IEEE Student Branch
University of Boumerdes

ML BOOTCAMP

Let's start!

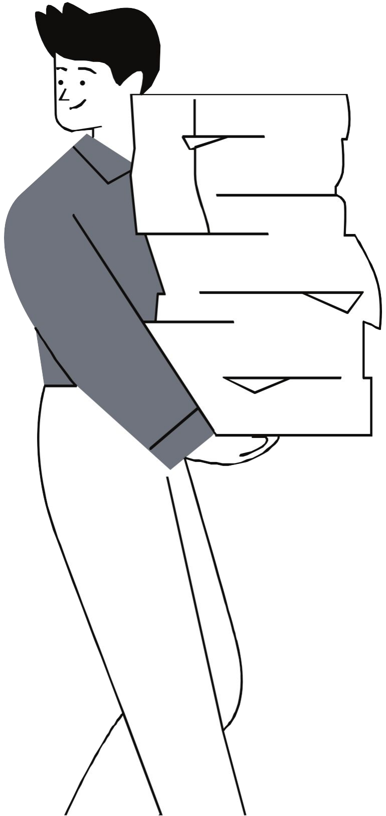
Program of the day

1. Mind refreshing
2. data cleaning and analysis
3. algorithms
4. git and github

ML
BOOTCAMP

Mind Refreshing

small review about machine learning and our previous workshop activities and projects



IEEE Student Branch
University of Boumerdes

types of ML

The most used types are supervised and unsupervised learning

because they are much easier to use and handle and don't take a very long time, unlike reinforcement learning.

SUPERVISED LEARNING

UNSUPERVISED LEARNING

steps to implement ML

Basically they are 6 steps

you don't have to remember them just understand the way they work and the link between them

✓ **1** importing data

2 ✗ cleaning data

✓ **3** split the data

4 ✗ choose the algorithm

✓ **5** evaluate the model

6 ✗ hyper parameter tuning

Durée : 5 minutes

Data cleaning

1 data cleaning

2 data visualization

3 data preprocessing

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted.

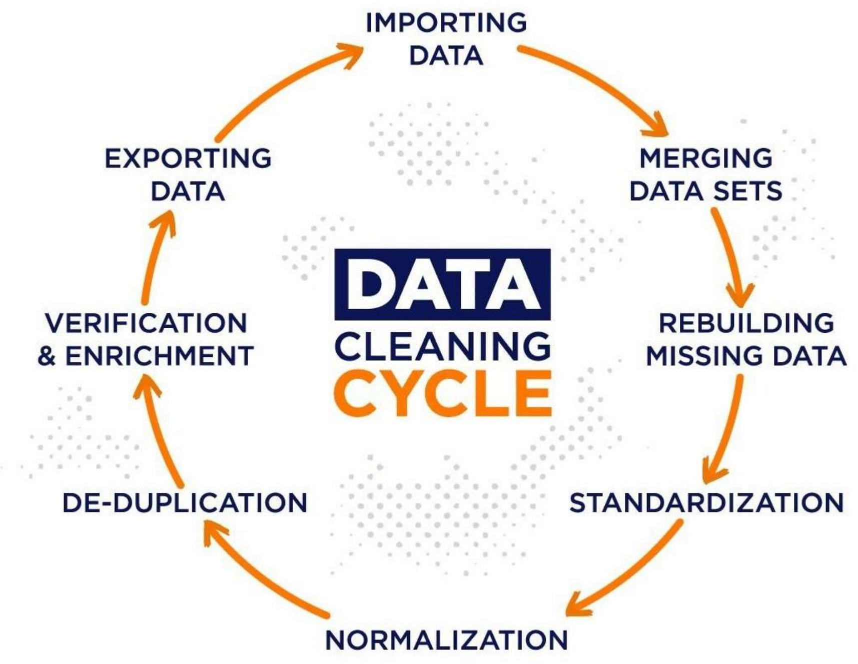
Data cleaning

1 data analysis

2 data visualization

3 data preprocessing

By following this schematic circle you will get the cleanest data

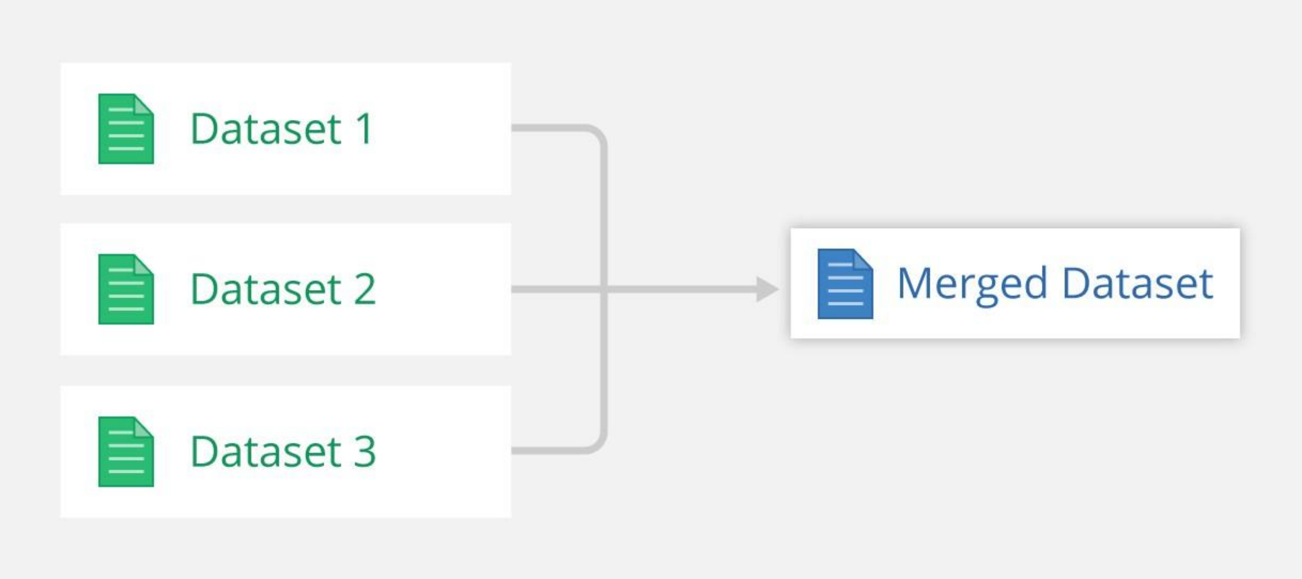


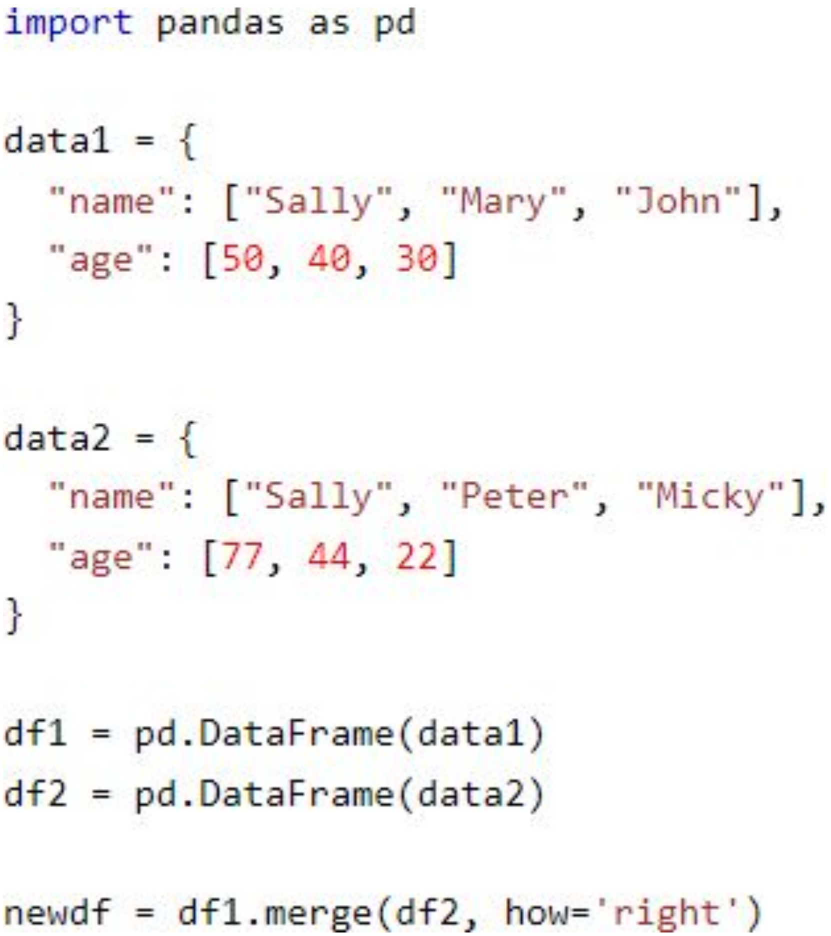
data merging!

what is a merged dataset?

Data merging is the process of combining two or more data sets into a single data set.

To merge two data frames (datasets) horizontally, use the merge function.





```
import pandas as pd

data1 = {
  "name": ["Sally", "Mary", "John"],
  "age": [50, 40, 30]
}

data2 = {
  "name": ["Sally", "Peter", "Micky"],
  "age": [77, 44, 22]
}

df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)

newdf = df1.merge(df2, how='right')
```

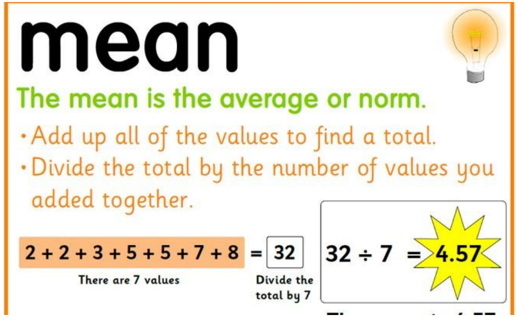
Rebuilding Messing Data

Liste du projet

Now let’s look at the different methods that you can use to deal with the missing data. The methods I will be discussing are

1. Deleting the columns with missing data
2. – Imputation
3. Filling with a Regression Model

filling with the mean value



mean

The mean is the average or norm.

- Add up all of the values to find a total.
- Divide the total by the number of values you added together.

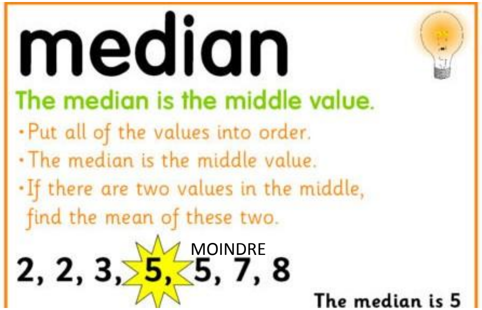
2 + 2 + 3 + 5 + 5 + 7 + 8 = 32

There are 7 values

Divide the total by 7

32 ÷ 7 = 4.57

filling with the median value



median

The median is the middle value.

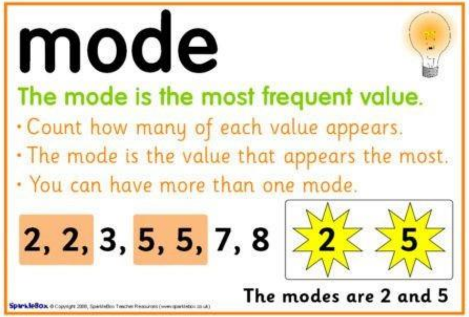
- Put all of the values into order.
- The median is the middle value.
- If there are two values in the middle, find the mean of these two.

MOINDRE

2, 2, 3, 5, 5, 7, 8

The median is 5

filling with the mode value



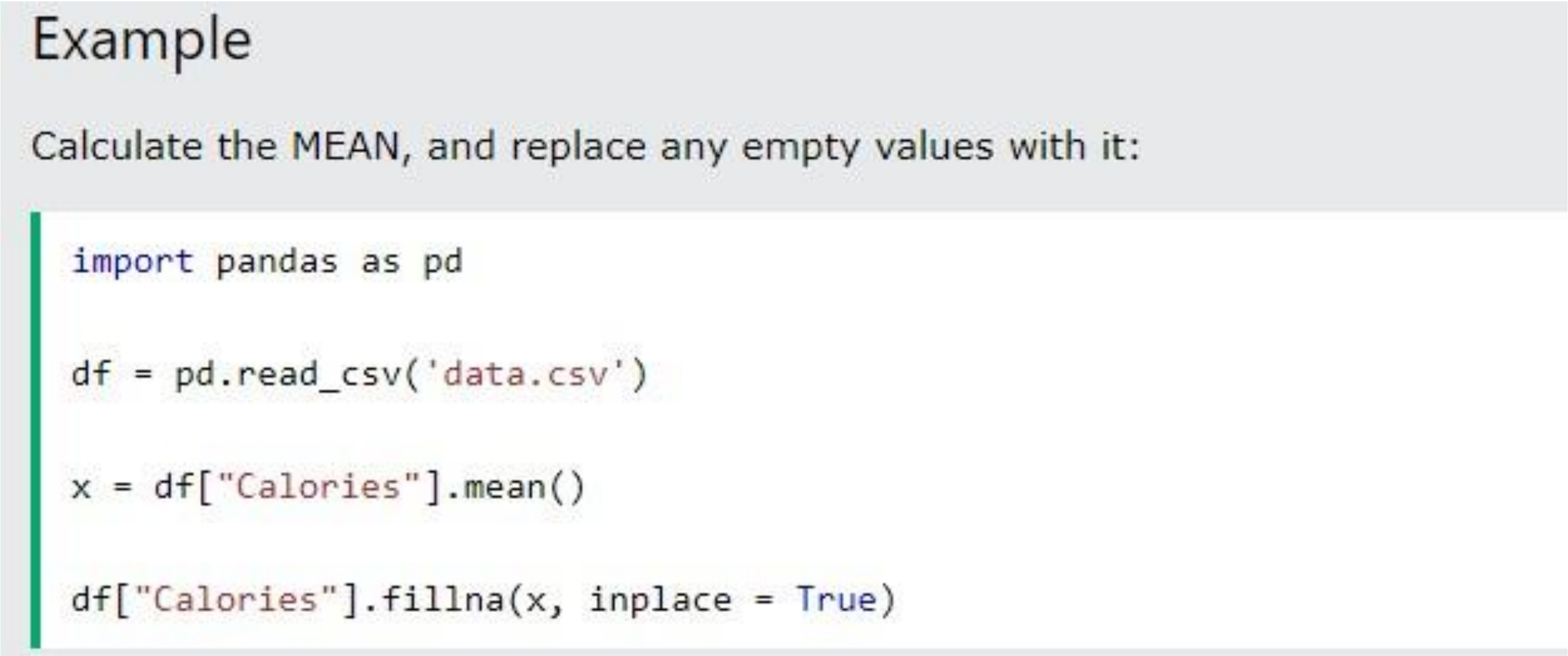
mode

The mode is the most frequent value.

- Count how many of each value appears.
- The mode is the value that appears the most.
- You can have more than one mode.

2, 2, 3, 5, 5, 7, 8 2 5

The modes are 2 and 5



Example

Calculate the MEAN, and replace any empty values with it:

```
import pandas as pd

df = pd.read_csv('data.csv')

x = df["Calories"].mean()

df["Calories"].fillna(x, inplace = True)
```

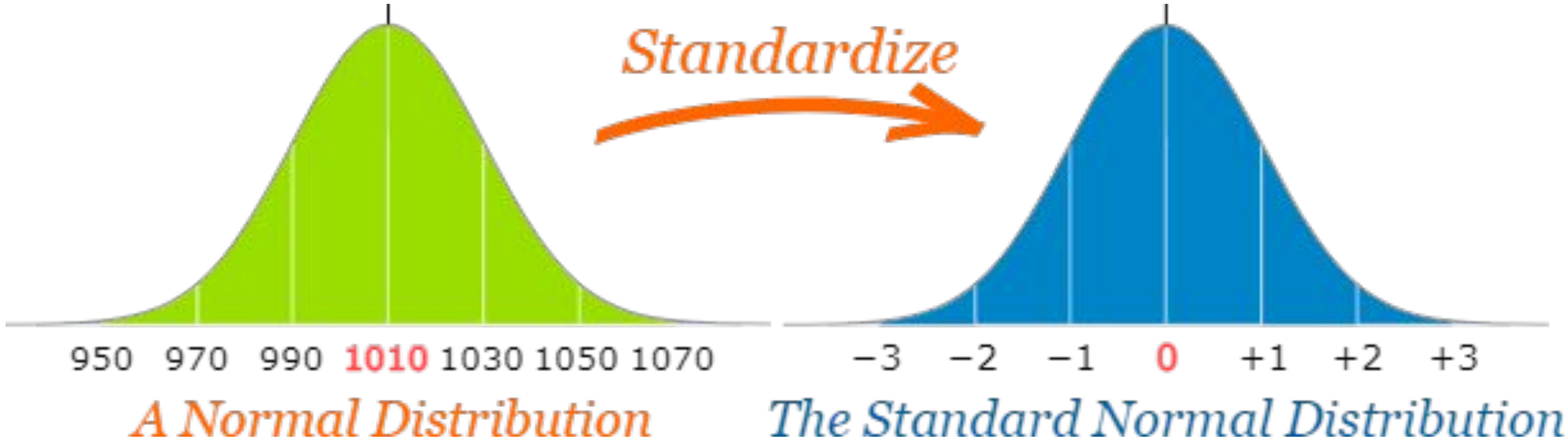
Standardization

Data standardization is the process of converting data to a common format to enable users to process and analyze it. Most organizations utilize data from a number of sources; this can include data warehouses, lakes, cloud storage, and databases.

1. We can do standarization using z_score function

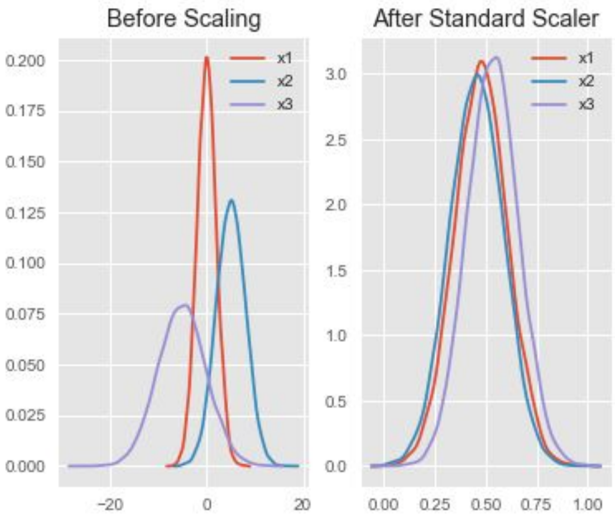
2. This is the relation behind it

$$Z = \frac{\overset{\text{Score}}{x} - \overset{\text{Mean}}{\mu}}{\underset{\text{SD}}{\sigma}}$$


Standardize
950
970
990
1010
1030
1050
1070
A Normal Distribution
−3
−2
−1
0
+1
+2
+3
The Standard Normal Distribution

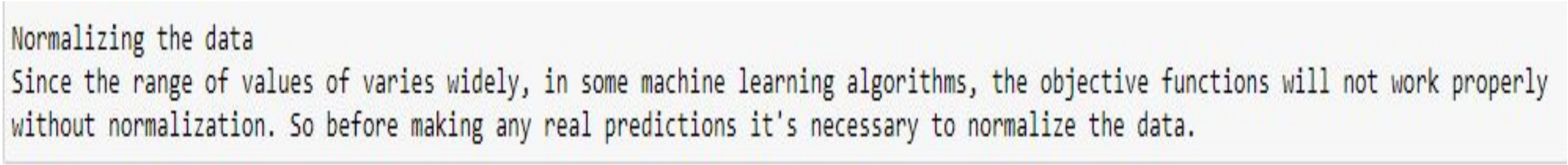
Normalization

- We can use some python functions to do this step like:
- Sctanderd scaling



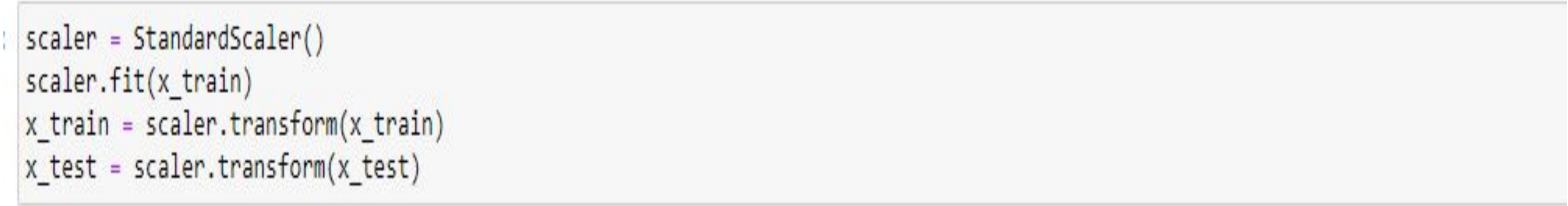
Normalization refers to rescaling real-valued numeric attributes into a 0 to 1 range. Data normalization is used in machine learning to make model training less sensitive to the scale of features.

Normalization



Normalizing the data

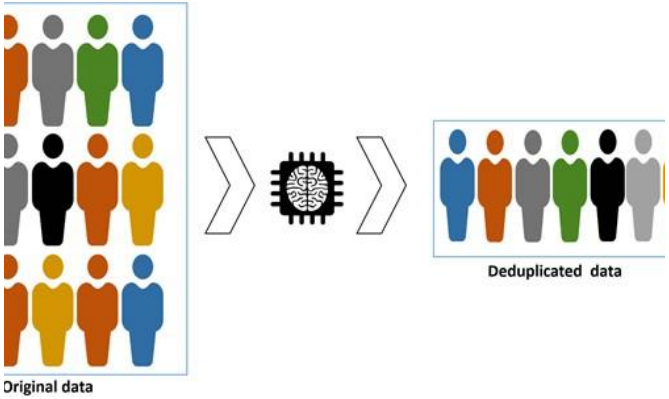
Since the range of values of varies widely, in some machine learning algorithms, the objective functions will not work properly without normalization. So before making any real predictions it's necessary to normalize the data.



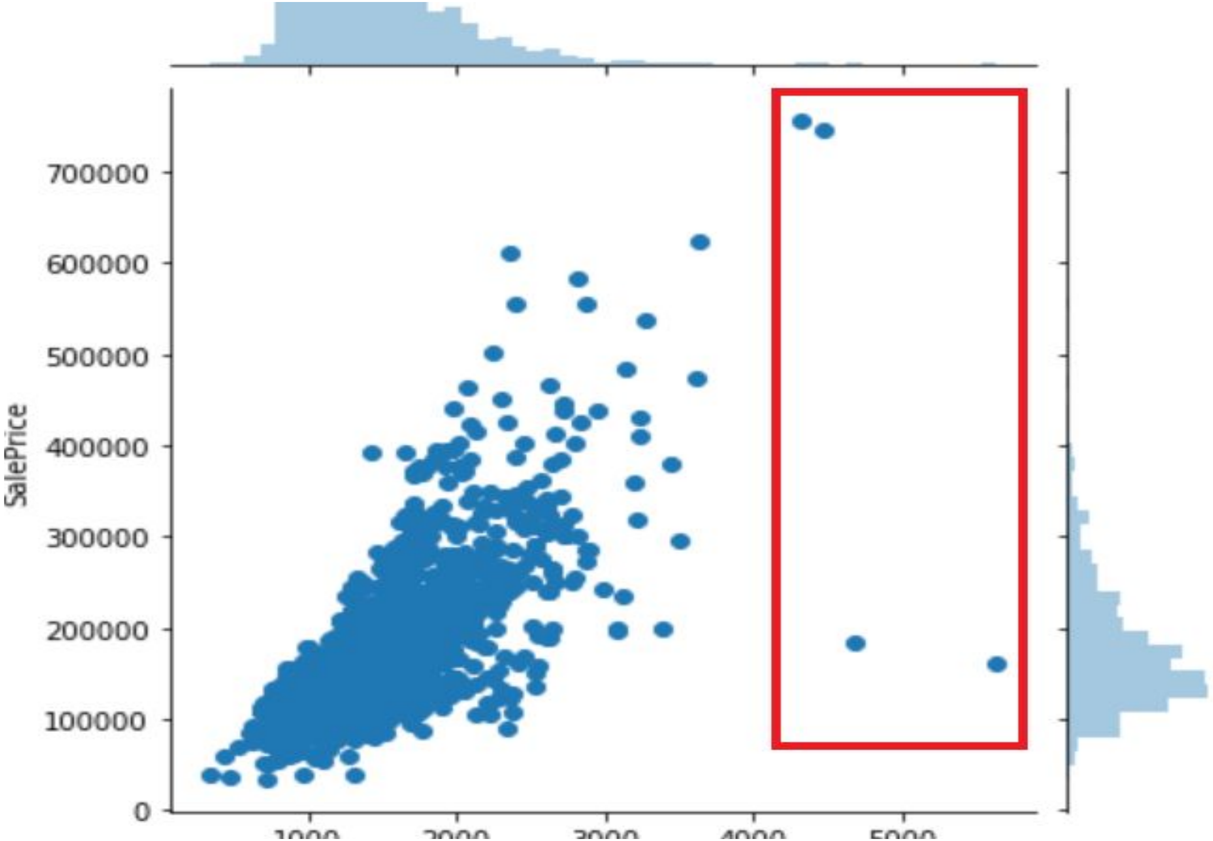
```
scaler = StandardScaler()
scaler.fit(x_train)
x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)
```

Deduplication

dedupe is a python library that uses machine learning to perform fuzzy matching, deduplication, and entity resolution quickly on structured data. dedupe will help you: remove duplicate entries from a spreadsheet of names and addresses.

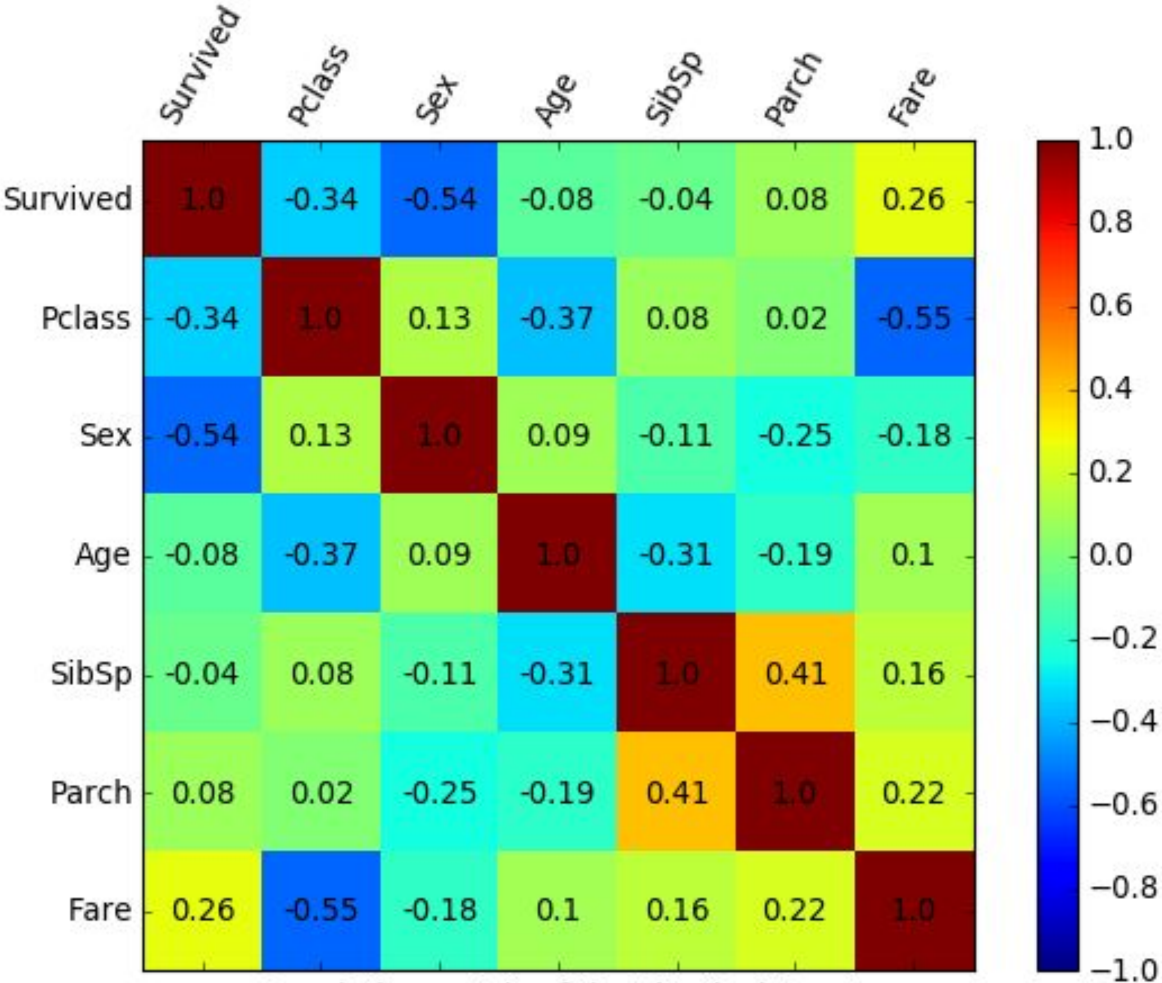


REMOVING OUTLIERS!



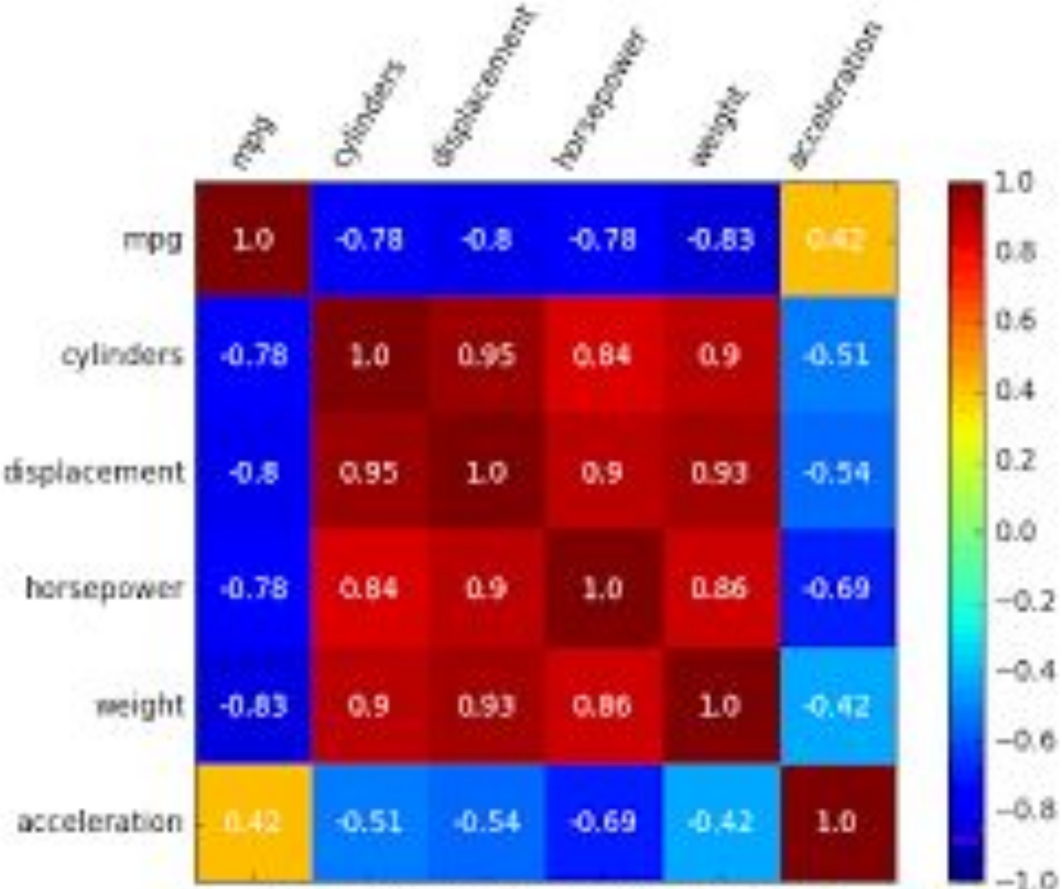
CORRELATION MATRIX

A correlation matrix is simply **a table which displays the correlation**. **It is best used in variables that demonstrate a linear relationship between each other**. coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table.



	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
Survived	1.0	-0.34	-0.54	-0.08	-0.04	0.08	0.26
Pclass	-0.34	1.0	0.13	-0.37	0.08	0.02	-0.55
Sex	-0.54	0.13	1.0	0.09	-0.11	-0.25	-0.18
Age	-0.08	-0.37	0.09	1.0	-0.31	-0.19	0.1
SibSp	-0.04	0.08	-0.11	-0.31	1.0	0.41	0.16
Parch	0.08	0.02	-0.25	-0.19	0.41	1.0	0.22
Fare	0.26	-0.55	-0.18	0.1	0.16	0.22	1.0

Correlation matrix of the Titanic dataset



	mpg	cylinders	displacement	horsepower	weight	acceleration
mpg	1.0	-0.78	-0.8	-0.78	-0.83	0.42
cylinders	-0.78	1.0	0.95	0.84	0.9	-0.51
displacement	-0.8	0.95	1.0	0.9	0.93	-0.54
horsepower	-0.78	0.84	0.9	1.0	0.86	-0.69
weight	-0.83	0.9	0.93	0.86	1.0	-0.42
acceleration	0.42	-0.51	-0.54	-0.69	-0.42	1.0

Correlation matrix of the Auto-MPG dataset

ALGORITHMS

We will continue our marathon through the classical algorithms , then we will mention the moderns ones with simple examples and showing the difference between them and some trick when we should use them

1 Naïve Bayes Classifier Algorithm

2 LOGISTIC REGRESSION

3 RANDOM FOREST

3 GRADIENT BOOST MACHINE

Naïve Bayes Classifier Algorithm

most effective Classification

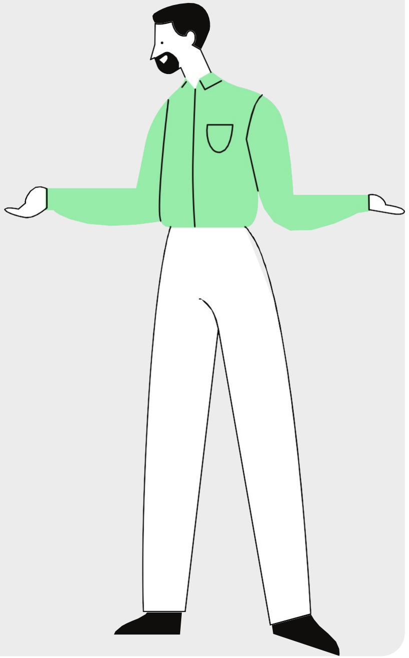
building the fast machine learning

can make quick predictions.

The naïve Bayes algorithm is a supervised learning algorithm, which is based on the **Bayes theorem** and is used for solving classification problems.

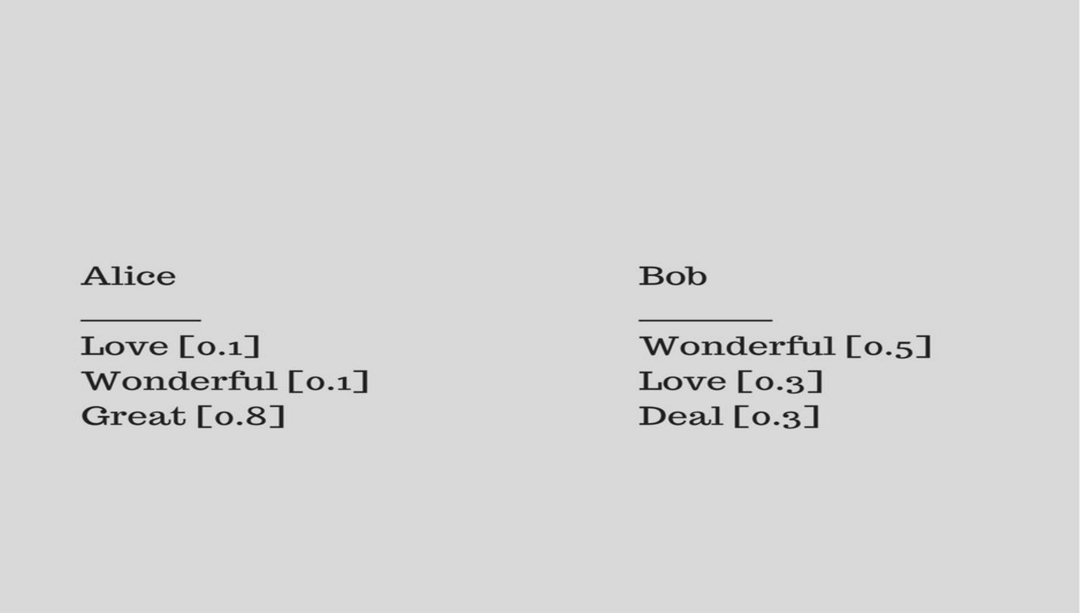
It is mainly used in text classification (Spam filtering and Sentiment analysis.)

It is one of the simple algorithms that helps in building fast machine learning models.



Naïve Bayes

Imagine two people Alice and Bob whose word usage pattern you know



Alice

Love [0.1]
Wonderful [0.1]
Great [0.8]

Bob

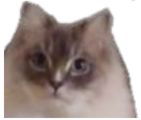
Wonderful [0.5]
Love [0.3]
Deal [0.3]

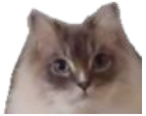
Now, can you guess who is the sender for the content : *“Wonderful Love.”* ?

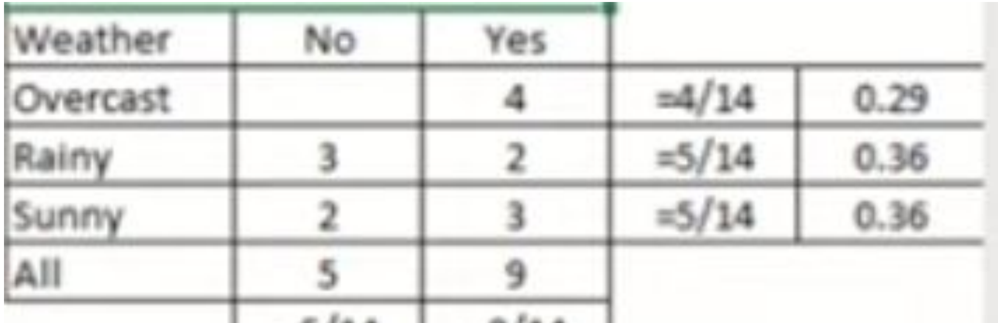
Bayes Theorem

It tells us how often A happens *given that B happens*, written **P(A|B)**, when we know how often B happens *given that A happens*, written **P(B|A)** , and how likely A and B are on their own.

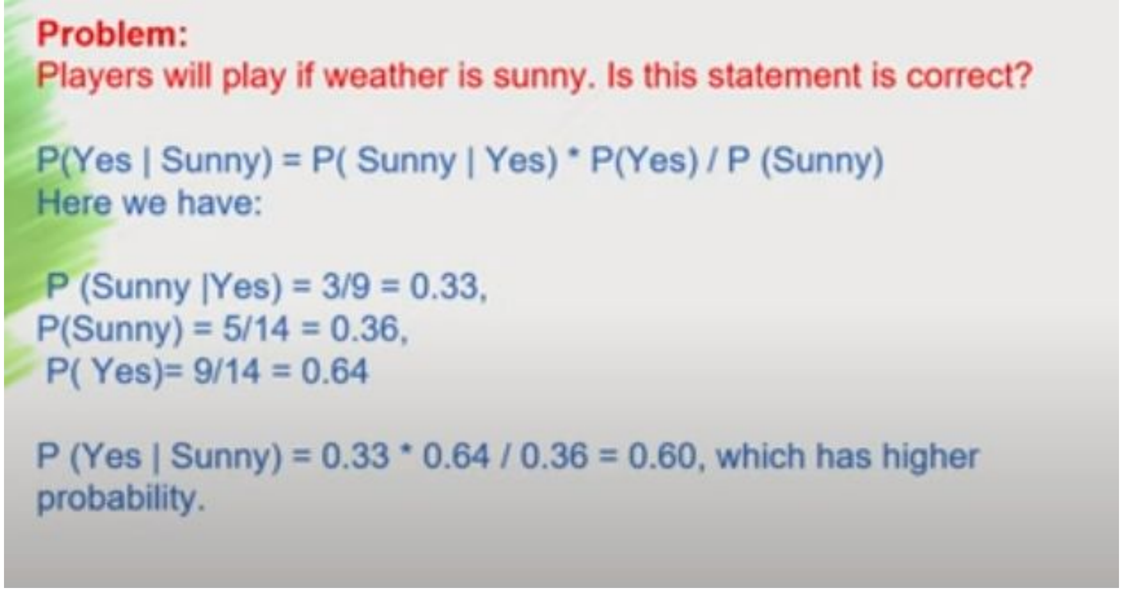
$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$







Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		



Problem:
Players will play if weather is sunny. Is this statement is correct?

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
Here we have:

P (Sunny |Yes) = 3/9 = 0.33,
P(Sunny) = 5/14 = 0.36,
P(Yes)= 9/14 = 0.64

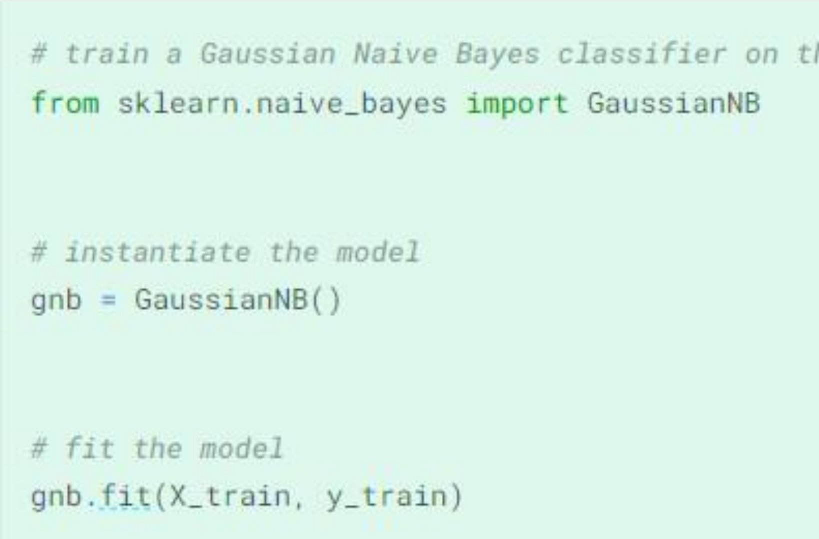
P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naïve Bayes Classifier Algorithm

an example of implementing naive Bayes algorithm

this is a gaussian naive Bayes algorithm

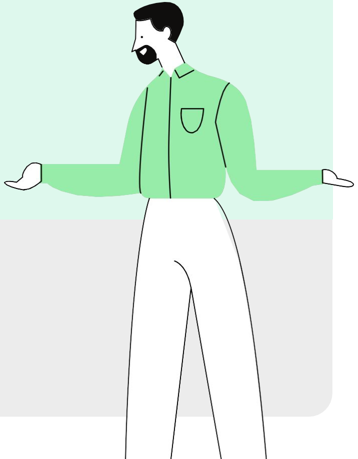
It is used in classification and it assumes that features follow a normal distribution



```python
# train a Gaussian Naive Bayes classifier on the training set
from sklearn.naive_bayes import GaussianNB


# instantiate the model
gnb = GaussianNB()


# fit the model
gnb.fit(X_train, y_train)
```



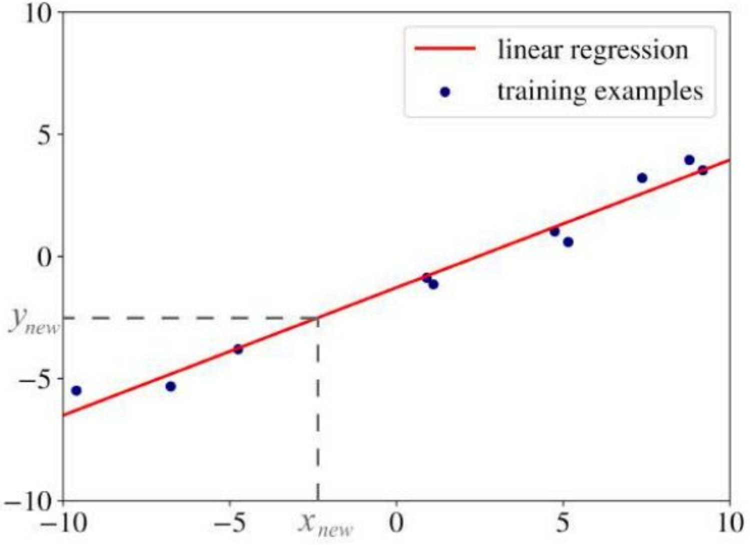
Disadvantages of Naïve Bayes

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Linear Regression

Algorithms that learns a model which represent the linear combination of input’s features

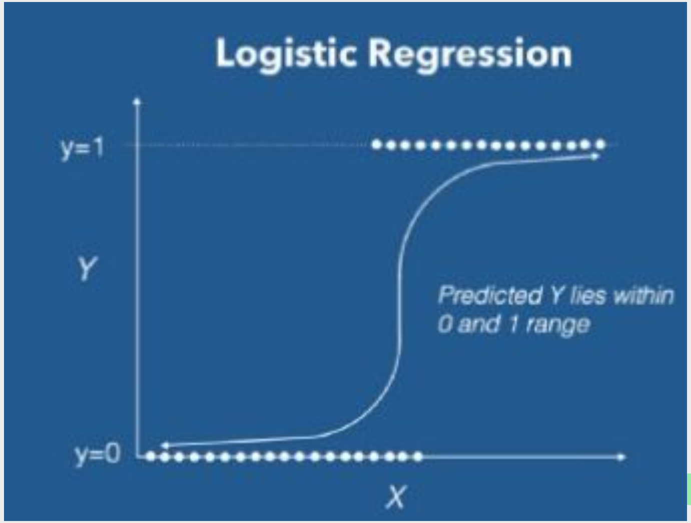
f(x)= wx+b

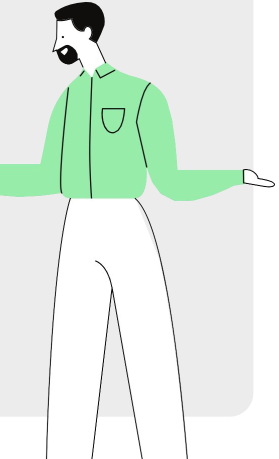


logistic regression

but is used to classify samples

f(w)= 1/ 1+e-(wx+b)





Logistic Regrassion

- **advantages:**

Logistic regression is easier to implement, interpret, and very efficient to train.

- **disadvantages:**

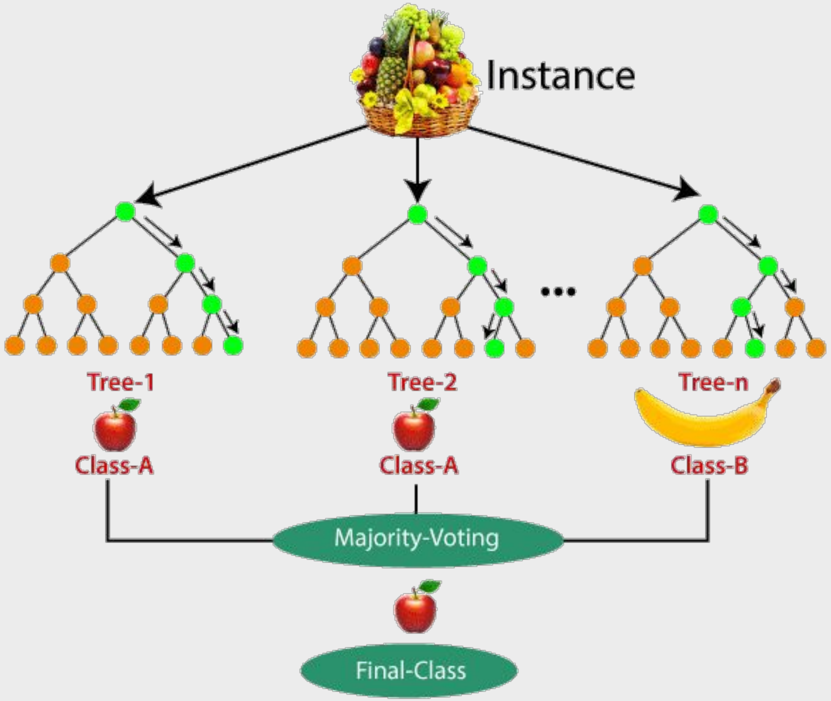
If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.

Random forest!

very powerful algorithm a popular machine learning algorithm that belongs to the supervised learning technique.

How does it work?

Random Forest can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.



What is ensemble learning

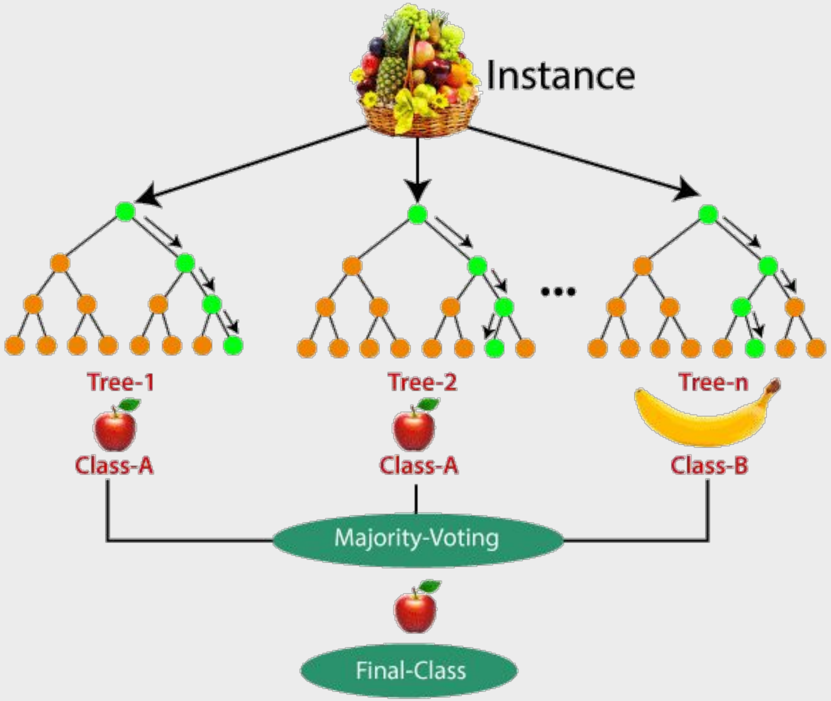
- « ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.. »

Random forest!

very powerful algorithm a popular machine learning algorithm that belongs to the supervised learning technique.

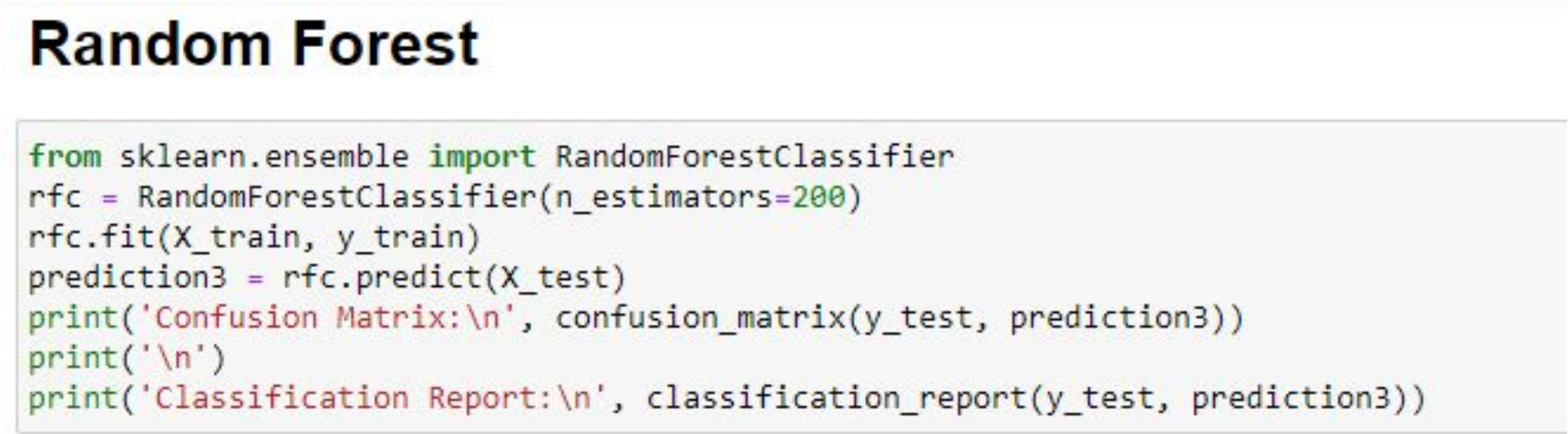
How does it work?

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.



It works in four steps

- Select random samples from a given dataset
- Construct a decision tree for each sample and get a prediction result from each decision tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction.



Random Forest

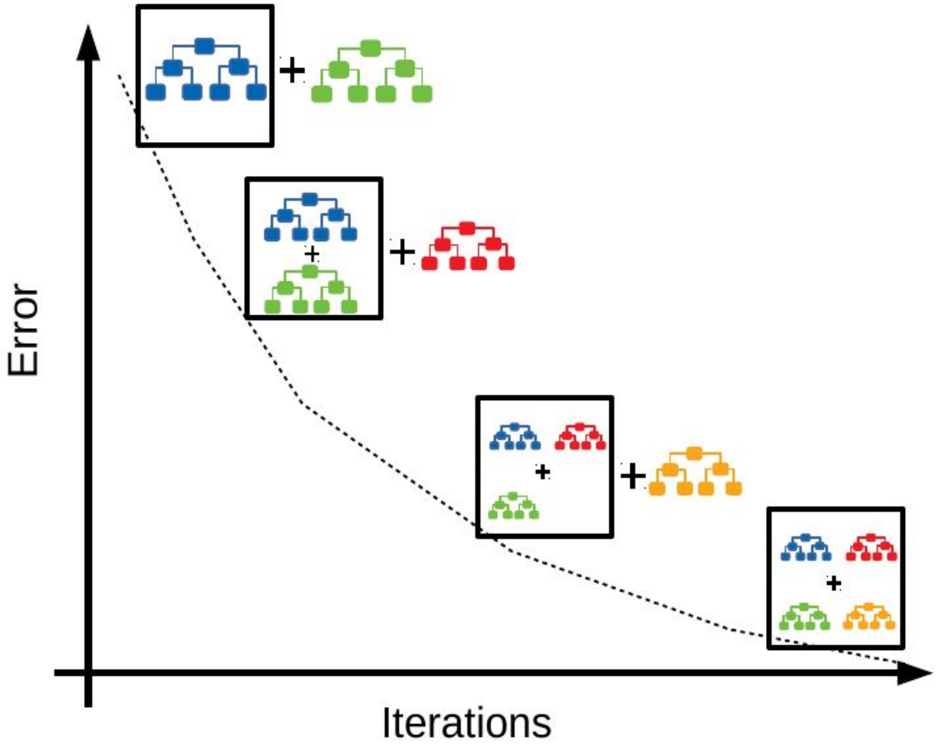
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
prediction3 = rfc.predict(X_test)
print('Confusion Matrix:\n', confusion_matrix(y_test, prediction3))
print('\n')
print('Classification Report:\n', classification_report(y_test, prediction3))
```

Gradient boost algrithm!

Called also gradient boost machine

How does it work?

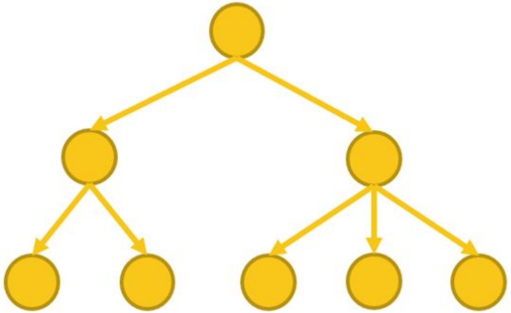
Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted tree



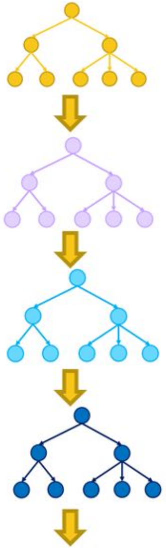
So....

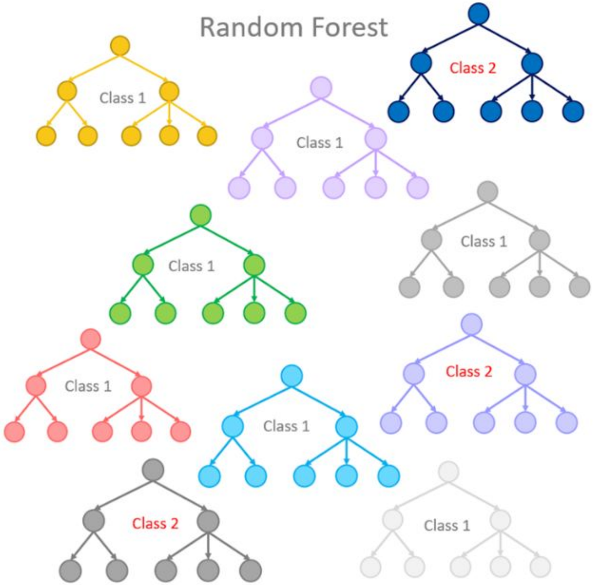
what is the difference between decision tree,random forest and and gbm

Single Decision Tree



Gradient Boosted Trees





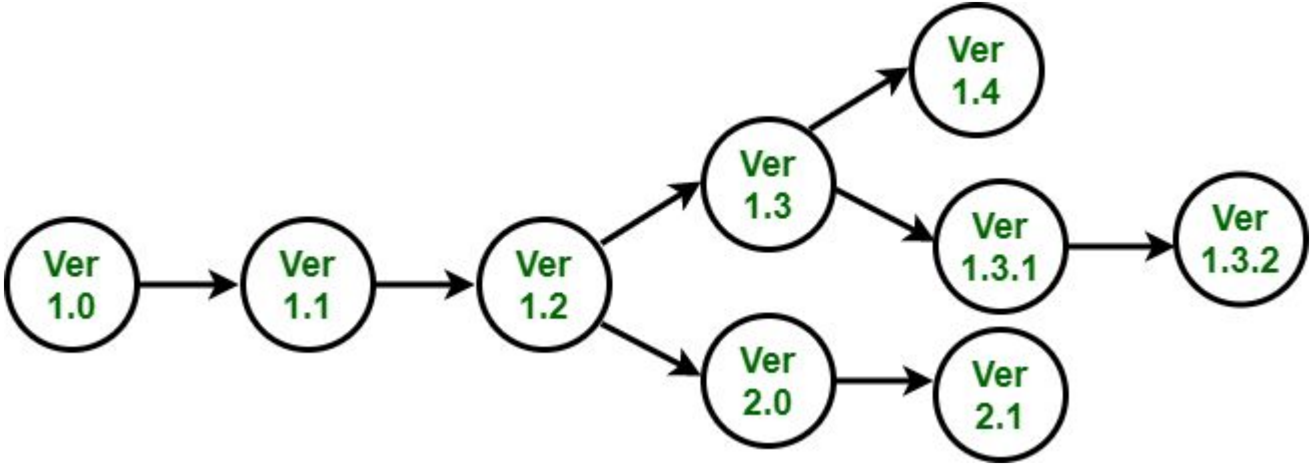
Random Forest

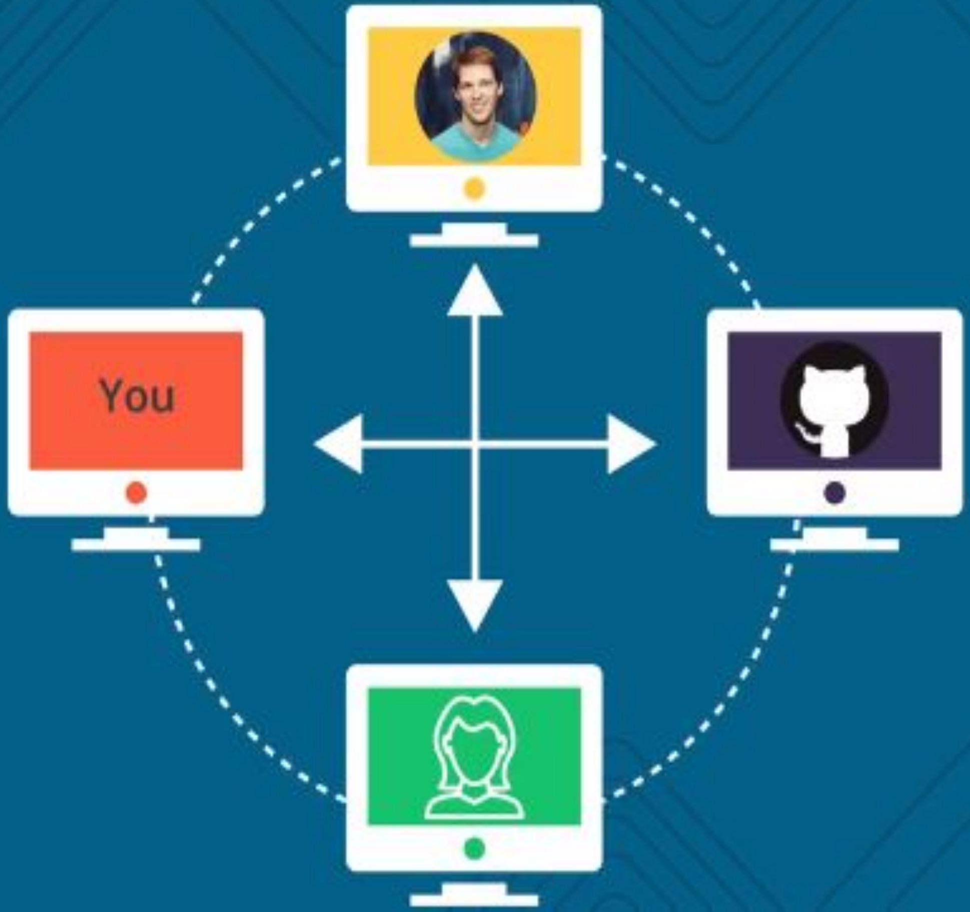
Now, let’s talk about

First things first

What Is Version Control System(VCS)?🤔

is the practice of tracking and managing changes to software code





git

Distributed Version Control System

What is Git?

- Git is a Version Control System (VCS) designed to make it easier to have multiple versions of a code base.

Created by Linus Torvalds, creator of linux in 2005

- It allows you to see changes you make to your code and easily revert them.
- Coordinates work between multiple developers
- local & remote repos
- Free and open source

Git Features

History:

Know exactly which files changed, who made those changes, and when those changes occured.

Backup:

Ability to have different versions of the code in different places.

Collaboration:

Collaborate easily with other people on the same project by uploading and receiving changes

What is GitHub?

- GitHub is a platform for code collaboration!
- GitHub uses Git for version control
- Hosting repositories on Github facilitates the sharing of codebases among teams by providing a GUI to easily fork or clone repos to a local machine

git

Version control tool

GitHub

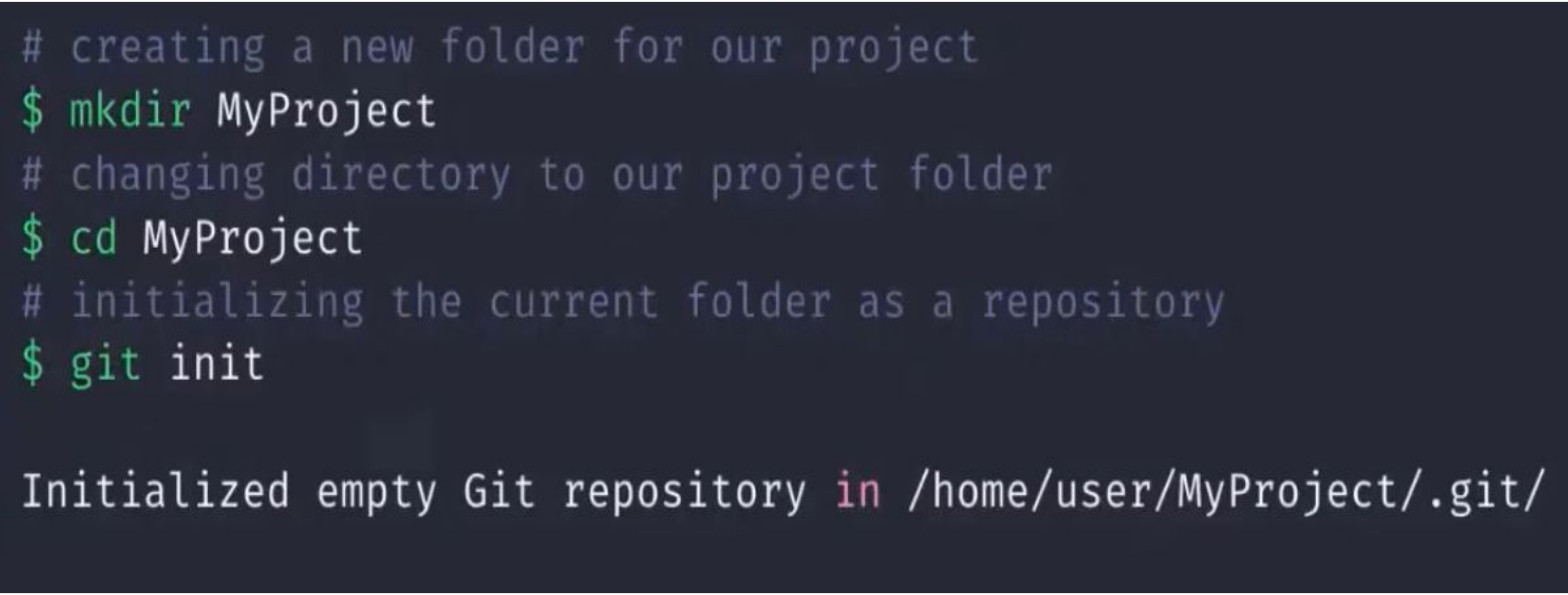
Service that hosts
Git projects

First Time Git Configuration

- Let’s define ourselves first
  - git config --global user.email "you@example.com"
  - git config --global user.name "Your Name"

Create a Git Repository

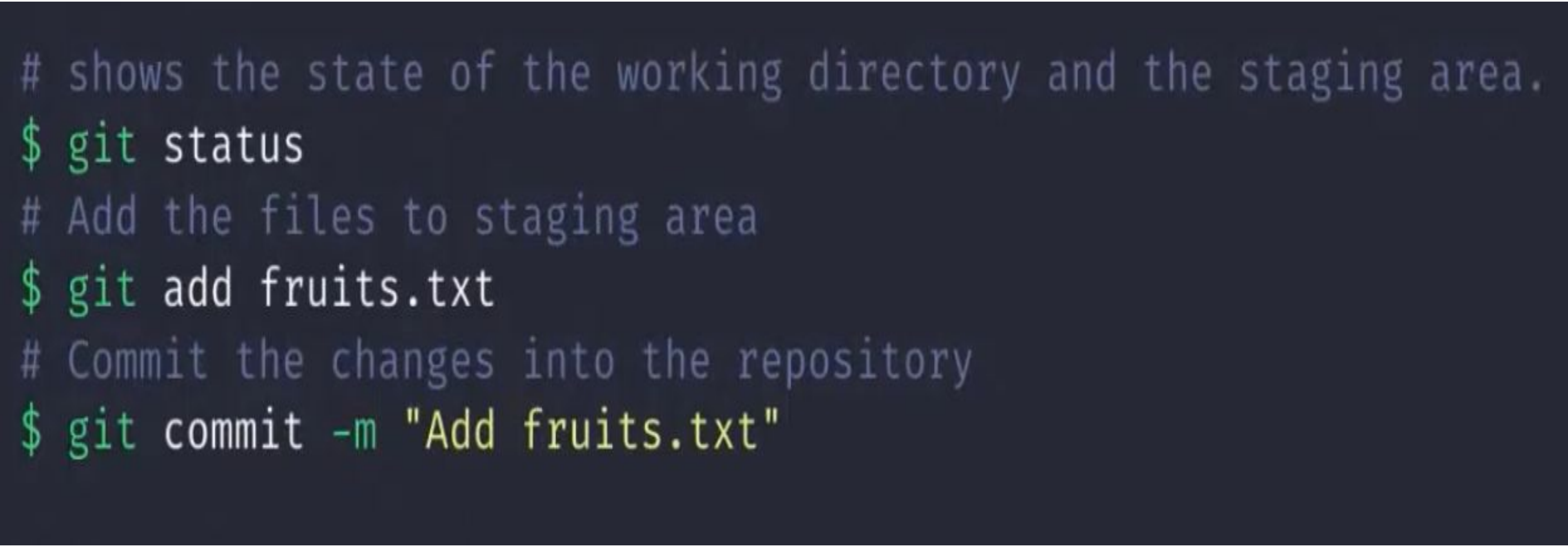
- first lets make a git folder in our computer : git init



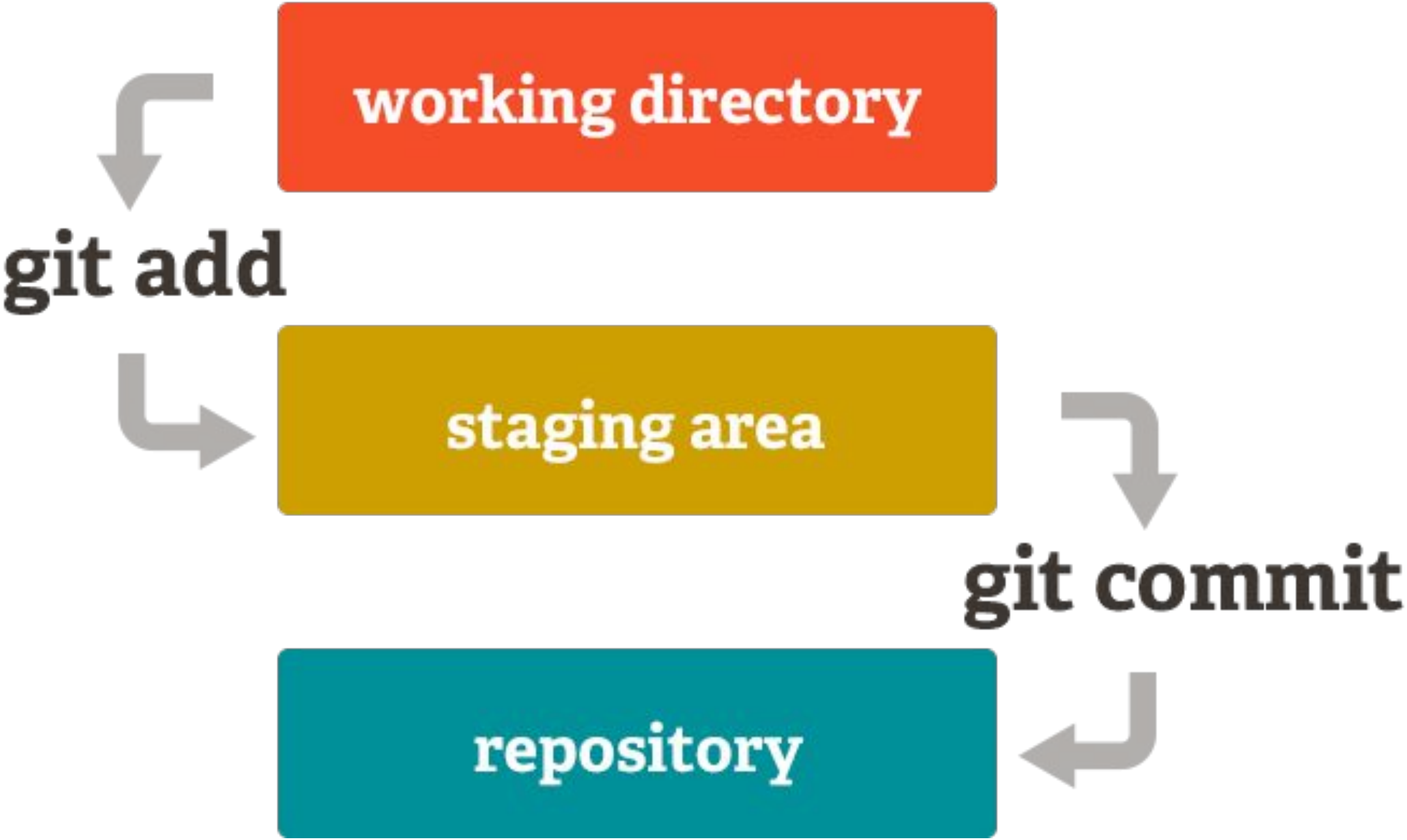
```
# creating a new folder for our project
$ mkdir MyProject
# changing directory to our project folder
$ cd MyProject
# initializing the current folder as a repository
$ git init

Initialized empty Git repository in /home/user/MyProject/.git/
```

What Next:



```
# shows the state of the working directory and the staging area.
$ git status
# Add the files to staging area
$ git add fruits.txt
# Commit the changes into the repository
$ git commit -m "Add fruits.txt"
```



working directory

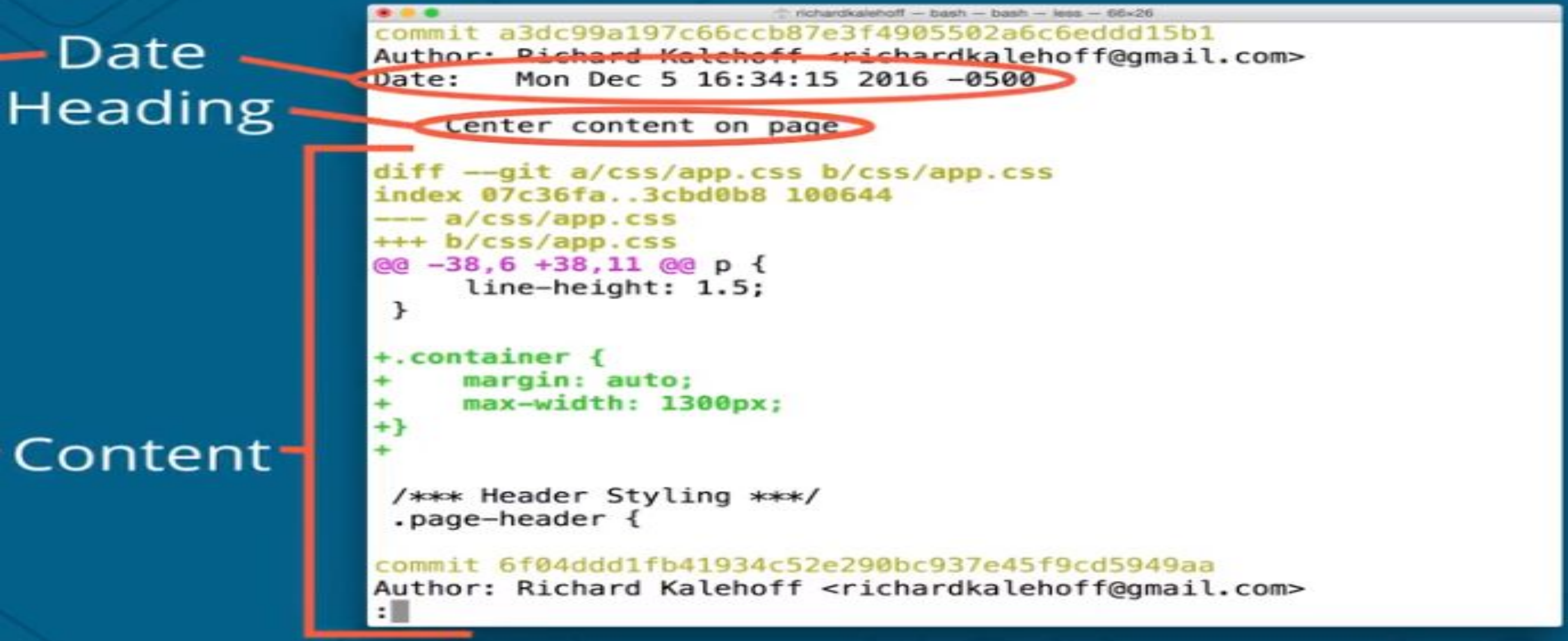
git add

staging area

git commit

repository

Commit



```
commit a3dc99a197c66ccb87e3f4905502a6c6eddd15b1
Author: Richard Kalehoff <richardkalehoff@gmail.com>
Date:   Mon Dec 5 16:34:15 2016 -0500

    Center content on page

diff --git a/css/app.css b/css/app.css
index 07c36fa..3cbd0b8 100644
--- a/css/app.css
+++ b/css/app.css
@@ -38,6 +38,11 @@ p {
     line-height: 1.5;
 }

+.container {
+    margin: auto;
+    max-width: 1300px;
+}
+

 /*** Header Styling ***/
 .page-header {

commit 6f04ddd1fb41934c52e290bc937e45f9cd5949aa
Author: Richard Kalehoff <richardkalehoff@gmail.com>
:
```

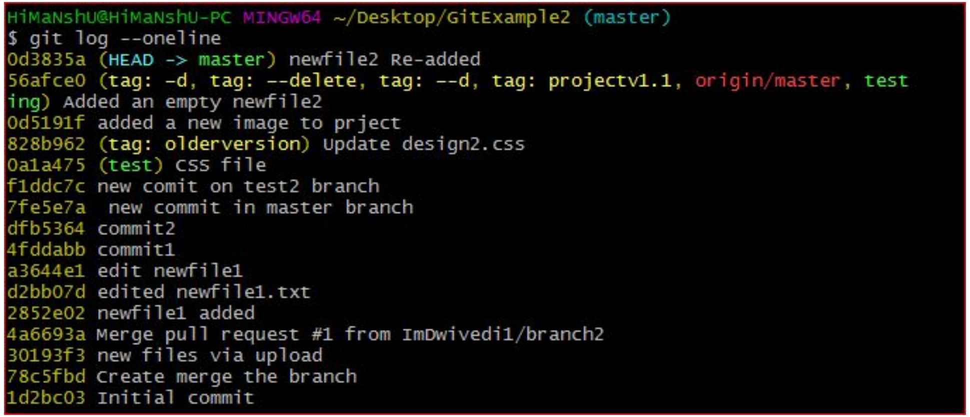
Date

Heading

Content

How to deal with commits

- **To inspect the history of commits (changes):** use the command ‘ git log –oneline ‘



```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git log --oneline
0d3835a (HEAD -> master) newfile2 Re-added
56afce0 (tag: -d, tag: --delete, tag: --d, tag: projectv1.1, origin/master, test
ing) Added an empty newfile2
0d5191f added a new image to prject
828b962 (tag: olderversion) Update design2.css
0a1a475 (test) CSS file
f1ddc7c new comit on test2 branch
7fe5e7a  new commit in master branch
dfb5364 commit2
4fddabb commit1
a3644e1 edit newfile1
d2bb07d edited newfile1.txt
2852e02 newfile1 added
4a6693a Merge pull request #1 from ImDwivedi1/branch2
30193f3 new files via upload
78c5fbd Create merge the branch
1d2bc03 Initial commit
```

- **To return back to previous state** : use the command ‘ git checkout <commitId> ‘

Gitignore file :

If you want to keep a file in your project's directory structure but make sure it isn't accidentally committed to the project, you can use the specially named file, .gitignore

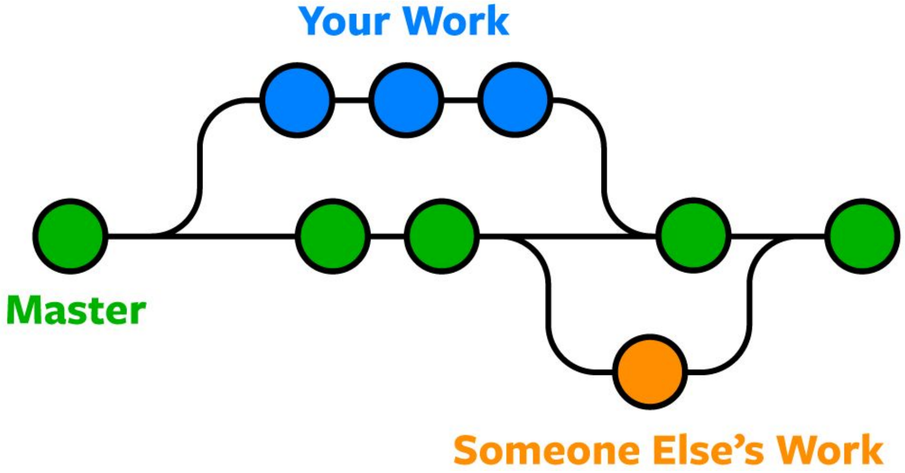
- The . gitignore file is a text file that tells Git which files or folders to ignore in a project.

```
#create gitignore file
touch .gitignore

# ignore ALL .png files
*.png
```

- **Branch:** a parallel version of the master copy of a repo. Making a branch allows you to edit code without accidentally breaking a working version

- Operations on Branches :
  - List the branches : ‘git branch’
  - Create a branch : ‘ git branch <branchName> ‘
  - Delete a branch : ‘ git branch -d <branchName> ‘
  - Switch branch : ‘ git checkout <branchName> ‘



How to deal with a repo

- **Repository** : the folder that contains the project(source code , assets …)

- From github :
  - Create the repo (if it does not exist)
  - Clone it to your local machine using ‘ git clone <Link> ‘

- From local machine :
  - Create the repo on github
  - Initialize the local repo using ‘ git init ‘
  - Commit the files using ‘ git add . ‘ and ‘ git commit -m”init repo” ‘
  - Configure the remote variables using ‘ git remote add origin <link> ‘
  - Push using ‘ git push origin master ‘

Push : upload the changes from your computer to your GitHub repository.

Pull : download the changes from your computer to your GitHub repository.

Let’s Practice!

- Open git bash
- git config --global user.name “Your Name”
- git config --global user.email “your email address”
- To show the current config : git config --list

Then

- cd desktop
- mkdir project
- cd project
- Go to your desktop, you will new folder created calls”project”, copy your code and iris data in it.
- Go to github, create a repository with the same name.
- Follow the instruction to push your first project to github

IEEE Student Branch University of Boumerdes

ML BOOTCAMP

ThankYou!