**INTRODUCTION**

Basically, my project is about handwritten detection and we will work with MNIST dataset, so we will test it with many classification's Algorithms and Artificial neural network, so let us start the journey with some AI concepts.

**WHAT IS ARTIFICIAL INTELLIGENT (AI) ?**

Actually, there is no specific definition or comprehensive definition for AI

because there is a complexity in the concept of intelligent itself, and there are multiple behaviors that the human do, so it is difficult to judge which behavior is intelligent and which is not, so for example we can define it like this.

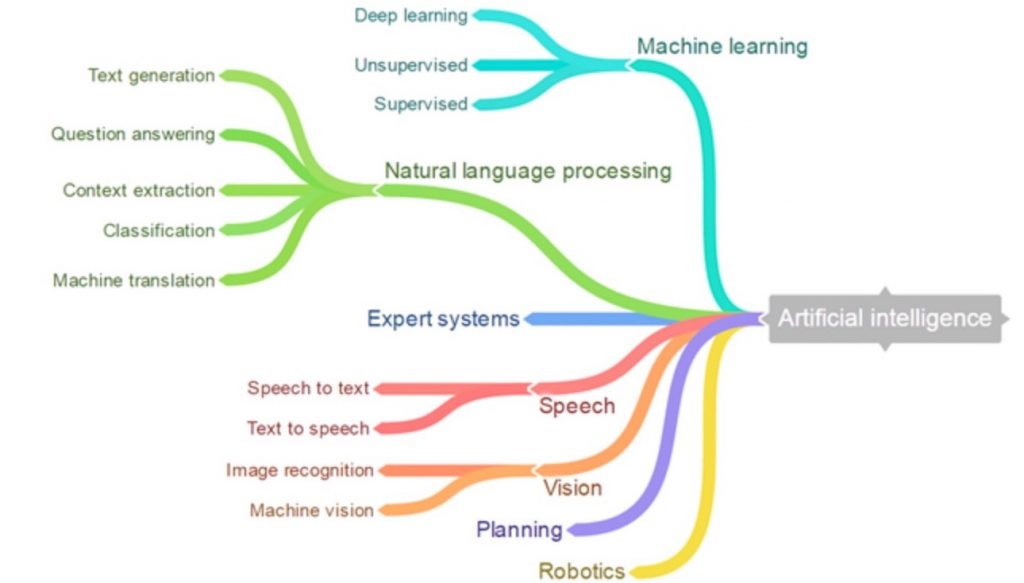
The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)

Or this:

"The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

And there are more definitions but all the definitions fall in for categories, like the first definition we talked about falls in the category of "the system that act like human" the second one falls in "the system that act rationally" and the rest of categories are "the system that think rationally", "the system that think like human".(29)

**AI is really huge scope and it has very interesting subtopics and for each subtopic, it considers as a branch and specialization in itself, the figure below will illustrate what I am saying (30):**

 Figure 1 :XYZ

Absolutely it is not enough to cover the main concepts of AI in few lines but as you can see it is really huge brunch, and my project depends on machine learning so I will focus on it, in addition to that there are some people still mix between the two concepts of ( AI and ML), so I hope to know the differences after finish reading.

**MACHINE LEARNING (ML)**

ML is a branch of artificial intelligent which contain many algorithms aims computer to learn from the experience automatically without being explicitly programming (31)

**HOW DOES MACHINE LEARNING WORK?**

Actually,machine learning is like human, learn from the experience so in the first place we have to feed our ML's algorithm the experience which represented as training data, then from these data the algorithm of machine learning will create a model, now when we introduce a new data, the algorithm will predict a new values based on the model, the next step which considers as very sensitive and important point that is to evaluate the accuracy of the results, if the accuracy is good then, the ML algorithm will deployed, otherwise we will train our algorithm again and again to get acceptable results, simply this figure will illustrate and summarize everything we used.

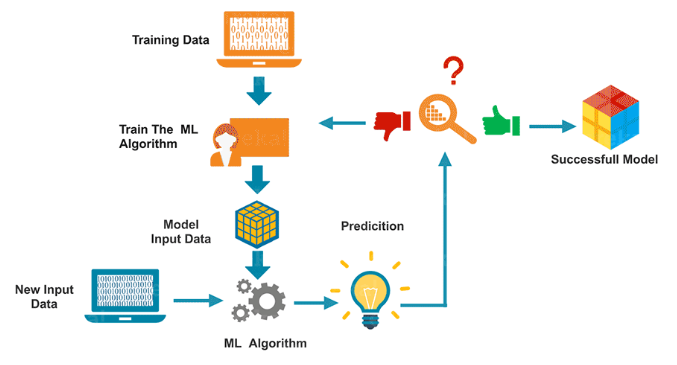


Figure 2

**TYPE OF MACHINE LEARNING**

There are three main types in machine learning which are:

1. Supervised
2. Unsupervised
3. Reinforsmant

And under those three types you can find many algorithms, each one has a specific situation or problem to use with.

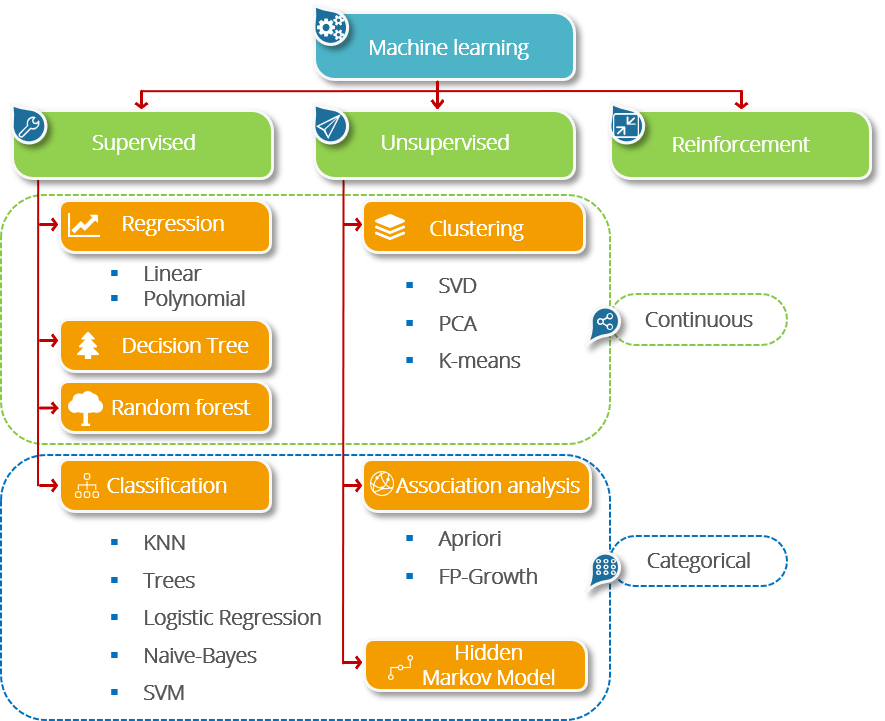


Figure 3

**ADVANTAGES AND DISADVANTAGES IN MACHIN LEARNING**

Actually, Machin learning has two faces as any coin and you have to know the properties of each face to make your decision clear that when you should use Machin learning and will you not, because in some situations ML will be useful and others it will be useless and just wasting your time, so let's start with some advantages

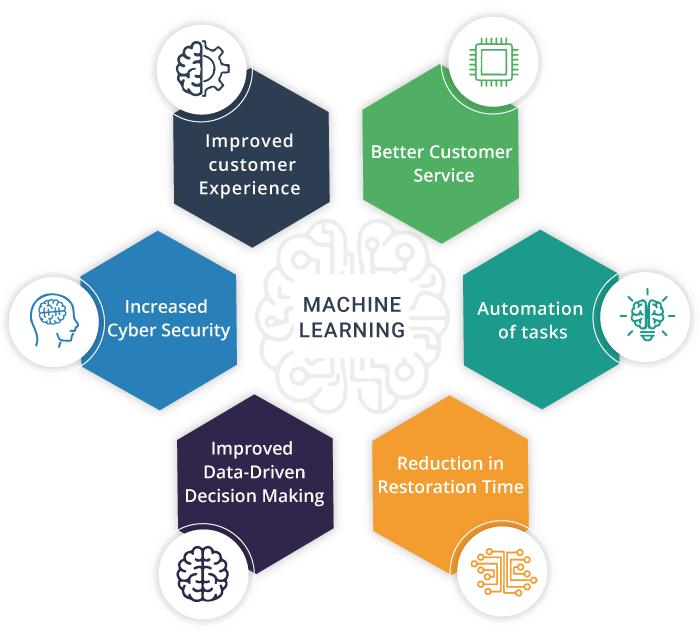
**ADVANTAGES OF ML:**

**1- Easily identify the trends and pattern :**unlike human ML can see and review a big amount of data like the data which come from browsing process, searching and purchased histories, all of that make ML able to recognize what is trending, and give pattern (ads) that interest you, like from a real example of that happen to me, 95% of ads on YouTube were about courses in ML and the cause of that is related to my searching to learn new concepts in ML

**2-Automation**: ML work automatically, that mean you don’t have to care about improving your program because it will do that by its own, all you have to do is to feed your algorithm with training data and the model will make predictions and improve to enhance the results, the software of anti-virus are good example of automation, they learn automatically to filter new threats as they are recognized.

**3-Continuous improvement**: ML program always gain experience after training, this makes it in continuous improvement in the accuracy of the result,

**4-Wide applications**

In addition to that we have some benefits can get from ML as below (40):

**Figure 4**

**DISADVANTAGES OF ML**

**1-Data acquisition**: actually, to train the ML's Algorithm we need a huge amount of data, so the first thing that the data must be inclusive and in good quality and the second thing, sometimes we have to wait until the data get generated

**2-Time and resources:** the algorithm in ML have to take enough learning time to give us an excellent accuracy, in addition to that some algorithm need a massive power to generate, so sometimes we need a super computer and this will be cost**.**

**3-High error-susceptibility**: yes, ML has very high susceptible to give error results, especially if my dataset was small, suppose I have some data related to customers and some of these data has a high biases, I want a program with ML to recommend products to those customers, in this case my result has a bias also and what will happen is that a specific ad for example will go to the wrong customer, so we have to be careful if we have a high biases in the data, another thing that sometimes there will be some bugs in my training but these bugs is undetected, and when it get noticed it will be difficult to know where is the source of this error and will consume us a lot of time to fix the problem .

**APPLICATIONS OF MACHIN LEARNING**

There are a wide rang in applications we can use ML with, nowadays ML have a big role in our life and day by day it get more popular, the figure gives us the most popular scope that it uses ML everyday (40):

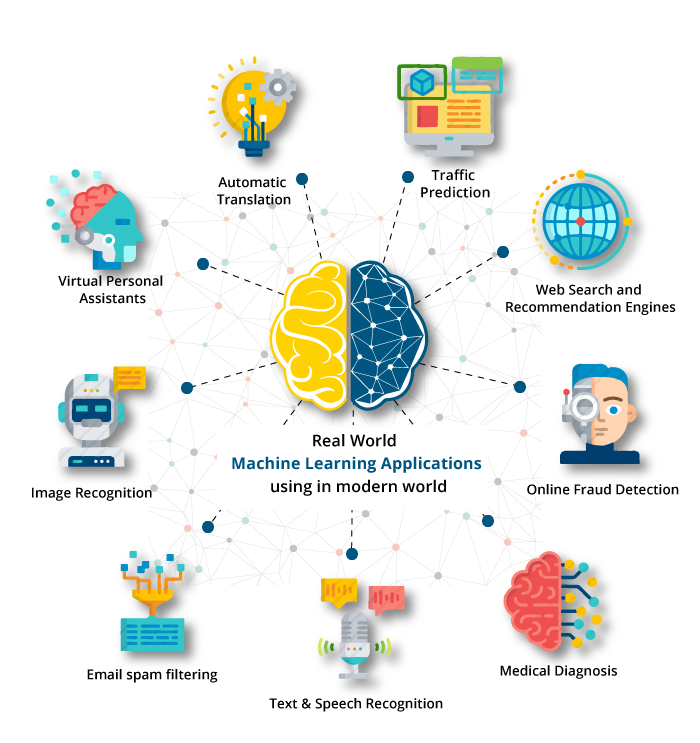
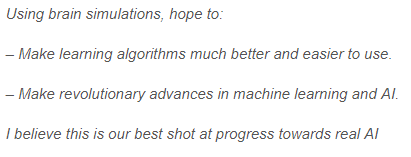


Figure 5

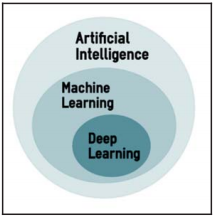
**DEEP LEARNING (DL)**

"DL is a particular subset of ML methodologies using artificial neural networks (ANN) slightly inspired by the structure of neurons located in the human brain"(41)

Andrew ng from coursera and the chief scientist at Baidu research said when he was describing about deep learning as

"

**Simply** and without complexityyou can find deep learning domain within AI and ML like the figure below:



**Figure 6**

Deep learning considers as Tsunami in machine learning because it is one of the most successful methodology that used in ML which take place in many domains like (images, text, speech...etc.) (41)

**But what does make DL different from author techniques?**

The answer of this question is simple, what make DL distinct from other methodologies that its ability to increase and enhance its performance while we train more and more data with it (42), in other word, as you give your neural network data as you get better performance, second: deep learning has a fantastic ability you can't find in other methodologies which is the feature extraction, DL can extract the features by itself, in addition to that DL is a simulation to our brain and this gives it a brilliant capabilities to learn from experience (44).

We were taking an overview about AI, ML, DL we have to talk about the main data set that will work with and apply it on difference ML'S techniques which is MNIST.

**What is MNIST dataset** **?**

First of all, "MNIST dataset" term is a shortcut of "modified national institute of standards and technology" dataset, which is large dataset and contains images of hand written digits that used widely to train and test in the cycle of machine learning, MNIST created by remixing samples from NIST's original dataset, the training dataset was collecting from American employee, on the other hand the testing data was collected form American high school students, the images of MNIST was black and white then it fitted to 28x28 pixels, the dataset contains 70,000 images and it splitted into 60,000 images for training and 10,000 for testing.

There are many scientific paper about applying ML on MNIST and the purpose of these papers to get the most lowest error rate, one of these paper used CNN with MNIST and it get 0.23% error, so in the end of this report we will be able to apply and experience many ML's techniques on MNIST and we will compare its rates of errors.

**METHODOLEGIES**

In the first step I had to learn basic things in python so I learnt the basic things in it, then I learnt the two popular libraries NumPy and pandas, after that I moved to learn some preprocessing steps and finally I moved to machine learning and some algorithms, so let's start with python from now

**PYTHON**

Python is a programming language and when we say that it is a programming language we mean that it is a language designed to express the computation that can be performed by the computers , Python has a specific vocabulary of syntax to do instructions and tasks with it, and generally we use the term "programming language" to refer to a high level language which is the third generation of programming language .(1)

**A brief history of Python**

Python developed by GUIDO VAN ROSSUM and the first version 1.0 released by its designer in 1991 at the National research institute in the Netherlands, python derived from many languages like c, c++, ABC... etc. and the story of the name ''python'' is funny, once he started implementing this language, he was reading a published script from BBC comedy series which its name "Monty python's flying circus" so ROSSUM wanted a short unique name so he thought python would be a perfect name. (1)

**Features of Python**

As we said before python is high-level, interpret, interactive and object-oriented language and has good combination of performance and features which make it powerful and easy to learn, python has many features which make it attractive to a huge audience, we will mention some of its features below:

**1-Simple**: it is the best feature in python that make it strength, writing a program in python is like writing with English language, python is really close to English and it is good aspect to make the programmer to concentrate on the problem and solution rather than to concentrate on the complexity of the code.

**2-Free and Open Source:** it is an open source software so anybody can distribute, read the source code, edit it and reusing the code to write new one

**3-High level language**: the programmer doesn’t have to concern about the low-level language and the details in between like managing the memory he just has to concentrate on the problem that he is solving

**4-Interactive:** python has an interactive mode which make the programmer to test and debug pieces of his code

**5-Extensive libraries:** python has a huge different library which are portable (work on different operating system with same interact and the same inter face) and allows the programmers to perform a wide range of applications starting form text processing to the GUI programming.

There are more than these features mentioned above, like the dynamic executes, object oriented, and secure, but what about the disadvantages or the limitations in python?

**Some limitation in Python**

* Python is slow if you compare it to c, c++ because it is an interpreted language and this makes it not very good choice for applications that take a lot of CPU processing like developing a high-graphic 3d game. (1)
* Python has a weak computation in mobile developing and you can see that there are few mobile applications that built with python. (2)
* Python has a high memory consumption and that because of its flexibility in data types so also it is not a good choice for memory intensive. (2)

**The future of Python**

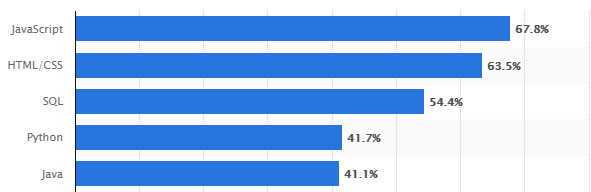
Python has a huge audience and constantly is growing year by year (4), and from the perspective of using a programming language python fall in the 4th rank among the languages that use it by developers in the worldwide, as of early 2019 as the figure below (3):

Figure 7

We can understand the strength of python by looking at the companies like Google, YouTube and

NASA which prefer it to its easy syntax and there are many companies have started to use python in a different application, all this attention on python is cause by its supporters, python has a huge community of operating system developers that support it on many aspects like opp, functional and parallel programming so with all of that we think that python is going to stay a long time and has a bright future ahead (1)

**Note**: Within my lovely IESTE internship duration I have started to learn python from the basic concepts like datatypes ,basic operations, conditions, for loops, different structures, and functions but unfortunately I can't mention and explain all of these concepts right now because I have no more time, so from now I will start to introduce a popular python's libraries that is useful for data preprocessing and performing a different ML's Algorithms through it.

So basically, there are two libraries you have to know if you interested in data analysis and machine learning which are:

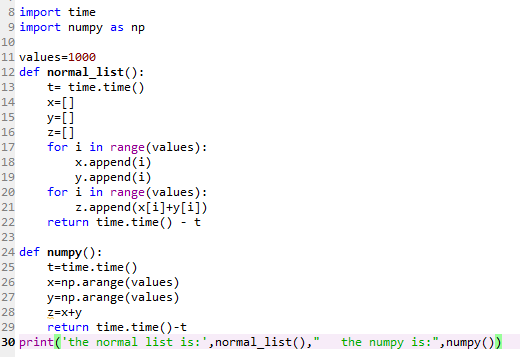
1-Numpy

2-pandas

**What is NumPy ?**

NumPy is a mathematical library, it considers as a fundamental package for scientific computation, it gives a multidimensional array and allows to apply a fast operation on its array like mathematical, logical, basic linear algebra operations and more.

The most incredible thinks in NumPy are the size, NumPy take a less space in the memory than other structures, also its speed when comparing it to other structures like the speed of list, and that is because NumPy don’t use loop which cost the time, also has a fixed size and all the indexes must be in the same data type therefore all the elements will take the same space in the memory, in addition to that NumPy stories its elements sequently unlike list, here is a simple example code to compare the speed between a normal list and NumPy (6):

****

**The output:**

****

The previous code shows a speed comparison between summation two lists using loop, two np.arr without using loop, obviously we can see that the best performance go to NumPy, absolutely the difference between two numbers is small and maybe will not affect too much in a simple program like that but conversely the speed and the performance will differ in complex application.

**Basics instructions in NumPy**

**1**-how to import NumPy library and is simply like this:

****

np is NumPy shortcut, you can choose any shortcut you want but np is most popular name between developers.

**2**- creating array from zero elements:



**The output** :



And you can choose how many elements you want …. Also, you can create nparray from one element using 'ones' as the same.

**3**- you can use "linspace ()" method to create an array with range of number you determine with start and stop like this

**The output**:

****

**4**- we can build a NumPy array using "np. array ()" and we have to put a list between the brackets or simply we can put the name of the list



**5** -there is "arange ()" method whereas you can determine a specific range or you can to determine the start and the stop with specific numbers for jumping in this range



**6**- we have "reshape ()" method which adjust our array to new shape in new dimensional but absolutely the numbers of element must be suitable with the new shape, otherwise it will give us wrong



 our array:

this is how to reshape it:

**The output**:



**7**-the ' shape' will return a Tuple illustrates the array shape



**The output**:



8 - we can do some slicing on the array like this:

**The output:**

****

**9**- we can perform many operations on the elements like "sum", " mean", "std", we can get the "max", get the "min" and many others.

**10**- we have 'size' which give us the number of the elements in the array, also there are more and many useful instructions in NumPy we will not talk about but now let's jump to pandas.

**What is Pandas?**

Pandas is an open source library in python, it allows us to deal with different data structure like csv, html, sql...etc. and provides us with high performance in data analyzing, it considers a good tool in data preprocessing as will and we are going to see some instructions in preprocessing in the next chapter (8)

**Basic instructions in pandas** (9)

**1-** we have to import pandas in the first place:

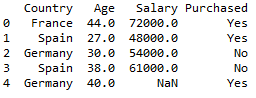


**2-** we can create data frame as we want using 'DataFrame':



**3-** simply we can read a different type of files with pandas using 'read.files\_type', for example I am going to read a 'csv' file:

but you have to know that the data which you want to read and the project that you work on must be in the same files or you have to write the path.

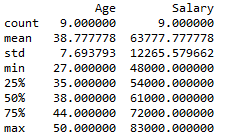
**4**- we can take by default the first 5 elements in any data frame using 'head ()' to see my columns:

****

Also we can choose any number of rows we want, like if I want the first 100 rows, I can get simply like this 'df.head(100)'

**5-** there is a method call 'tail()' which its job like the method 'head()' but it start from the last row and so on

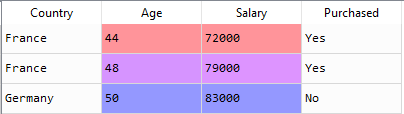
**6-**we have many methods which are exactly the same as NumPy methods like size, shape, many mathematical operations on columns like the mean, median, std …. Etc.

**7-** there is 'describe' which we can use it with brackets and without it, if we use it with brackets it will give us a statistical information about numerical columns like this:

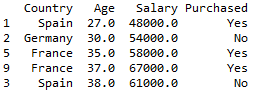


But using describe without brackets will print the all rows we have

**8-** in the case of our example we can take a new data frame with some conditions like if we want to take the person's data whose age is above 40:

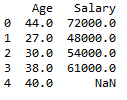
****

****

**9**- we can sort our data by 'sort\_values', for example I will sort depending on the age:



**10**- A very important instruction is "iloc[]" to take a range of columns in the data frame, we use it generally to choose the features and labels from the dataset to feed it our model in machine learning, here are our features for example "Age", "Salary":

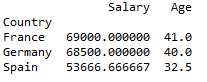




**11**- also we have a method is 'groupby()' to take more than one columns to a specific purpose, let’s move on our example and suppose we want to know the mean of both the salary and the age in each country so 'groupby()' what will do is to classify the country columns to categories and give us the information we need to each category.



Okay this the result now**:**

****

**I think mentioned the most popular and the most used functions in both NumPy and pandas' libraries, I will speak more about pandas in the next chapter because it has a big role in data preprocessing, so let’s talk about cleaning data and then we will move to machine learning.**

**WHAT IS DATA PREPROCESSING IN ML?**

t is a term that refer to some techniques and transformations that applied to our data to make it suitable for machine learning algorithms. (10)

**In another world it is about group of steps to make your dataset ready to fit in your model**.

**THE NEED OF PREPROCESSING**

In the real world data generates every day and you will get the most of these data in raw format which make it unstructured, uncompleted , bad for modeling and it will cause some problems through modeling process, so as you know data is very important factor in ML either to predict something or to recommend, therefore as good quality of data you have as good prediction will get and her is the mission of preprocessing, so it is all about converts the raw data to structured readable data , handles the missing values and many other things which make your dataset perfect before feeding it in your algorithm (11)

**STEPS FOR DATA PREPROCESSING**

Before talking about the steps we should know that almost any programming languages can deal with ML problems like python, c++, c#, java ,R…etc. which absolutely all of them can do preprocessing on the data, but of course there is a deference between writing the methods by your hand and using ready-made methods so this the main factor which determines the ease of the language (12).

Fortunately, python make it easy for both preprocessing and machine learning and this for its scientific libraries and due to my new experience in python the examples and the illustration figures will be in it so let's jump in.

Usually you will find the same concepts of data preprocessing in all languages but the differences will be in the own programing instructions to each language (3), now there are many steps in data preprocessing but the most widely and important ideas you could deal with are:

(1) handling missing values

(2) working with data types

(3) class distribution

(4) standardization

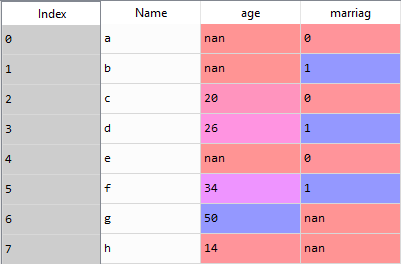
(5) label encoding

(6) One hot encoding

**Handling missing values:**

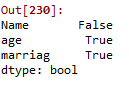
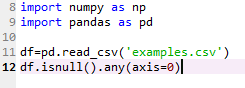
Sometimes you will find some missing values in your data and this not good for your model, you will face difficulty to manage your data and probably will give you inaccurate result so whatever you use either regression or classification or any other kind of problems you must check the data.

Here is a simple example of a data set has missing values (its express the relation between age and marriage situation)



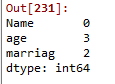
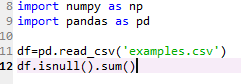
Obviously, you can see 'nan' row which in Python **indicates** to missing values but suppose that my data is large and contain like more than 1000 rows, how can I know which column has nan values?

Simply there is a method called 'ISNULL()' form pandas, it can show us the rows which have nan values …. Let us check:



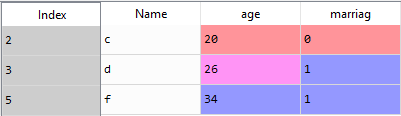
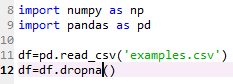
**The Output** :

But as you can see, we added "any(axis=0)" with "isnull ()" method to give us the columns which have missing values, to know the numbers of missing values in each column just use "sum ()" with "isnull ()''

 T**he Output:**

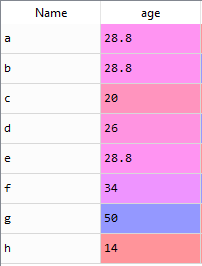
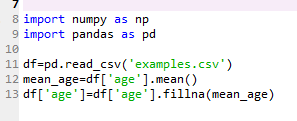
Now there are many choices to deal with these "nan" values

one of them is "dropna()" method, it removes the complete line if it is contains one missing value or more



(the result)

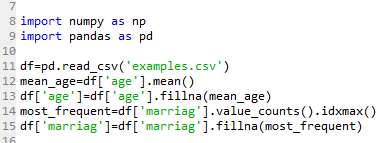
However it's not always good choice to remove the rows which have missing values by that method ,if our data is large and have a view missing values will be ok and this will not effect on the results and the accuracy, but suppose that our data has like 20 % nan values of the data, this will lead to lose a valuable data therefore you can use "fillna()" method which can fill NAN values with.

By going back to my data, we can fill the nan values in "age" column by the mean or median in the same way:

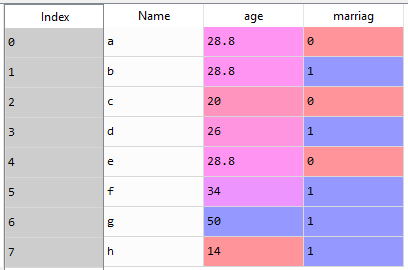
At the first point we calculate the mean value of age column which is equal to 28.8 then we filled it in the nan values by "fillna()"

For marriage column we cannot fill it with mean or median because its rows contain zero or one which express the situation of marriage so we can take the most frequent value and fill it in nan values rows.

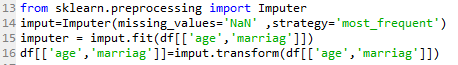
So for that we can use "value\_counts()" method which give a series of the values and for each value it gives the count of how many has the value frequented in the column then we can use "idxmax()" method to select the most frequented value and we can fill the nan values with it.



Here is the data now, clean and has no missing values:

****

In addition to that we can use the Imputation with specific value like mean and most frequent value ,you can use "impute" method by import it from "sklearn.preprocessing" library, by the way this method has many parameters like "missing\_value" ,this parameter is for specifying what the value you want to replace, for example int or NAN…etc. and by the default it will be the NAN values ,there is another important parameter which is "strategy", and we mean by strategy is that to choose how you want to change the NAN values, in this method we have three strategies to use which are as we said before (mean, medin, most\_freqent) (24) here is the implementation of the code on the same dataset and it will give us the same results:

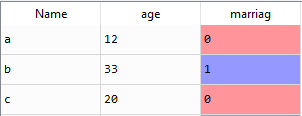


- for Further information about the code we can say that the "fit" generally do the calculations aspects and obviously "transform" its job to transform using the fit's calculations (25), as to the example her the "fit" choses the most frequent value in each columns for both: "age", "marriage" and "transform" fills the NAN values with it.

**Working with data types:**

**C**ommonly we use many data types to store information like: int ,float string…etc in pandas, int equals to int64 (64 refers to the location where the data stored) and so on, also an object in pandas refers to string or mixing types, the important thing you have to know that the most of machine learning Algorithms deal only with numerical values and sometimes there are some rows that seem to contain numerical values but surprisingly and by mistake of user they are object, in situations like this we can use "astype()" method to transfer the object (numerical value written by string) to an int or float…(18)

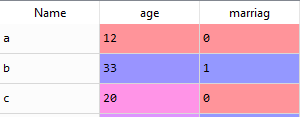
Here is an example:



As you can see in "age" column there is no colored rows which mean that these rows written as an object, we can check by "dtype"



**Now** we can transfer this column to int using "astype()" method :

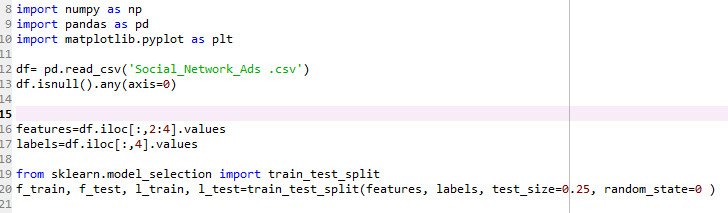


**Class distribution**

A very important step you have to know is to divide your data and split them into training sets and testing sets, this will help you testing your model with unknow data and to estimate how much you get an accurate result, but suppose you fit all your data in training process and you use some of the same data for testing , this will not be useful because your model know the data by heart, another thing that why splitting data is important that is because it avoid us to get overfitting to our model(19)

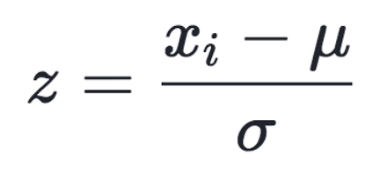
Overfitting refer to: " Good performance on the training data, poor generalization to other data."(20)

We can split data by using "train\_test\_split" method by import it from "sklearn.model\_selcation" library ,this method spilt data by default to 75% for training and 25% for testing, all you have to do is specify the features and labels in tow data frames and then 4 data frames, two for testing and another two for training (19) as illustrated in the this figure (21):

****

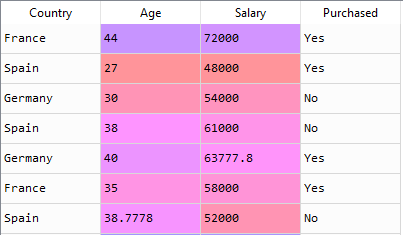
**STANDARDIZATION**

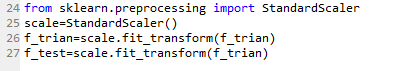
It is one of scaling methods that we use it to rescale our features through making the mean of the values equal to 0 and the standard deviation equal to 1 , we used to do standardization when we have more than one feature and when the differences between the values is very large, all you have to do is to calculate the std and the mean of the values and then applied this equation for every value :



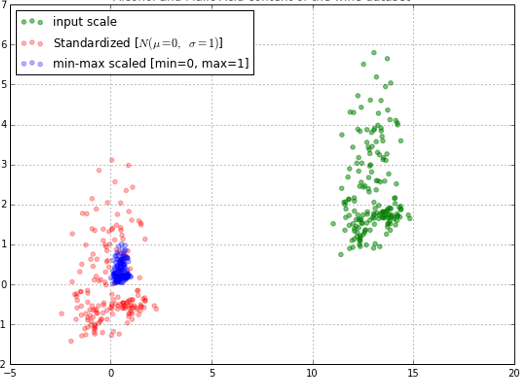
μ: the mean of the values, σ: the standard deviation

This is only the mathematical aspect of the standardization but the good news is that instead of use this equation on every value we can use "sklearn.preprocessing" library to import "StandardScaler" method (15), here an example for dataset need to applied standardization on its features (22) (the data shown after handling its missing null values)



Suppose our features are the Age and Salary columns, you can see the big difference of the values between these columns and the weightage is for Salary for sure, so we need to rescale these columns, note that you have to do the rescaling for both the train features and test features:

For more illustration, this is a visualizing for the rescaling process that we did as the same as the d points in the figure bellow (23):



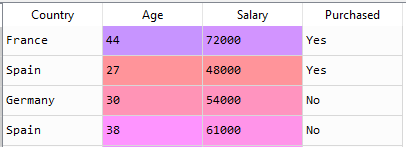
**LABEL ENCODING & ONE HOT ENCODING**

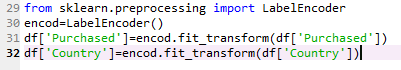
As we know almost ML algorithms deals with numerical values only, and sometimes categorical data(text) considers as a valuable factor to our model, may be these categorical data are either features or labels, so the process of transforming our data to numerical is very important, there are many way to convert the categorical data to numerical values, one hot encoding one of them but be for we talking about it we should talk about label encoding first.

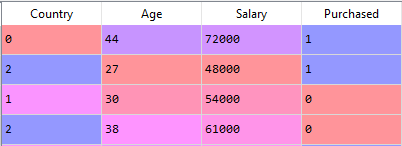
**What is Label Encoding ?**

It is the process of transforming data to int value starting from zero to N number, depending on how many categories we have in the column , also when a specific value assigned by 1 for example it well be the same value even if it repeats in the same column and generally label encoding is good with classification problem (26)

By going back to the sample (22) you can see that it is a classification problem, in "Purchased" column we have yes and no categories, also in "Country" column we have many countries, so we have to transfer those categories to 0 and 1 and so on by using "**LabelEncoder**" method which we can import it form "**sklearn.preprocessing**" library as below:



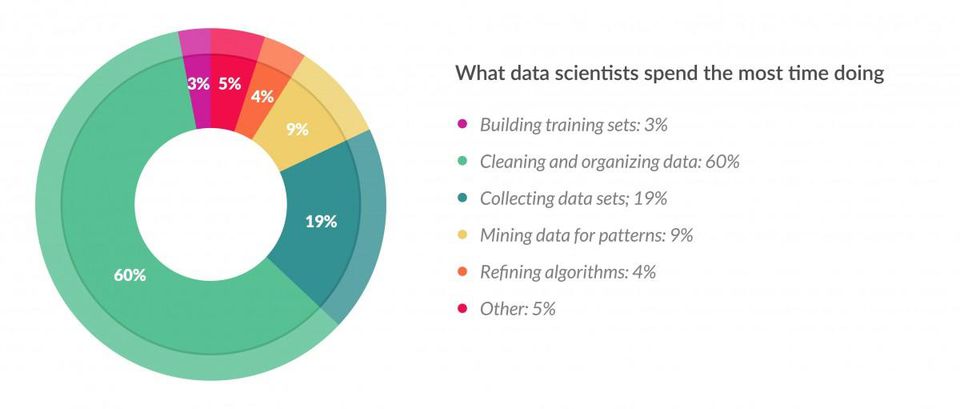




But still there is a problem here and this problem depend on the dataset we have, yes, we have converted the categories to different values but actually these values have no relation between other rows and may be our model will misunderstand and take the data in order like 0 < 1 <2. (27)

For example, in our dataset the model probably think that as well as the number of the country increase the salary will increase and the person will purchase, but actually this is not the case and the order of the values here has no importance so the solution here is one hot encoding so what is it?

**One hot encoding** is a term refer to represent the values in binary vectors, essentially it makes the categories data more expressive and it work when there is no ordinal relation between the rows, we can do one hot encoding in many ways ,for example by using "sklearn.preprocessing" we can import "OnHotEncoder" from it but note that by using this way we should use label encoding on the categorical data before applying one hot encoding on it , also we can use manual way to make it or by using keras library in python by import "to\_categrical" method which also we must apply it on int values (it need label encoding first) (26) .

Okay I think I mentioned the most using techniques that we use in data preprocessing, for sure there are more techniques and more concepts about cleaning data I didn’t mention it, acutely it is really difficult to cover all the concepts in few papers, also from the practical aspect ,data scientist spend the most of their time in preparing the data and cleaning it as the statistic below (28)

**Yes, it is not expected that the data scientist spend 60% in cleaning the data and this is just a strong evidence how much the data preprocessing is important and how much effectiveness it has so we have to care about this point because as the data is good ,our model will be good.**

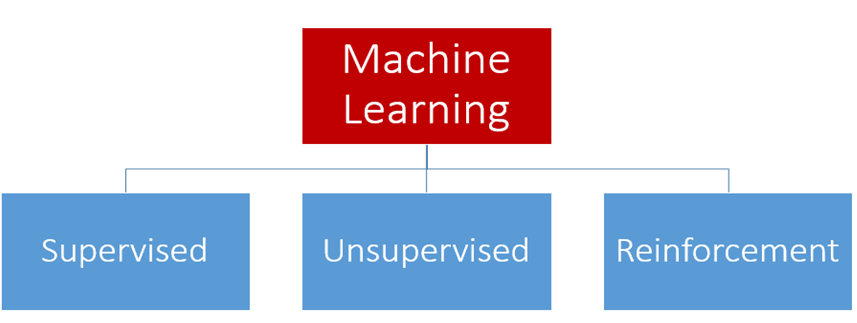
**Let's move now to discover some techniques and algorithms in Machine learning.**

**MACHINE LEARNING (ML)**

As we said before, ML is a branch of artificial intelligent which contains many algorithms aims computer to learn from the experience automatically without being explicitly programming (31)

**Type of Machine Learning**

There are three main types in machine learning as to figure below:



So as to my internship which is 2 months and to my project (handwriting detection) which consider a supervised problem I will focus on supervised machine learning especially on classification algorithms.

**SUPERVISED LEARNING**

It is the most category used in machine learning, simply if you want to explain it, you can say it is like teacher who supervise the learning process in the class, so the teacher know who is a good student and who is the bad one, also he can predict the situation of new student biased on his experience, conversely here the teacher here will be the training data and the student will be the values that I want to predict, from mathematical aspect it is all about we have many X variables and to each variables we have its label Y (THE TARGET) and from these information our ML's algorithm will create a mapping function (model) and basically form the mapping function we can predict the labels (Y) for new data (X), in the end, if our algorithm do well we will stope learning otherwise we still train the model (33).

**Type of Supervised Learning**

There are two categories for supervised learning which are:

1-**Regression**

2-**Classification**

And for each category it considers as a family of many algorithms in itself and we use each algorithm depending on the problem we have.

**Note**: there are more categories under supervised learning like Decision tree and Random forest (32), but I will not talk about them because this is out of my knowledge

**Where do we use Regression?**

Generally, we use regression when we need to predict a value of some data ("continues-response value''), for example we can use it if we want to predict a house price in specific region, or to predict how many calories that the person will spend in the gem, and many other problems .(34)

There are many Algorithms in regression problem (32):

1-**Linear Regression**

2-**Polynomial Regression**

**Where do we use Classification?**

Classification uses when we want to predict a "categorical-response value", it is to separate your data and put it in a specific classes, for example you have data about some people like their heart rate ,the sugar in the blood, age, gender…etc. and some of these people have a diabetes so we want to predict if someone has diabetes or not , it is like yes no question and when we have yes no question it will call binary classification.

Also, there are many Algorithm in classification:

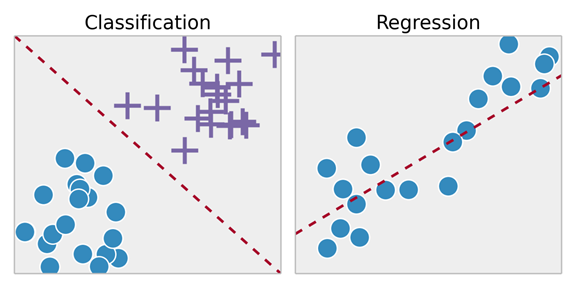
**1-KNN**

**2-TREE**

**3-Logistic regression**

**4-SVM**

**5-Naivy-Bayes**

Generally, the mapping function for both regression and classification will be like this in the end of training:

Okay after presenting many algorithms in both regression and classification, now I will start to explain some of these algorithms as follows:

1- I will explain linear regression algorithm, it has no relation with MNIST but it easy to start learning ML's algorithms from it because it is simple

2- I will move to classifications algorithms and explain logistic regression, KNN, SVM

3-I will move to talk about artificial neural network

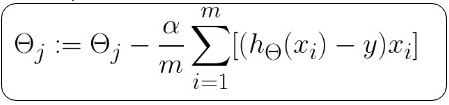
**LINEAR REGRESSION**

As I said before linear regression is a supervised learning algorithm uses to predict values Y form given values X, so form our training data the function between x and y will give us a linear relation and this why this algorithm call liner regression, from mathematical aspect the hypothesis function like this:

Through training process, we give x and y to our algorithm, after training, the model gives the best fit line to predict y, note that you can gets the best fit regression line for your model by finding the best**θ1 & θ2**, then we can predict new values y from new inputs x, but how can we find them?

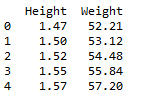
We can use the cost function (J), it is all about to update the values of **θ1 & θ2** to get the most minimum error between y that we predict and the true y that we have, so this is the cost function which calculate the error between predicted and true values y (35):

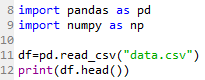
In addition to cost function we can use ' Gradient Descent' to update **θ1 & θ2** and the purpose of that to minimize the cost function, so we start with random value of **θ1 & θ2** and iteratively we update these values to get the minimum error of the cost function, here the final form of Gradient Descent equation after deriving the cost function (36) :



**How do we implement linear regression with Python?**

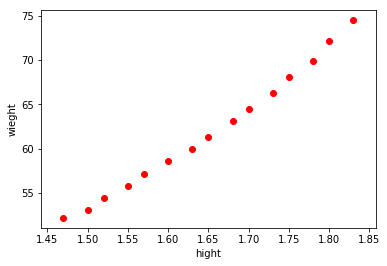
We can implement it buy using the lovely library of python call "sklearn", so from '' sklearn.linear\_model '' we can import our linear regression method.

Here an example of data set related to American women's Hight and weight and their age is between 30-39, so form the Hight we can predict the weight, note that it is a simple linear regression because we have one feature which is the Height



Our data is simple as you can see and it doesn’t need rascaling because I have one feature and its label, let's check if I have NAN value



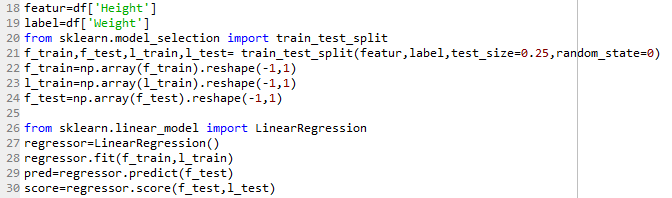


So, our data is clean and ready

Here is the data using "matplotlib.pyplot" library



After that we have chosen the feature and label, and we have splitted the data to training and testing data, then we import our method "LinearRegression" to train our data, after that, we predicted the label for the remain feature of the data (feature testing) by using "predict" method as you can see in the code.



After predicting we have to see the accuracy of the predictable data so we have used "score" method (in the line 30), so "score" compare the results that we predict and the true labels we have, as to our example this is the score:

Yes, the accuracy is 96% and it is considered very good.

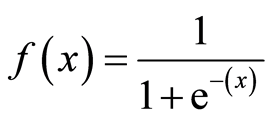
This is the simple linear regression, by the way the difference between the simple and multiple linear regression is that in the multiple we have more than one features so maybe we have to rescale our feature if it needs but the steps are the same in both.

**LOGISTIC REGRESSION**

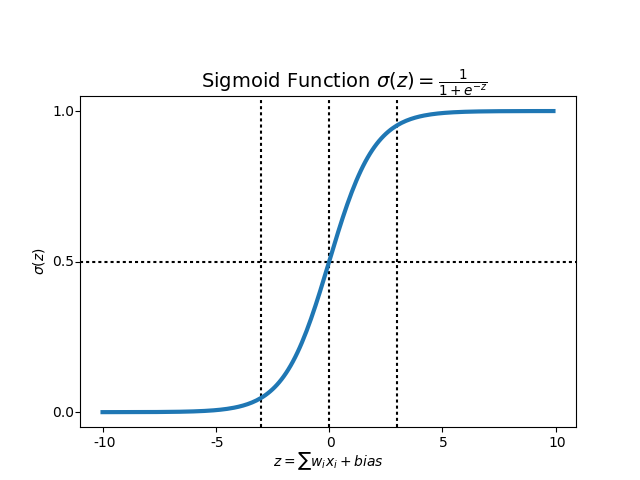
It is from classification's Algorithms family and it use when we want to predict a categorical value, basically the name ''logistic regression " come from logistic function (sigmoid function)

If you look on the features of your problem in logistic regression, you will see that they features combined linearly as the same as linear regressing but the difference here in the output (y), in logistic regression the output (y) is classed into two classes ,either 1 or 0, in another word logistic regression is a linear method but the prediction is transformed to classes and this by using logistic function (45)

The sigmoid function is observed as below:



Here is the sigmoid map:



**What does the sigmoid function do?**

It transfers all the predictable data to a range between 0 and 1 and while you using it is impossible to get a value more or less than the scope [o,1], so we use it when we want to get probabilities (46)

In fact, using logistic regression doesn't give us a direct value as 0 or 1, the idea is there will be a threshold between 0 and 1 and almost the value of the threshold will be equal to 0.5 but this depend on the problem itself, so suppose I have dataset of apple and mango images and I want to classify them, 0 value will be apple and 1 value will be mango, my threshold is 0.5, so if my algorithm give me a predicted value like 0.7 so this will classify as mango, if the predicted value was below the threshold it will be an apple and so on (46).

**Type of Logistic Regression:**

1- Binary classification (we use it especially when we have yes &no question)

2-Multinomial Logistic Regression (we use it when we have more than two categories like many colors red, green, blue :)

3-ordinal logistic regression (three or more classes and it give us the result with ordering like the rate of movies from one to ten)

**How can we use Logistic Regression with Python?**

We can do that by using the lovely sklearn library, from "sklearn.linear\_model" we can import our method "LogisticRegression" simply.

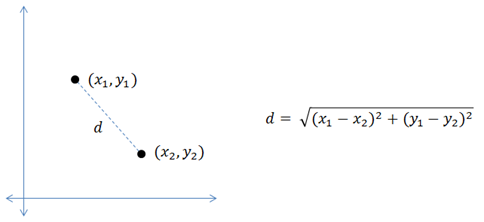
**K-Nearest-Neighbors** **(KNN)**

It is a classification algorithm and it considers as the simplest algorithms in classification, we can use KNN in regression problem but it uses commonly with classification, KNN has no model to depend on, other that there is sorting in the entire data we have, so there is no learning required (49)

**But if there is no learning required in KNN, How it works?**

Acutely KNN algorithm makes the prediction directly, for example I have an X variable and I want to predict that which class X is from, what the algorithm will do is that it will search through the entire training data for the K variables which are very similar to this point , then it will be classified with them(49).

To determine which is the nearest k variables that similar to mine, I can use many ways, one of the most popular way is Euclidean distance as below (50):



There are more ways to measure the distance like Hamming distance, Manhattan distance and Minkowski distance and just what I said before the Euclidean distance is the most popular used but notes that sometimes one of these ways will be more suitable to some specific problem so we have to be careful and it is good to try more than one way to see which one give you the lowest error rate in your problem (49).

**How do we choose the number of K?**

For k value it will be 5 by default, and it is good to try many training with different number of K to see what is the best performance but it is better to choose the value of k as odd if you have an even number of classes and invers, if you have an odd number of classes use an even value of k.(49)

**How to implement KNN with python?**

From "sklearn. neighbors" library import this method "KNieghborsClasifire" and it will work simply (51).

**SUPPORT VECTOR MACHIN (SVM)**

It's also another algorithm from classification family and we can use it in regression problems but the main using is in classification domain, the purpose of SVM is to make a hyperplane (line that will be between two classes as separative line) in N-dimensional space so that make the two classes separated. (52)

**How do we choose the perfect hyperplane?**

There are many hyperplanes we can created but the best one will be the hyperplane which has the maximum margin (the maximum distance between the data of two classes), in addition to that the dimension of the hyperplane depends on the number of the features I have, If I have two features then my hyperplane will be one dimension(line) , if my features were 3 then my hyperplane will be three dimensions , and it is possible to be 3 dimensions if my features exceed the number 3 (52).

**How does SVM work?**

**Theoretically**, First of all the SVM algorithm get a linear output from linear function ,then it transfers it and classify it into a specific class (maybe the first one or the second one) we have, the process of that similar to logistic regression, there are in both Algorithm a specific threshold, for example in LG the value was (0.5), here in SVM differ to range between 1,-1, SO if the linear output was more than 1 it will classify to first class and if the output was lower than -1 it will classify to the second class (52).

**How do we can implement SVM with Python?**

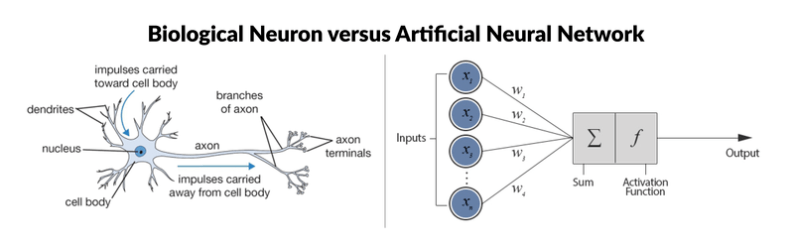
As usual from sklearn library, so from "sklearn" import "svm", that is it (53).

**So, in the introduction we talked about deep learning which basically depend on ANN so let us illustrate Artificial Neural Network.**

**ARTIFICIAL NEURAL NETWORK (ANN)**

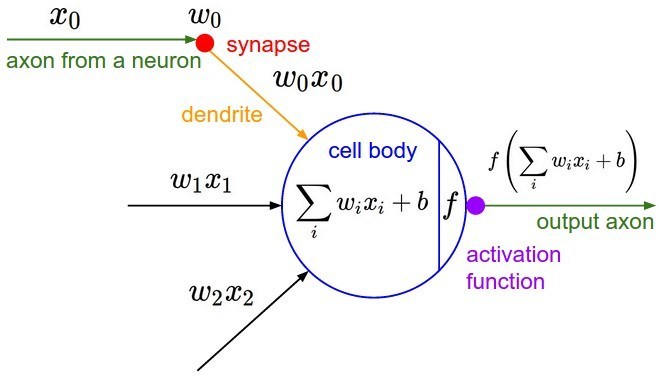
Basically, ANN is a simulation to how our brains work, neural network consists of many interconnected neurons that connects to other neurons exactly like the biological neuron in our brain (54).

**Brief comparison between a biological neuron and ANN**:



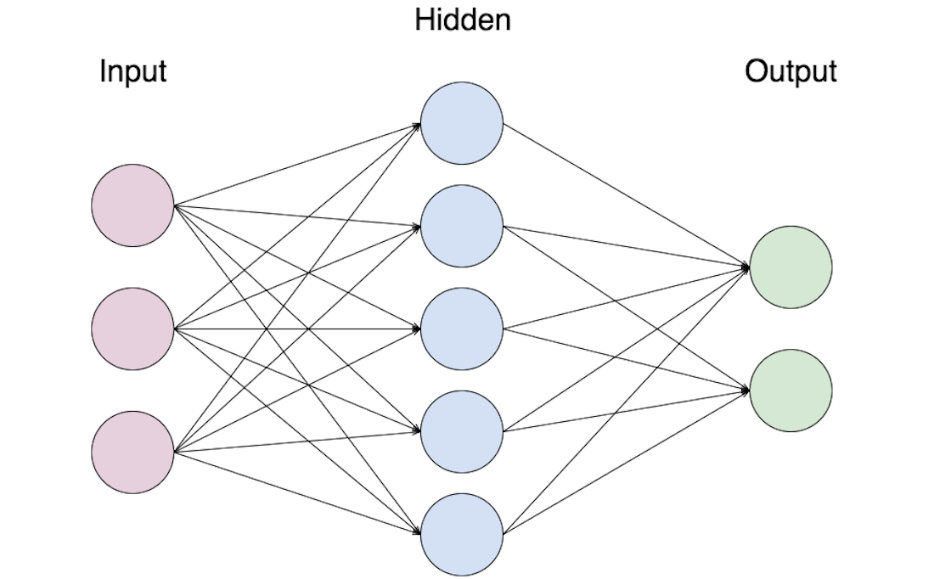
In the first perspective the both of two neurons have many similarities, you can see that each on considers as a small part of a huge operation processing and each one has its role, second both of them has input and output, the third thing that the both of them connect to its neighbors, but for sure if you looked deeply you can find a very big differences like the biological one works with chemicals and electricity, in in contrast the second one is just about some algorithms and note that these algorithms and equations simply are transistors, the problem here is that the transistor which consists of some logic gats can connect with at most three other transistors, in contrast the biological neuron can connect with millions of neurons and this what make the biological neurons very fast and amazing compared to ANN (55).

Let us look deeply in a single neuron (the perceptron):



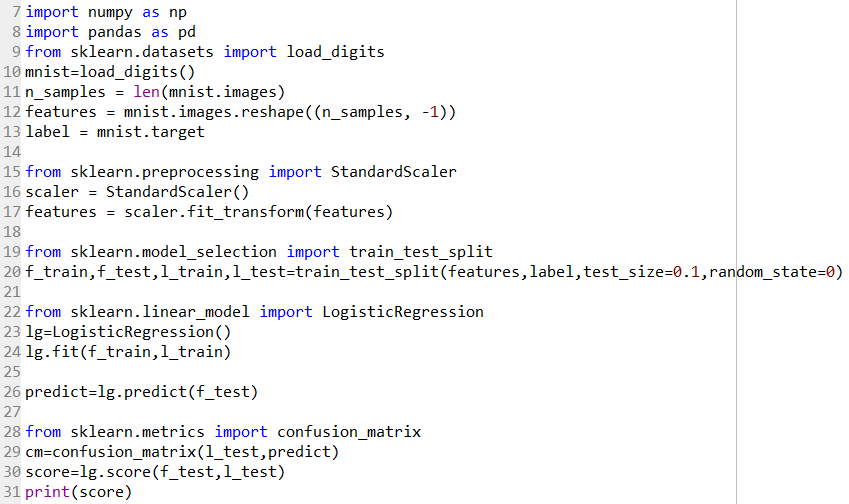
We can call it perceptron or a single lyre neural network which is the simplest model of ANN, first of all, from the figure above we can see (w) which express the **weight**, actually every input has a specific weight multiplied with, we can say that each weight expresses the probability of Xi (input) to be the output, if a particular weight was less than 1 it will weakens its input, in contrast if was greater than one it will amplify its input, also there is "b" which we call it the bias, its work is to fixed the weight in the algorithm, second thing we can note is the **activation function**, what does the activation function do is that it determine the output of perceptron, the concept of the activation function similar to the threshold of the nerve impulse in biological neuron, suppose I put my finger in hot water, here the hot water will cause a strong effect because it exceed the threshold of nerve impulse and this make the neuron get fire to the brain to generates the feeling, unlike if there a small fly set on my body, I will not feel it because its effect doesn't exceed the threshold of nerve impulse, similarly the activation function give you either one or zero and this is depend on the threshold in your function, there are many function we can use in this level, one of the most popular function is sigmoid function which we explained before (55).

Now in more complex form the ANN has at least three layers: input, output and between them there is at least one hidden layer, we said before that deep learning basically built on ANN so actually when we say "deep" we mean that our ANN has at least 10 hidden layers, yes but the "deep" term change by the time, now days it is sufficient to identify your network as deep if you have 100 hidden layers (41).

**This is how ANN look like:**

**IMPLEMENTATION**

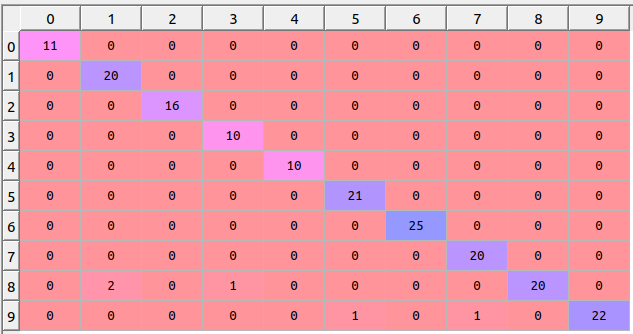
**The implementation with MNIST dataset**

**1-Using Logistic Regression Algorithm:**

The result is:

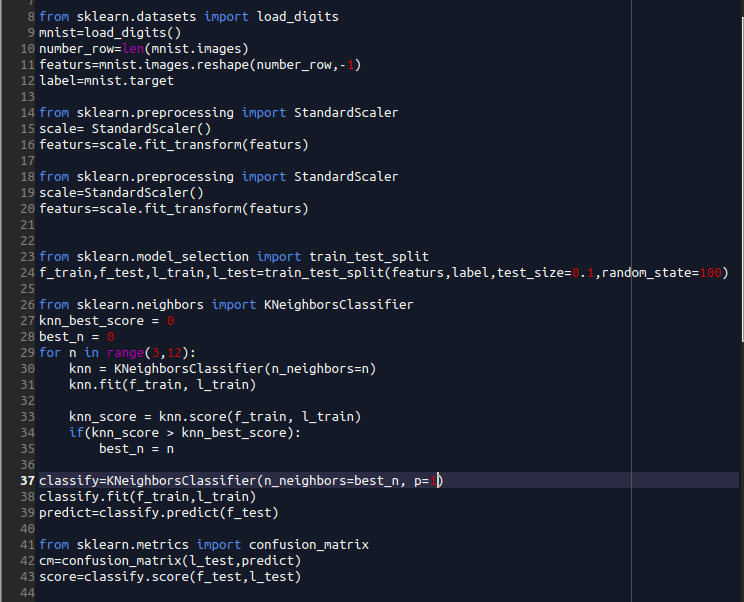


The Confusion Matrix for Logistic Regression is:



**2-Using KNN algorithm**

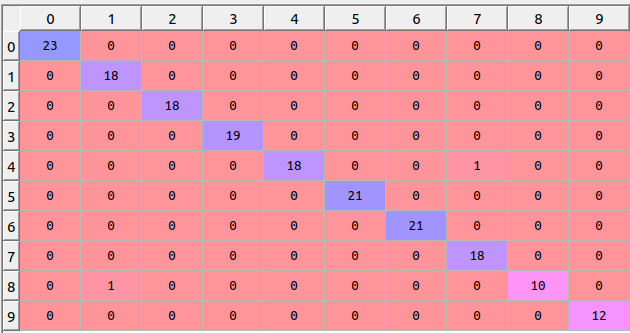
Using p = 1



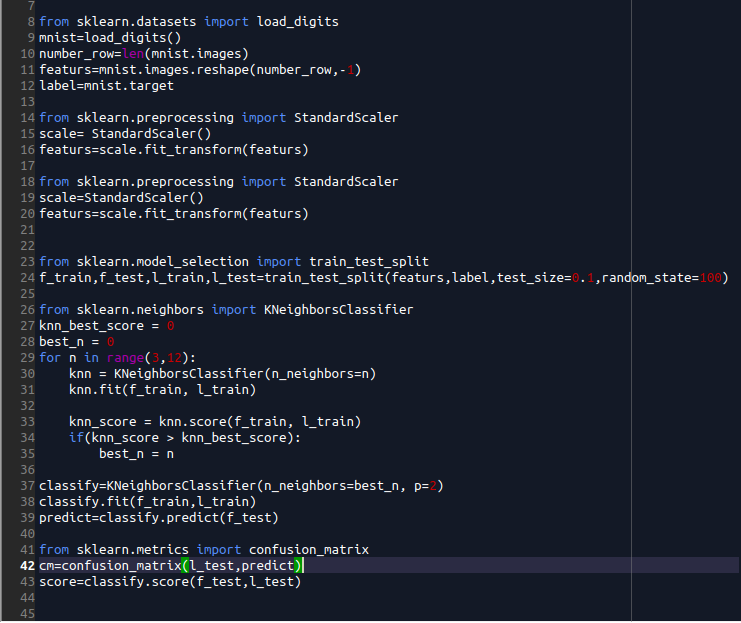
The Score is:



The Confusion Matrix is:

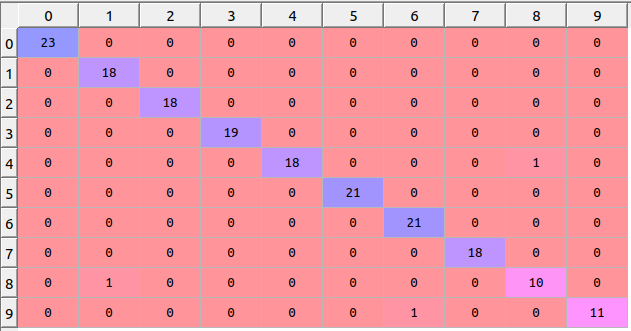


Using p = 2



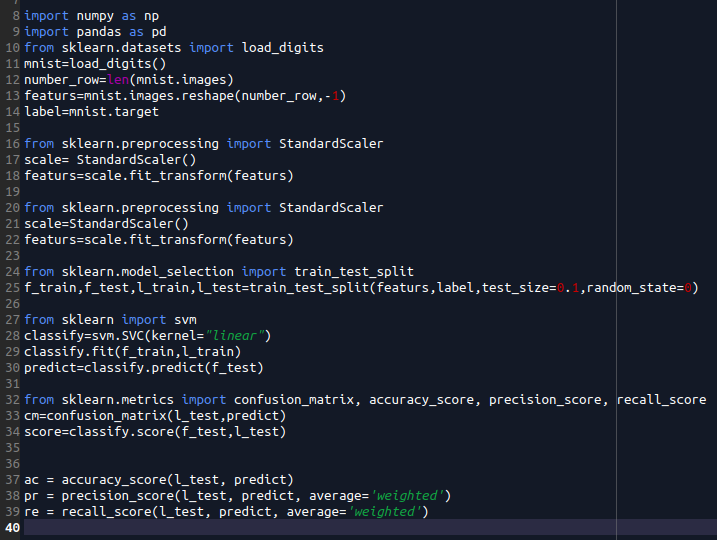
The Score is:

The Confusion Matrix is:



**3-Using SVM Algorithm:**

1- Using Linear Kernel



**The result is:**

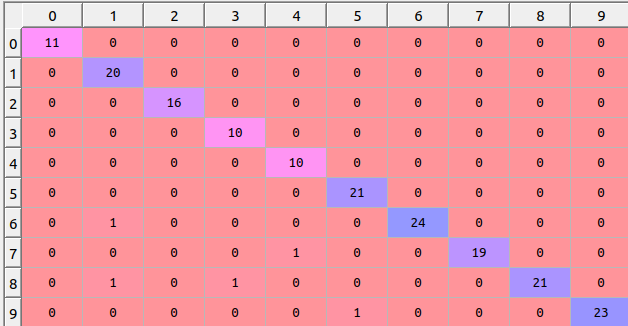
**score** : 0.9722

**accuracy**: 0.9722

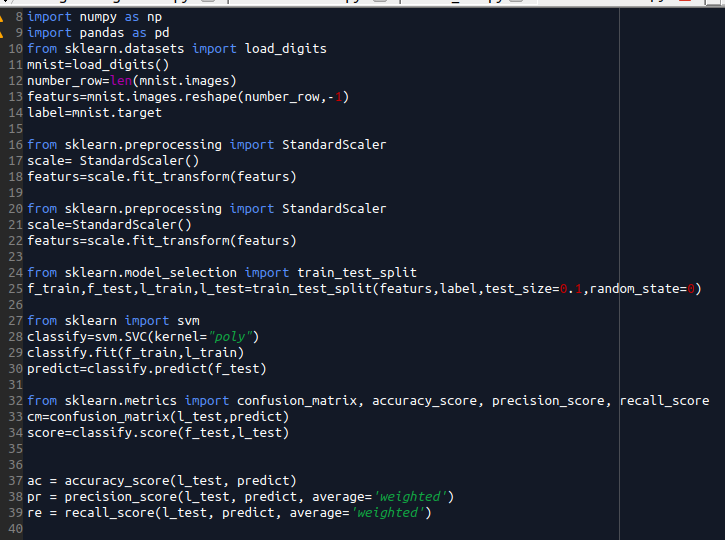
**precision**: 0.9744

**recall**: 0.9722

**The Confusion matrix is:**



2- Using Polynomial Kernel



**The result is:**

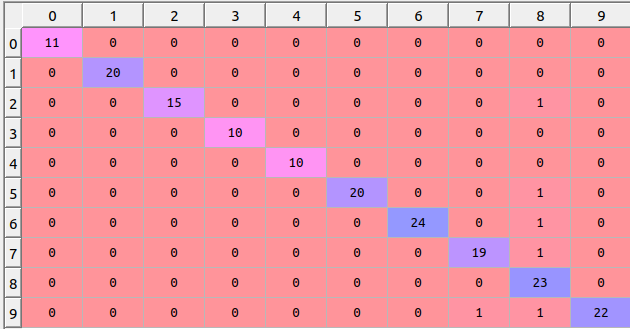
**score** : 0.9667

**accuracy**: 0.9667

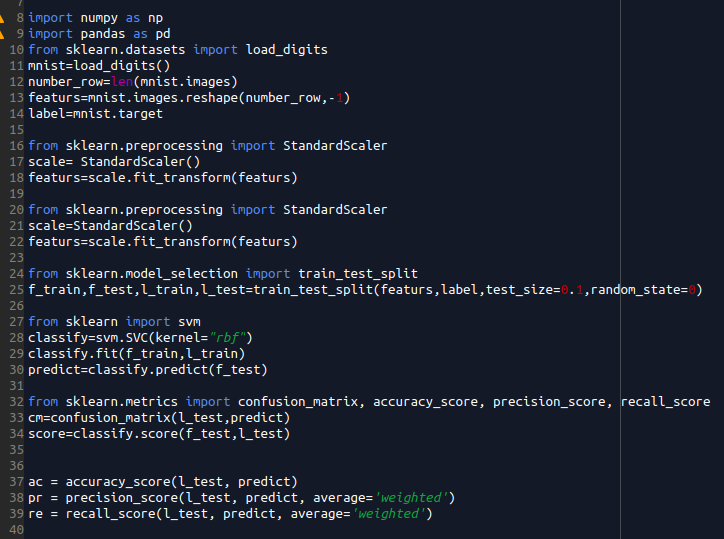
**precision**: 0.9716

**recall**: 0.9667

**The Confusion matrix is:**



3- Using RBF Kernel



**The result is:**

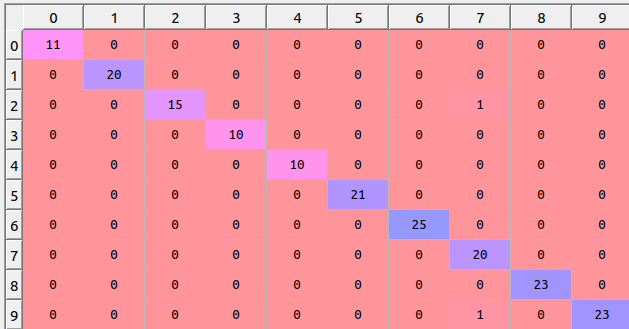
**score** : 0.9888

**accuracy**: 0.9888

**precision**: 0.9898

**recall**: 0.9888

**The Confusion matrix is:**



**COMPARISON**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Score | Accuracy | Recall | Precision |
| Logistic Regression |  | 0.9722 | 0.9722 | 0.9742 |
| SVM- linear | 0.9722 | 0.9722 | 0.9722 | 0.9744 |
| SVM-RBF | 0.9888 | 0.9888 | 0.9888 | 0.9898 |
| SVM-poly | 0.9667 | 0.9667 | 0.9667 | 0.9716 |
| KNN | 0.9833 | 0.9833 | 0.9833 | 0.9838 |
| KNN-Manhattan | 0.9889 | 0.9889 | 0.9889 | 0.9894 |

**REFERENCES**

(1) python programming using problem solving approach book by REEMA THAREJA page: 63,83,84,85,87

(2) <http://net-informations.com/python/iq/disadvantages.htm>

(3) <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>

(4) statistic for using the programming language for 2018, so you can compare it with 2019: <https://hackernoon.com/top-3-most-popular-programming-languages-in-2018-and-their-annual-salaries-51b4a7354e06>

(5) <https://docs.scipy.org/doc/numpy/user/whatisnumpy.html>

(6) similar to the example in this website:

<https://webcourses.ucf.edu/courses/1249560/pages/python-lists-vs-numpy-arrays-what-is-the-difference>

(7) <https://www.youtube.com/watch?v=xECXZ3tyONo> <https://docs.scipy.org/doc/numpy/user/quickstart.html>

(8) <https://pandas.pydata.org/>

(9) ma'am ANUJA lecture in pandas library

(10) <https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/>

(11) <https://www.datasciencelearner.com/top-steps-data-preprocessing-machine-learning/>

(12)<https://www.c-sharpcorner.com/article/best-programming-language-for-machine-learning/>

(13) dataset form my randomly made only to illustration

(14) <https://campus.datacamp.com/courses/preprocessing-for-machine-learning-in-python/introduction-to-data-preprocessing?ex=1>

(15) <https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d>

(16) <https://stackoverflow.com/questions/22391433/count-the-frequency-that-a-value-occurs-in-a-dataframe-column>

(17) <https://www.geeksforgeeks.org/python-pandas-dataframe-idxmax/>

(18) <https://campus.datacamp.com/courses/preprocessing-for-machine-learning-in-python/introduction-to-data-preprocessing?ex=4>

(19) <https://campus.datacamp.com/courses/preprocessing-for-machine-learning-in-python/introduction-to-data-preprocessing?ex=7>

(20) look at the summary <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>

(21) it is a logistic regression code from my supervisor ANUJA Ma'am's E-mail

(22) <https://www.kaggle.com/spidy20/simple-data-for-preprocessing/downloads/simple-data-for-preprocessing.zip/1>

(23) [https://stackoverflow.com/questions/32108179/linear-regression-normalization-vs-standardization/32110985#32110985](https://stackoverflow.com/questions/32108179/linear-regression-normalization-vs-standardization/32110985" \l "32110985)

(24) <https://medium.com/@contactsunny/handle-missing-data-in-your-training-dataset-with-scikit-imputer-f4bbf005fc5e>

(25) <https://datascience.stackexchange.com/questions/12321/difference-between-fit-and-fit-transform-in-scikit-learn-models>

(26) <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>

(27) <https://towardsdatascience.com/choosing-the-right-encoding-method-label-vs-onehot-encoder-a4434493149b>

(28) [https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#10a164106f63](https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/" \l "10a164106f63))

(29) artificial-intelligence-modern-approach BY Stuart J. Russell and Peter Norvig /page: 5

(30) <https://aidaily.news/an-incredibly-cool-tree-of-artificial-intelligence-ai-and-its-subtopics/>

(31) <https://www.expertsystem.com/machine-learning-definition/>

(32) <https://www.edureka.co/blog/what-is-machine-learning/>

(33) <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>

(34) <https://medium.com/deep-math-machine-learning-ai/different-types-of-machine-learning-and-their-types-34760b9128a2>

(35) <https://www.geeksforgeeks.org/ml-linear-regression/>

(36) <https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/>

(37) <https://www.kaggle.com/tmcketterick/heights-and-weights/downloads/heights-and-weights.zip/1>

(38) <https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/>

(39) <https://data-flair.training/blogs/machine-learning-applications/>

(40) <https://www.credosystemz.com/courses/machine-learning-training-chennai/>

(41) Deep learning with keras book by ANTONIO GULI & SUJIT PAL /page: 2,3

(42) <https://machinelearningmastery.com/what-is-deep-learning/>

(43) <https://en.wikipedia.org/wiki/MNIST_database>

(44) <https://www.edureka.co/blog/what-is-deep-learning>

(45) <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>

(46) <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>

(47) <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>

(48) <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

(49) <https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/>

(50) <https://mccormickml.com/2013/08/15/the-gaussian-kernel/>

(51) <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

(52) <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

(53) <https://scikit-learn.org/stable/modules/svm.html>

(54) <https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf>

(55) <https://jasonroell.com/2017/06/12/from-fiction-to-reality-a-beginners-guide-to-artificial-neural-networks/>

(56) <https://medium.com/@jamesdacombe/an-introduction-to-artificial-neural-networks-with-example-ad459bb6941b>

(57)

(58)

(59)

(60)