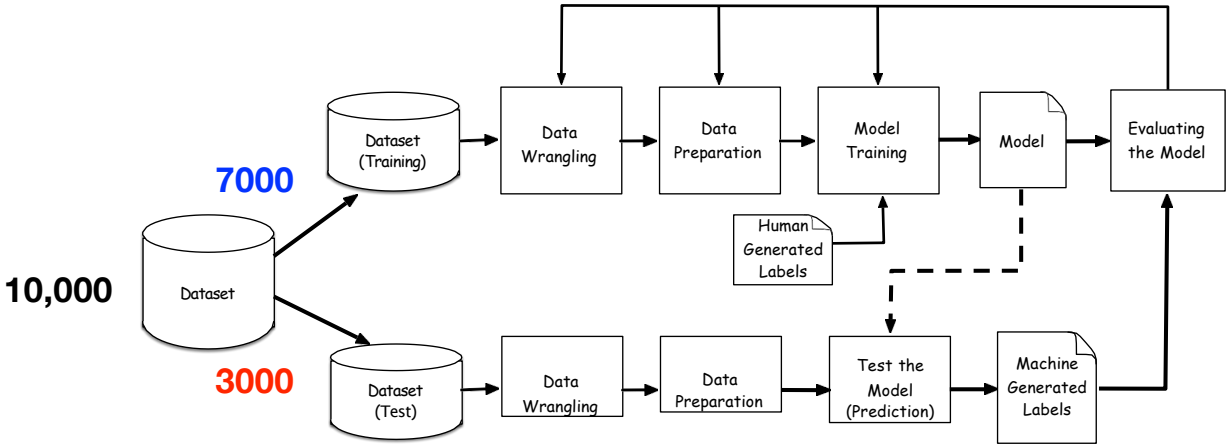
Supervised Learning



Supervised Learning



Supervised Learning



Python

- Python has become the bridge language for many data science applications.
- Python combines the power of general-purpose programming languages with the ease of use of domain- specific scripting languages like MATLAB or R.
- Libraries for data loading, visualization, statistics, natural language processing, image processing, …
- You can interact directly with the code, using a terminal or other tools like the Jupyter Notebook.

scikit-learn

- scikit-learn is an open-source project, meaning that it is free to use and distribute, and anyone can easily obtain the source code.
- The scikit-learn project is constantly being developed and improved, and it has a very active user community.
- It contains a number of state-of-the-art machine learning algorithms, as well as comprehensive documentation about each algorithm.
- scikit-learn is a very popular tool, and the most prominent Python library for machine learning
- scikit-learn works well with a number of other scientific Python tools. It depends on two other Python packages, NumPy and SciPy.

Anaconda

- Anaconda is a distribution of the Python and R programming languages for scientific computing, large-scale data processing and predictive analytics
- It simplifies package management and deployment.
- The distribution includes data-science packages suitable for Windows, Linux, and macOS.
- It includes libraries such as **NumPy**, **SciPy**, **matplotlib**, **pandas**, **IPython**, **Jupyter Notebook**, and **scikit-learn.**

Jupyter Notebook

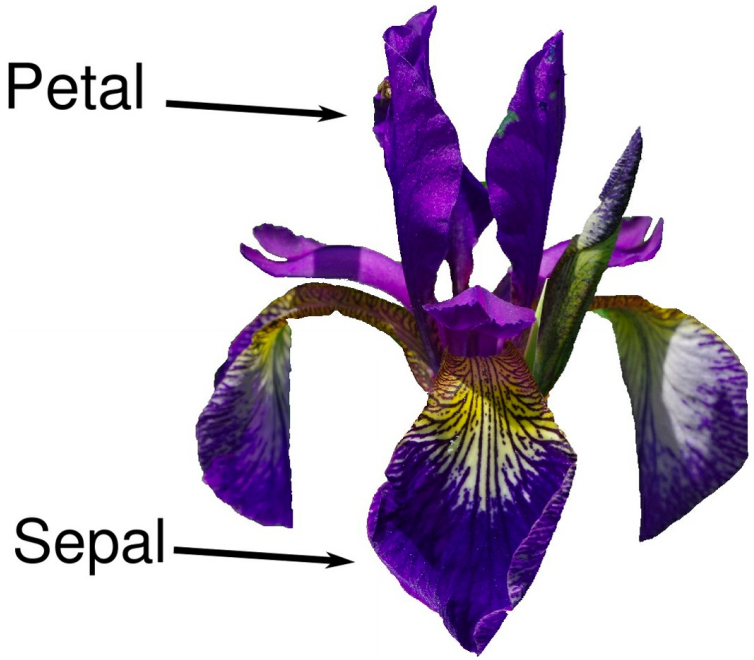
- Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages.
- It is a great tool for exploratory data analysis and is widely used by data scientists.

Classifying Iris Species

- We will go through a simple machine learning application and create our first model.
- In the process, we will introduce some core ML concepts and terms.

Iris Species Classification

- We have the measurements of some irises that have been previously identified by an expert botanist as belonging to the species setosa, versicolor, or virginica.
- The possible outputs (different species of irises) are called classes.
- The goal is to build a machine learning model which can learn from labeled samples, so it can predict the species for a new iris.



In [1]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
```

In [2]:
```
# LOAD THE IRIS DATASET BY CALLING THE FUNCTION
iris_dataset = load_iris()
print("Keys of iris_dataset:\n", iris_dataset.keys())
```

```
Keys of iris_dataset:
 dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

In [3]:
```
print(iris_dataset['DESCR'][:900] + "\n...")
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:
...
```

In [4]:
```
print("Type of data:", type(iris_dataset['data']))
print("Shape of data:", iris_dataset['data'].shape)
```

```
Type of data: <class 'numpy.ndarray'>
Shape of data: (150, 4)
```

In [5]:
```
print("First five rows of data:\n", iris_dataset['data'][:5])
```

```
First five rows of data:
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
```

In [6]:
```
print("Type of target:", type(iris_dataset['target']))
print("Shape of target:", iris_dataset['target'].shape)
```

```
Type of target: <class 'numpy.ndarray'>
Shape of target: (150,)
```

In [7]:
```
print("Target:\n", iris_dataset['target'])
```

```
Target:
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

Training and Testing

- A model should generalize well meaning it performs well on the new data
- The data that is used for evaluating the model should be different than the data that is used to build the model.
- train_test_split function shuffles the dataset and splits it into training and test sets. (75% and 25% default split)

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], test_size=0.2, random_state=0)
```

In [9]:
```
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
```

```
X_train shape: (120, 4)
y_train shape: (120,)
```

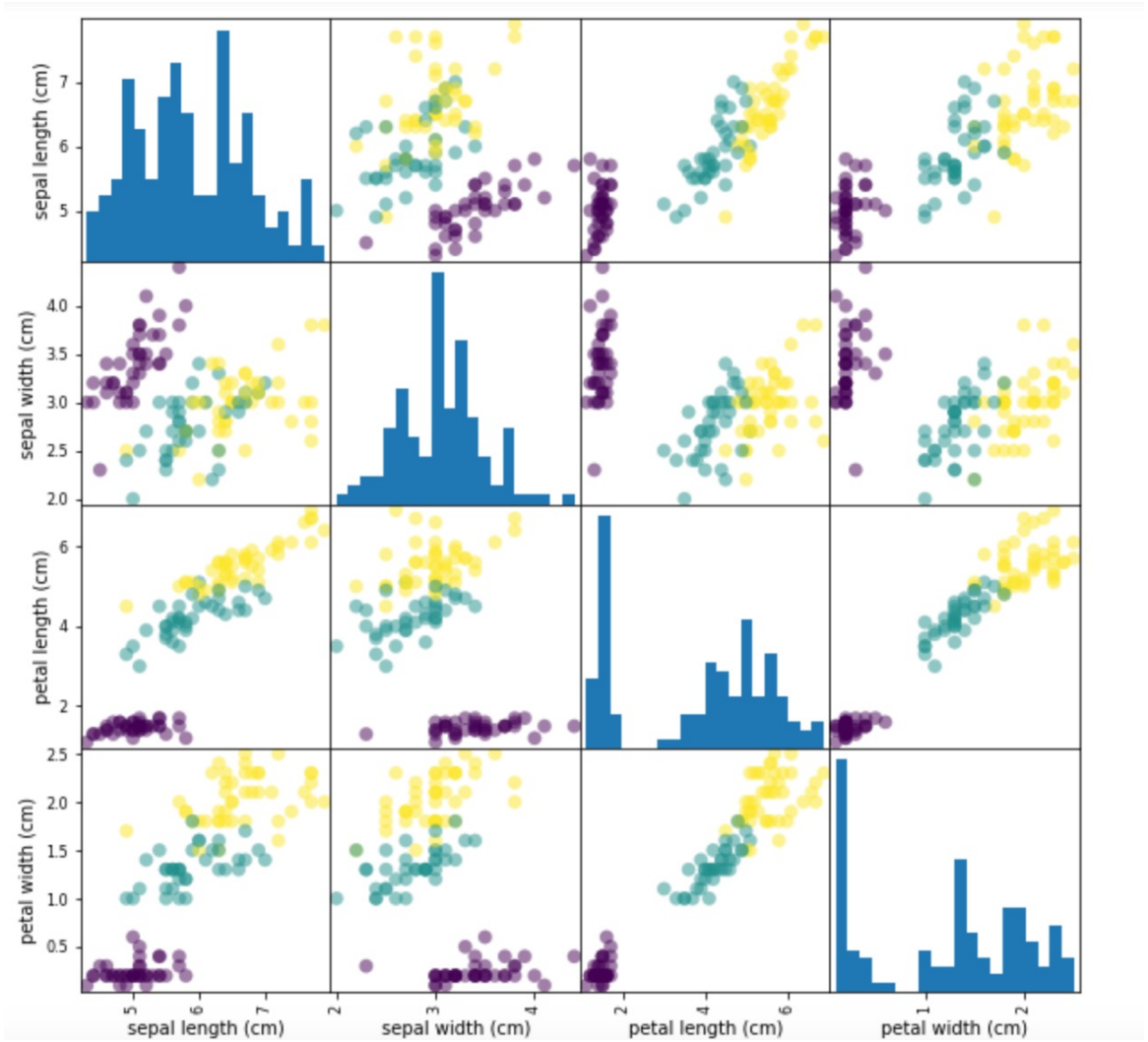
Exploratory Data Analysis (EDA)

- EDA is the process of analyzing data sets to summarize their main characteristics by statistical analysis and visualization.
- EDA used for extracting vital features and trends in the data.
- A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing.

Exploratory Data Analysis

- By visualizing a pair plot of the features in the training set, we can see how relevant each feature could be. The data points are colored according to the species the iris belongs to.
- Pandas has a function to create pair plots called scatter_matrix. The diagonal of this matrix is filled with histograms of each feature:

```
In [10]: iris_dataframe = pd.DataFrame(X_train,columns=iris_dataset.feature_names)
         # create a scatter matrix from the dataframe, color by y_train
         pd.plotting.scatter_matrix(iris_dataframe, c=y_train, figsize=(10, 10),
                                    marker='o', hist_kwds={'bins': 20}, s=60,
                                    alpha=.5)
```


sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

EDA findings

- From the plots, we can see that the three classes seem to be relatively well separated using the sepal and petal measurements. This means that a machine learning model will likely be able to learn to separate them.

Building Your First Model: k-Nearest Neighbors (KNN)

- KNN algorithm is a non-parametric supervised learning method
- KNN classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data.
- The k in KNN signifies that instead of using only the closest neighbor to the new data point, we can consider any fixed number k of neighbors in the training (for example, the closest three or five neighbors). Then, we can make a prediction using the majority class among these neighbors.

In [11]:
```
from sklearn.neighbors import KNeighborsClassifier
iris_knn = KNeighborsClassifier(n_neighbors=5)
```

In [12]:
```
iris_knn.fit(X_train, y_train)
```

Out[12]:
```
KNeighborsClassifier()
```

In [13]:
```
y_pred = iris_knn.predict(X_test)
print("Test set predictions:\n", y_pred)
```

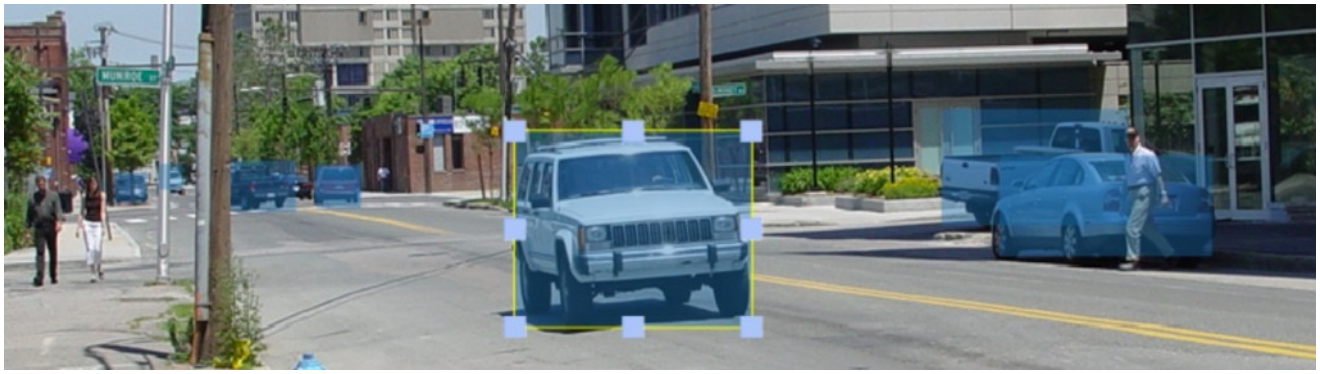
```
Test set predictions:
 [2 1 0 2 0 2 0 1 1 1 2 1 1 1 2 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
```

In [14]:
```
print("Test set score: {:.2f}".format(np.mean(y_pred == y_test)))
```

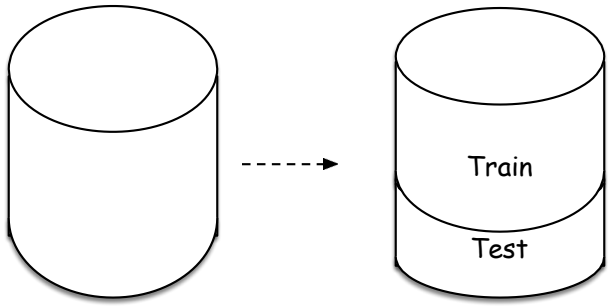
```
Test set score: 0.97
```

Ground Truth Dataset

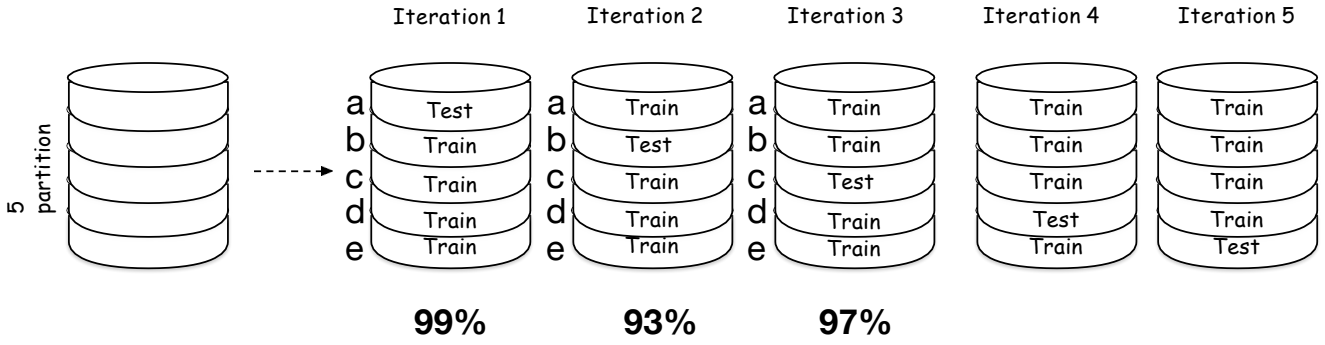
The ground truth dataset is a dataset that is annotated manually by humans and we assume the annotations are correct. To measure an accuracy of an algorithm we will compare its output to the ground truth dataset.



Hold Out Validation Method



k-fold Cross Validation Method



Unsupervised Learning

- Types of unsupervised learning:
  - **Unsupervised transformations**
    - Algorithms that create a new representation of the data which might be easier for humans or other machine learning algorithms to understand compared to the original representation of the data.
    - A common application of unsupervised transformations is dimensionality reduction
    - A common application for dimensionality reduction is reduction to two dimensions for visualization purposes.
  - **Clustering algorithms**
    - Partition data into distinct groups of similar items (clusters). The idea is that objects in the same cluster are more like each other than to those in other clusters.

Dimensionality Reduction

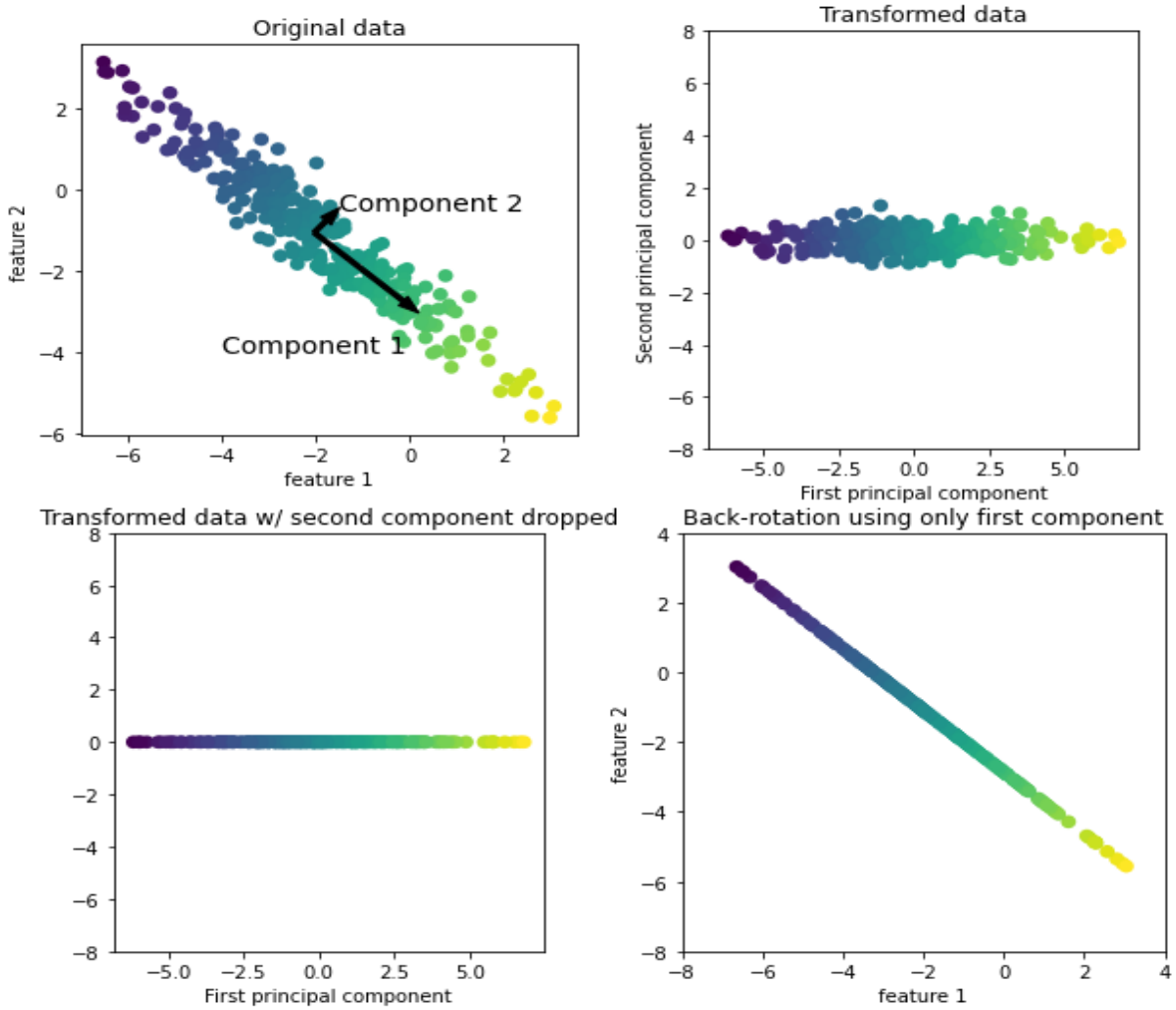
- Transforming data using unsupervised learning can have many motivations. The most common motivations are visualization, compressing the data, and finding a representation that is more informative for further processing.
  - Principal Component Analysis (PCA), is most commonly used when many of the variables are highly correlated with each other and it is desirable to reduce their number to an independent set.
    Non-Negative Matrix Factorization (NMF), which is commonly used for feature extraction.
  - t-Distributed Stochastic Neighbor Embedding (t-SNE), which is commonly used for visualization using two-dimensional scatter plots.

Principal Component Analysis (PCA)

- Principal component analysis is a method that rotates the dataset in a way such that the rotated features are statistically uncorrelated.
- This rotation is often followed by selecting only a subset of the new features, according to how important they are for explaining the data.
- The following slide illustrates the effect of PCA on a synthetic two-dimensional dataset.

```
import mglearn
import matplotlib.pyplot as plt
mglearn.plots.plot_pca_illustration()
plt.suptitle("pca_illustration");
```

pca_illustration

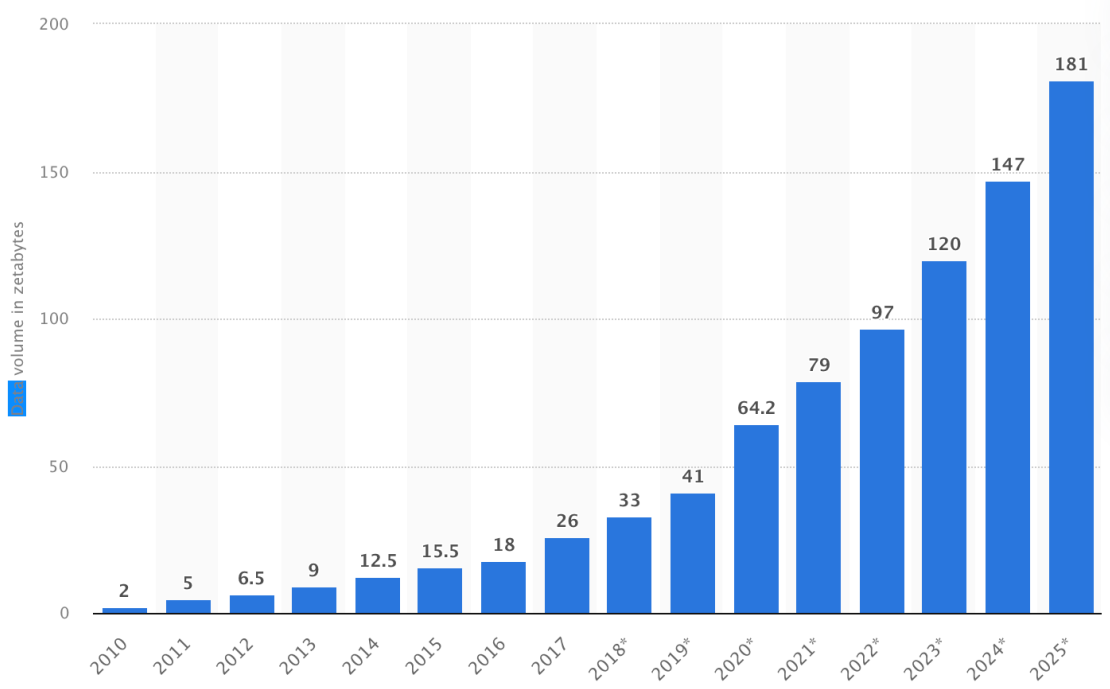


MET CS 688
Web Scraping & Web Crawling

Leila Ghaedi – Fall 2024

Creator: cemagraphics | Credit: Getty Images/iStockphoto

Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 (In Zettabytes 10^{21})



https://www.statista.com/statistics/871513/worldwide-data-created/

Web Scraping and Web Crawling

- The automated gathering of data from the internet is nearly as old as the internet itself.
- **Web scraping** is the practice of gathering data through any means other than a program interacting with an API.
- Alternative terms:
  - Screen Scraping
  - Data Mining
  - Web Harvesting
- **Web Crawler** are programs that specifically traverse all web pages of a site or a number of sites . These programs, or bots, are most commonly used to create entries for a search engine index.

Web Scraping vs Browsing

- If the only way you access the internet is through a browser, you're missing out on a huge range of possibilities.
- Web scrapers are excellent at gathering and processing large amounts of data quickly. They enable us to view databases spanning thousands or even millions of pages at once.

Web Scraping vs APIs

- APIs are designed to provide a convenient stream of well-formatted data from one computer program to another.
- In general, it is preferable to use an API (if one exists), rather than build a bot to get the same data. However, an API might not exist or be useful for your purposes, for several reasons:
  - The data you want is relatively small or uncommon, and the creator did not think it warranted an API.
  - The source does not have the infrastructure or technical ability to create an API.
  - The data is valuable and/or protected and not intended to be spread widely.
  - Even when an API does exist, the request volume and rate limits, the types of data, or the format of data that it provides might be insufficient for your purposes.

Web scraping basic steps:

1. Retrieving HTML data from a domain name
2. Parsing that data for target information
3. Storing the target information
4. Optionally, moving to another page to repeat the process

Build a web scraper

urlopen is used to open a remote object across a network and read it. It is a fairly generic function (it can read HTML files, image files, or any other file stream with ease).

More accurately, this outputs the HTML file page1.html, found in the directory <web root>/pages, on the server located at the domain name http://pythonscraping.com.

In [1]:
```
from urllib.request import urlopen
html = urlopen('http://pythonscraping.com/pages/page1.html')
print(html.read())
```

```
b'<html>\n<head>\n<title>A Useful Page</title>\n</head>\n<body>\n<h1>An Interesting Title</h1>\n<div>\nLorem ipsum
dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaeca
t cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.\n</div>\n</body>\n</html>\
n'
```

BeautifulSoup

Beautiful Soup, so rich and green,

Waiting in a hot tureen!

Who for such dainties would not stoop?

Soup of the evening, beautiful Soup!

The BeautifulSoup library was named after a Lewis Carroll poem of the same name in Alice's Adventures in Wonderland.

It helps format and organize the messy web by fixing bad HTML and presenting us with easily traversable Python objects representing XML structures.

```
$ pip install beautifulsoup4
```

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page1.html')
bs = BeautifulSoup(html.read(), 'html.parser')
print(bs.h1)
```

```
<h1>An Interesting Title</h1>
```

- This returns only the first instance of the h1 tag found on the page.
- By convention, only one h1 tag should be used on a single page, but conventions are often broken on the web, so be aware that this will retrieve the first instance of the tag only.

BeautifulSoup

- As in previous web scraping examples, we calledhtml.read() in order to get the HTML content of the page.
- In addition to the text string, BeautifulSoup can also use the file object directly returned by urlopen, without needing to call .read() first.

```
html = urlopen('http://www.pythonscraping.com/pages/page1.html')
bs = BeautifulSoup(html.read(), 'html.parser')
print(bs.h1)
```

```
<h1>An Interesting Title</h1>
```

BeautifulSoup Object

- This HTML content is then transformed into a BeautifulSoup object, with the following structure:
- html → <html><head>...</head><body>...</body></html>
  - head → <head><title>A Useful Page<title></head>
    - title → <title>A Useful Page</title>
- body → <body><h1>An Int...</h1> Lorem ip... </body>
  - h1 → <h1>An Interesting Title</h1>
  - div → Lorem Ipsum dolor...
- The h1 tag that we extract from the page is nested two layers deep into your BeautifulSoup object structure (html → body → h1).
- All these will have the same output:
  - bs.h1
  - bs.html.body.h1
  - bs.body.h1
  - bs.html.h1

Connecting Reliably & Handling Exceptions

```
from urllib.request import urlopen
from urllib.error import HTTPError
from bs4 import BeautifulSoup

def getTitle(url):
    try:
        html = urlopen(url)
    except HTTPError as e:
        return None
    try:
        bs = BeautifulSoup(html.read(), 'html.parser')
        title = bs.body.h1
    except AttributeError as e:
        return None
    return title

title = getTitle('https://www.wellsfargo.com/')
if title == None:
    print('Title could not be found')
else:
    print(title)
```

```
<h1 class="hidden" id="skip">Wells Fargo</h1>
```

Scrapy

- Scrapy is a free and open-source web-crawling framework written in Python. Originally designed for web scraping, it can also be used to extract data using APIs or as a general-purpose web crawler.
- You still need to define page templates, give it locations to start scraping from, and define URL patterns for the pages that you're looking for. But in these cases, it provides a clean framework to keep your code organized.

Scrapy vs BeautifulSoup

- BeautifulSoup is a library and Scrapy is a complete framework.
- Scrapy can perform similar tasks quicker than BeautifulSoup.
- You would input a root URL to Scrapy then it starts crawling. You can specify constraints on how many (number of) URLs you want to crawl and fetch.
- BeautifulSoup only fetches the contents of the URL that you give and then stops. It does not crawl unless you create a loop with certain criteria.

History of Web Analysis

- Starting 1995, the internet community started seeing a **hit counter** on web sites.
- This plug-in counter communicated web sites' popularity, so almost everyone wanted it.
- However, such counters are not accurate.

As the web-design industry matured, hit counters slowly vanished.

Web Analysis

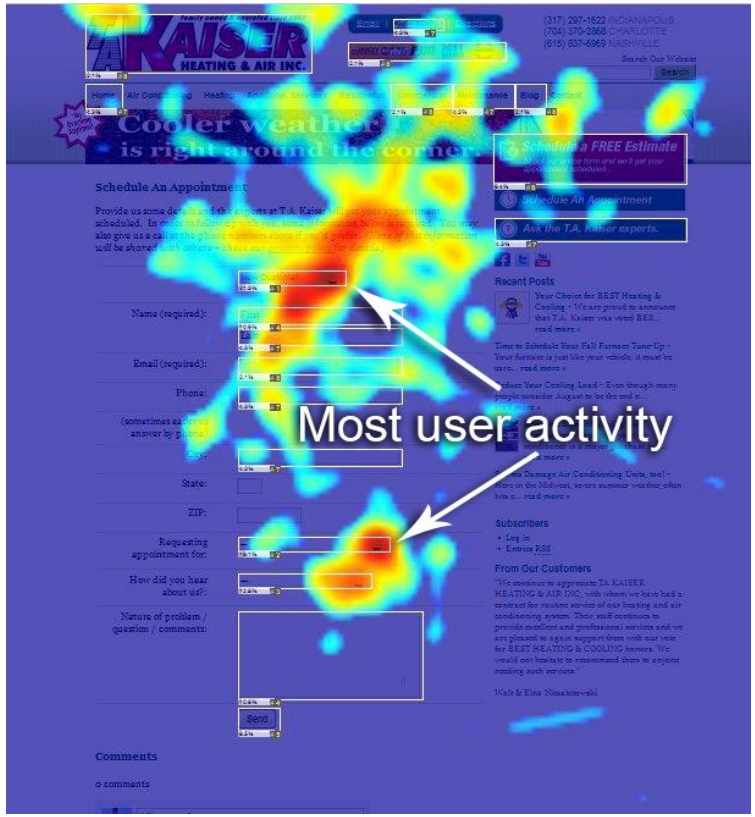
- Commercial web-analytics programs appeared later, with a company called *Webtrends* leading the way. The *Webtrends* software package produced visualizations that appealed to many users.

webtrends

- Years later, however, the web site–analytics industry introduced software that was able to measure ***click density***, or ***site overlay***,and ***heatmaps****.*

Heatmaps are used to show how users interact with a website, while hit counters show how popular a website is.

Heatmaps give you an aggregate of engagement across your website pages and can be split by clicks, taps, and scrolling behavior—allowing you to quickly see things like the average drop-off point on your blog posts or what your users expected to be clickable during checkout.



source: http://zachhiellermarketing.com/blog/2015/2/4/testing-with-heat-maps

Web Page Analytical Tools

1. Google Analytics
2. Adobe Analytics
3. Custom made tools

Google Analytics vs Adobe Analytics

- Pricing:
  - Google Analytics provides a lot of free features for its users, but Adobe Analytics is a paid tool. Google also offers paid Analytics 360 with more features, which remains less expensive than Adobe Analytics.
- Real time reporting:
  - Adobe Analytics does better real-time reporting than Google Analytics. Real-time data analysis is crucial for identifying trends and optimizing marketing efforts. Adobe provides faster, more advanced visualization of real-time data and lets you retain it for a lifetime. Google Analytics allows retention for 24 months only.

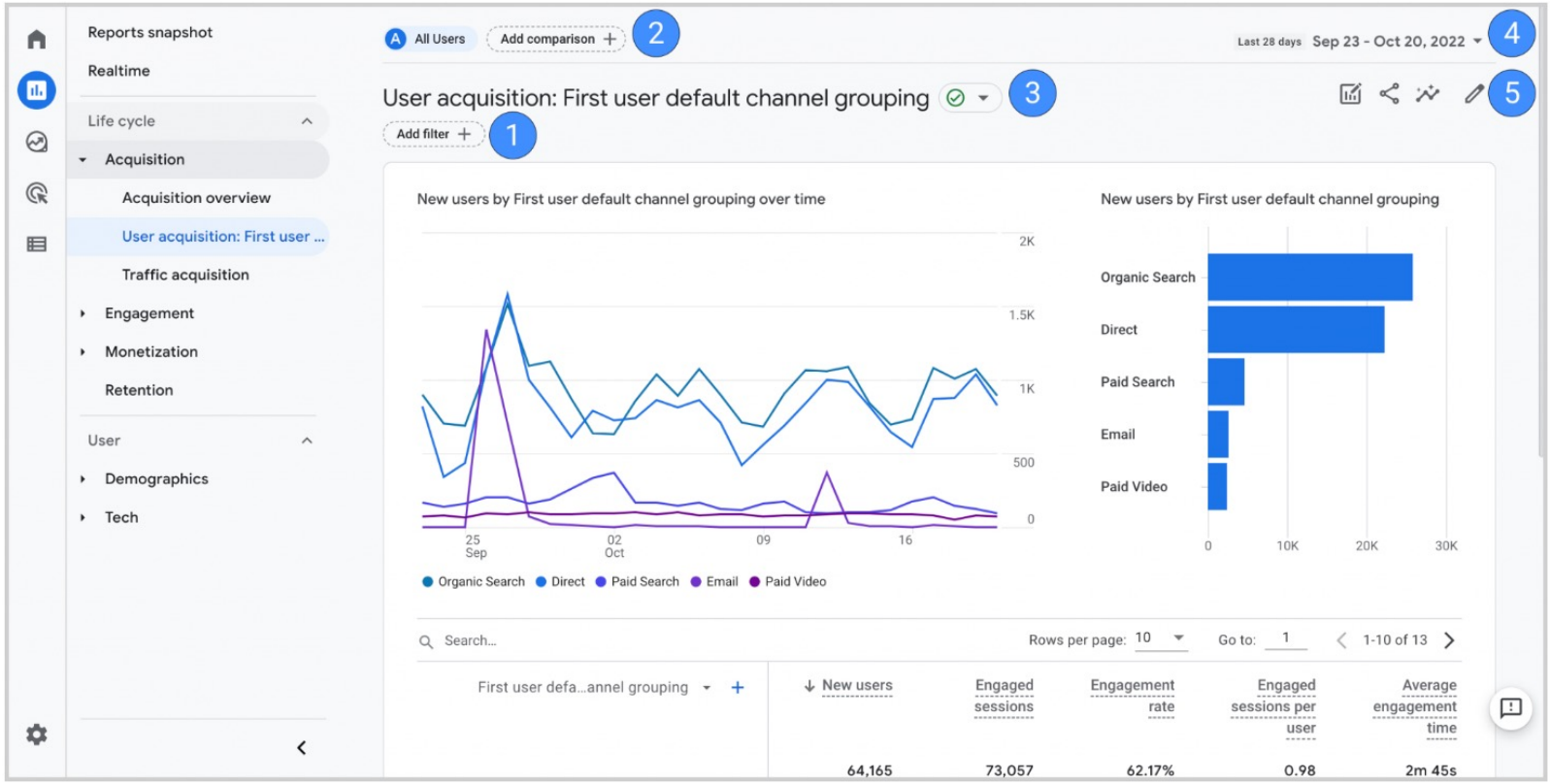
https://salespanel.io/resources/google-analytics-vs-adobe-analytics/

Google Analytics vs Adobe Analytics

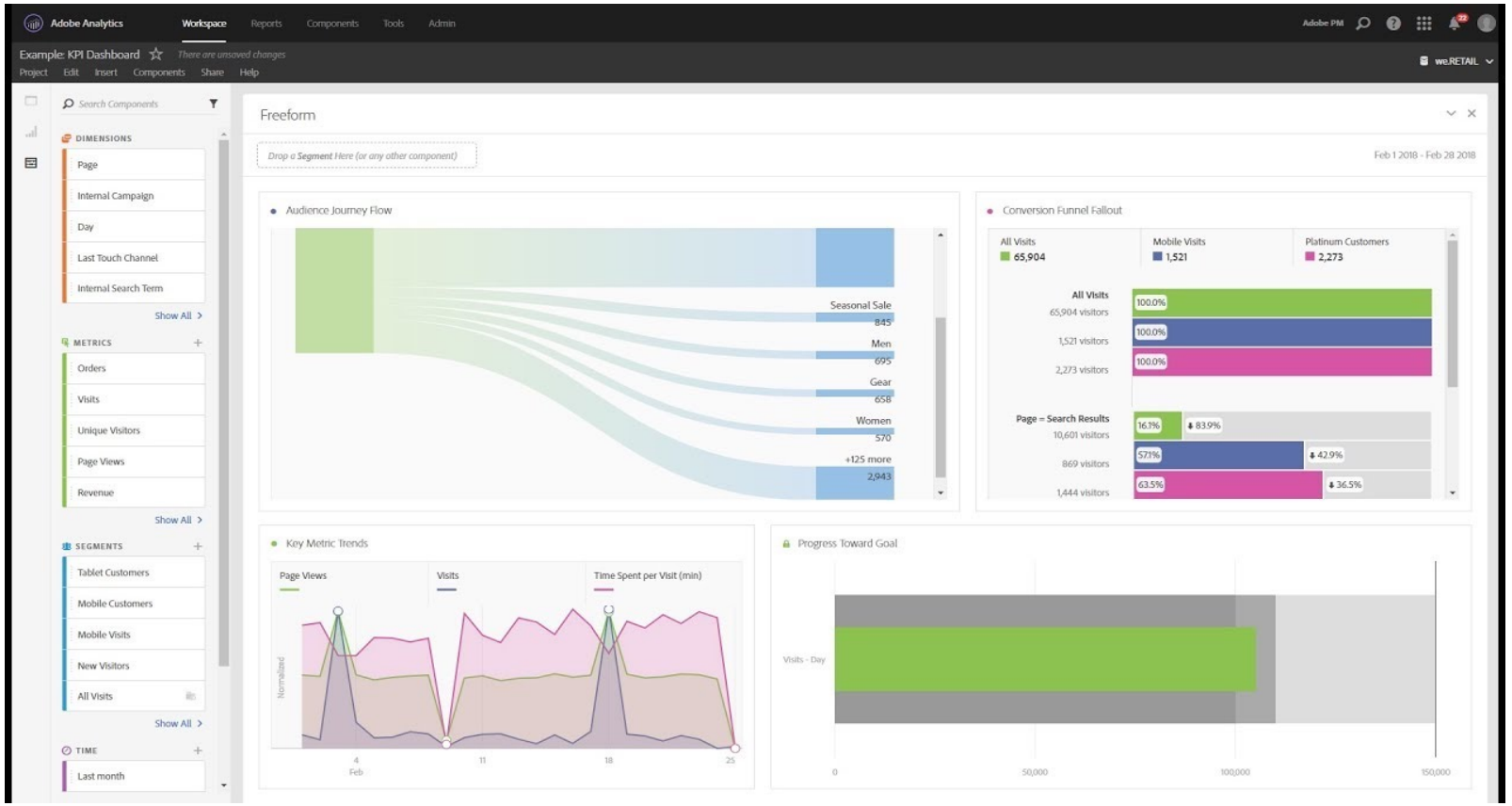
- Segmentation and Personalization:
  - By segmenting the right audience, a personalized experience can be tailored for website visitors. You can segment the audience on various criteria like demographics, behavior, and purchasing pattern for more targeted marketing. Adobe Analytics provides advanced segmentation features for better personalization compared to Google Analytics.
- User Interface and Navigation:
  - Google Analytics has a user-friendly interface for easier analysis of data. Adobe Analytics has a more complex user interface so a new users will require more training to analyze data effectively.

https://salespanel.io/resources/google-analytics-vs-adobe-analytics/

Google Analytics

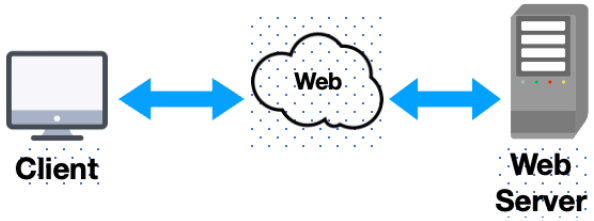


Adobe Analytics



Web Analytics Data Sources

- Server log files
  - Text documents that contain all activities of a server over a period of time. Web log analysis software uses these files to provide information about how, when, and by whom a web server is visited.
  - Standard and easy to access
  - Includes search engine access info
- Page Tagging
  - Cookies track mouse events, movie plays, …
  - Inexpensive access to visitor data
  - Click analysis
  - Visitor Geolocation analysis



Sample Server Log

- Each line from the server log shows the originating IP address, the date and time, the page called and, where available, the link origin.

65.26.xxx.xxx - - [04/Nov/2002:01:51:53 +0000] "GET /ivsat.htm HTTP/1.1" 200 9430 "http://search.dogpile.com/texis/search?q=Satellite+Internet+Access+Dish&format=clone&brand=dogpile&attrib=rs" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"

65.26.xxx.xxx - - [04/Nov/2002:01:51:53 +0000] "GET /901-342s.jpg HTTP/1.1" 200 8600 "https://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 65.26.149.185 - - [04/Nov/2002:01:51:54 +0000] "GET /pas1rkuh.gif HTTP/1.1" 200 4189 "https://www.satsig.net/ivsat.htm" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"

http://www.satsig.net/logfile.htm

Sample HTML File

```
<html>
<!-- Text between angle brackets is an HTML tag and is not displayed.
Most tags, such as the HTML and /HTML tags that surround the contents of
a page, come in pairs; some tags, like HR, for a horizontal rule, stand
alone. Comments, such as the text you're reading, are not displayed when
the Web page is shown. The information between the HEAD and /HEAD tags is
not displayed. The information between the BODY and /BODY tags is displayed.-->
<head>
    <title>Enter a title, displayed at the top of the window.</title>
</head>
<!-- The information between the BODY and /BODY tags is displayed.-->
<body>
    <h1>Enter the main heading, usually the same as the title.</h1>
        <p>Be <b>bold</b> in stating your key points. Put them in a list: </p>
    <ul>
        <li>The first item in your list</li>
        <li>The second item; <i>italicize</i> key words</li>
    </ul>
    <p>Improve your image by including an image. </p>
    <p><img src="http://www.mygifs.com/CoverImage.gif" alt="A Great HTML Resource"></p>
    <p>Add a link to your favorite <a href="https://www.dummies.com/">Web site</a>.
        Break up your page with a horizontal rule or two. </p>
    <hr>
    <p>Finally, link to <a href="page2.html">another page</a> in your own Web site.</p>
    <!-- And add a copyright notice.-->
    <p>&#169; Wiley Publishing, 2011</p>
</body>
</html>
```

source: https://www.dummies.com/web-design-development/site-development/a-sample-web-page-in-html/

Web Scraping Applications

- Web scrapping can read pages, that is hard for human to read and summarize.
  - Example: 'cheapest flight from X to Y.'
- Web Scrapping is so useful that some web sites provide their own API (Application Programable Interface) for scrapping their data such as Twitter API, Instagram API, Youtube API, WikiPedia API, …
- Many applications benefit from web scrapping
  - Market forecasting and market studies, like scrapping online product review from Amazon, identifying public opinion about a corporation from twitter, what are trends on crypto currencies, etc.
  - Machine-language translation, like using web text as a template to reconstruct a sentence correctly .
  - Medical diagnostics (retrieve and analyze data from news sites, translated texts, and health forums), like reading health forums to identify a symptom of a drug in large scale.

Web scraping basic steps

1. Retrieving HTML data from a domain name
2. Parsing that data for target information
3. Storing the target information
4. Optionally, moving to another page to repeat the process

Some Available Data Examples

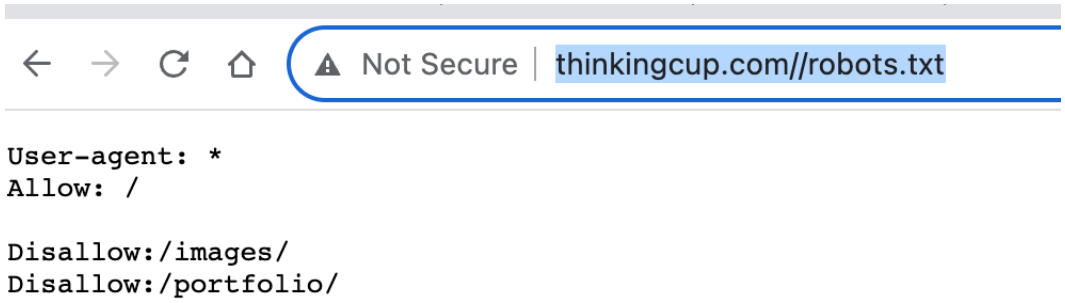
- Crowdfunding
  - Indiegogo https://webrobots.io/indiegogo-dataset/
  - Kickstarter https://webrobots.io/kickstarter-datasets/
- Academia
  - Pubmed: PubMed® comprises more than 36 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full text content from PubMed Central and publisher web sites. https://www.nlm.nih.gov/databases/download/pubmed_medline.html
  - DBLP: (Digital Bibliography & Library Project) dataset is a comprehensive bibliographic database of computer science research papers and proceedings. https://dblp.uni-trier.de/xml/

Scraping Example

- How to find out the website allows Scraping:

  Take the root of the url and add '/robots.txt' and enter it to the browser

  http://thinkingcup.com/robots.txt

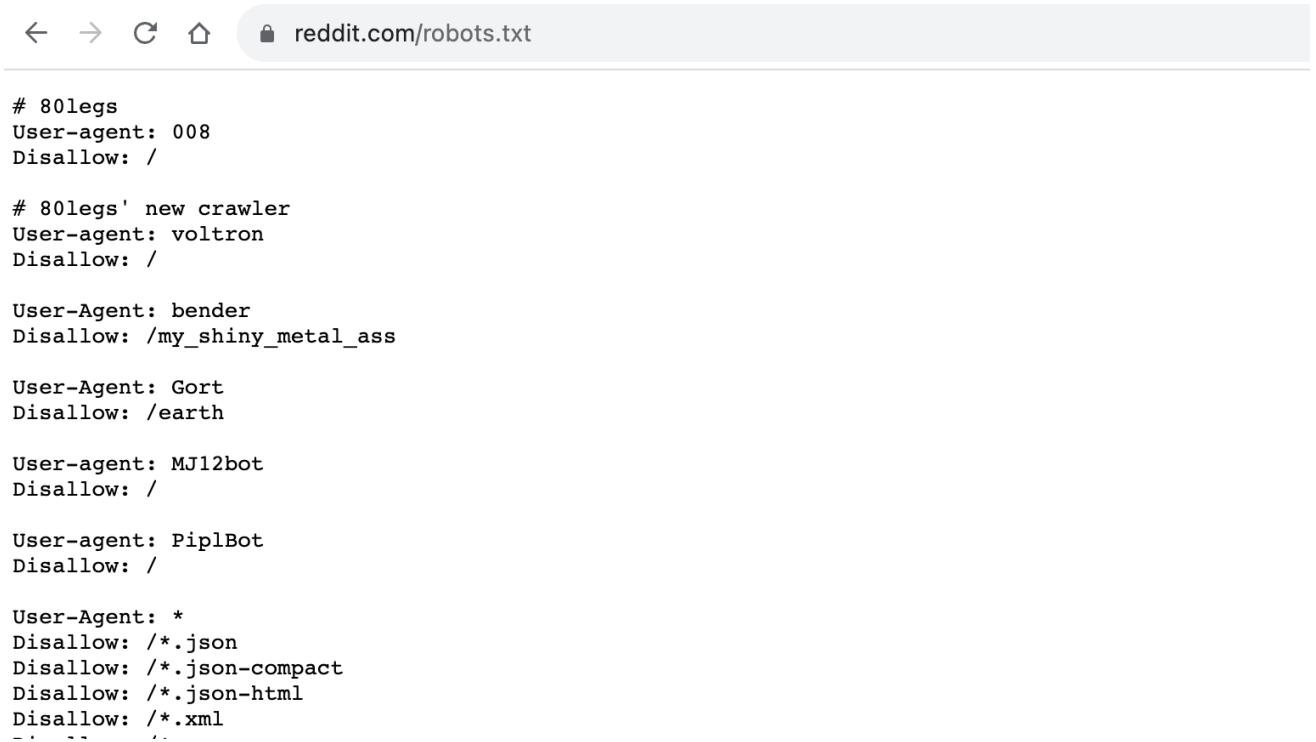


What is a user agent in crawling?

- It’s a string which helps the destination server identify which browser, device and operating system is being used.
- "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36,gzip(gfe)”

Scraping Example (few months ago)

- https://www.reddit.com/robots.txt



Scraping Example (now)

- https://www.reddit.com/robots.txt

robots.txt

- robots.txt is the filename used for implementing the **Robots Exclusion Protocol**, a standard used by websites to indicate to visiting web crawlers and other web robots which portions of the website they are allowed to visit.
- This relies on voluntary compliance. Not all robots comply with the standard; email harvesters, spambots, malware and robots that scan for security vulnerabilities may even start with the portions of the website where they have been told to stay out.

https://en.wikipedia.org/wiki/Robots.txt

References

Müller, Andreas C.; Guido, Sarah. Introduction to Machine Learning with Python. O'Reilly Media. Kindle Edition.
Mitchell, Ryan. Web Scraping with Python. O'Reilly Media. Kindle Edition.

[Hey ’09] Hey, T., Tansley, S., & Tolle, K. M. (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery (Vol. 1). Redmond, WA: Microsoft Research.

[Mitchell ’97] Mitchell, T.M. (1997) Machine Learning. McGraw Hill.

[Géron ‘17] Géron, A. (2017). Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly.

[Provost ‘13] Provost, F., & Fawcett, T. (2013). Data Science for Business: What you need to know about data mining and data-analytic thinking. " O'Reilly Media, Inc.".

[Russell ’09] Russell, S., Norvig, P. (2019). Artificial intelligence: A Modern Approach (4th Edition). Prentice Hall.

[Chapmann ’17] Chapmann, J. (2017) Machine Learning: Fundamental Algorithms for Supervised and Unsupervised Learning With Real-World Applications.

[Knuth ‘97] Knuth, D. E. (1997). The Art of Computer Programming (Vol. 1). Pearson Education.

[Bhargava ‘16] Bhargava, A. (2016). Grokking Algorithms: An illustrated guide for programmers and other curious people. Manning Publications Co.

[Han ‘11] Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. 3rd Edition. Elsevier.