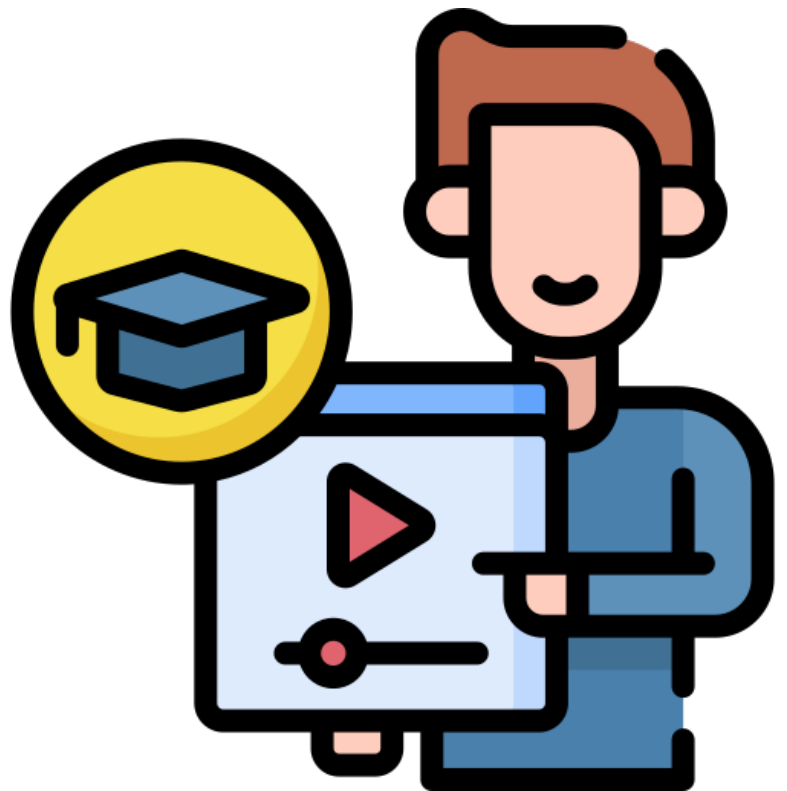
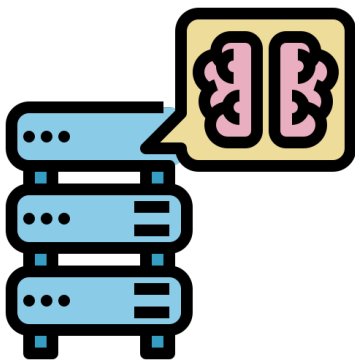


20/07/2020

Projet machine Learning diabète



Présenté par
BOULENOUAR Réda
Et ABBASSI Abdelilah

REMERCIEMENT

Avant d'entamer ce rapport, nous tiendrons à témoigner nos profondes gratitude à toutes les Personnes qui ont participé de loin ou de près à l'élaboration de ce travail. Nous tenons à offrir nos sincères remerciements et exprimer nos profonde gratitude à notre professeur **MAHMOUDI Abdelhak**, pour le temps précieux qu'il nous a consacré, pour son aimable disponibilité sans réserve, pour ses conseils et son aide durant toute la période de notre projet. Veuillez croire à l'expression de notre grand respect et notre grande reconnaissance.

Table des matières

REMERCIEMENT.....	1
Introduction.....	3
Chapitre I Le Machine Learning :	4
Introduction:	4
Definition :	4
L'apprentissage supervisé:.....	6
Régression :	7
Classification :	7
Chapitre II Réalisation du projet :.....	10
Méthodologie de conduite du projet :	10
Réalisation du projet :	15
Régression Linear :.....	16
Code :.....	17
Régression Logistique :	20
Conclusion :	25

Introduction

Un flux massif de données dans un format structuré, non structuré ou hétérogène a été accumulé en raison de l'augmentation continue du volume et des détails des données saisies par les organisations, Ces quantités massives de données sont produites en raison de la croissance du Web, de l'essor des médias sociaux, de l'utilisation du mobile et de IOT par et au sujet des personnes, des choses et de leurs interactions. L'ère du Big Data est arrivée.

Le Machine Learning est idéal pour exploiter les opportunités cachées du Big Data. Cette technologie permet d'extraire de la valeur en provenance de sources de données massives et variée. Plus les données injectées à un système Machine Learning sont nombreuses, plus ce système peut apprendre et appliquer les résultats à des insights de qualité supérieure.

Chapitre I :

Le Machine Learning

Introduction:

Les outils analytiques traditionnels ne sont pas suffisamment performants pour exploiter pleinement la valeur du Big Data. Le volume de données est trop large pour des analyses compréhensives, et les corrélations et relations entre ces données sont trop importantes pour que les analystes puissent tester toutes les hypothèses afin de dégager une valeur de ces données.

Définition :

Le Machine Learning, aussi appelé apprentissage automatique en français, est une forme d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Cette technologie permet de développer des programmes informatiques pouvant changer en cas d'exposition à de nouvelles données.

Le Machine Learning est une méthode d'analyse de données permettant d'automatiser le développement de modèle analytique. Par le biais d'algorithmes capables d'apprendre de manière itérative.

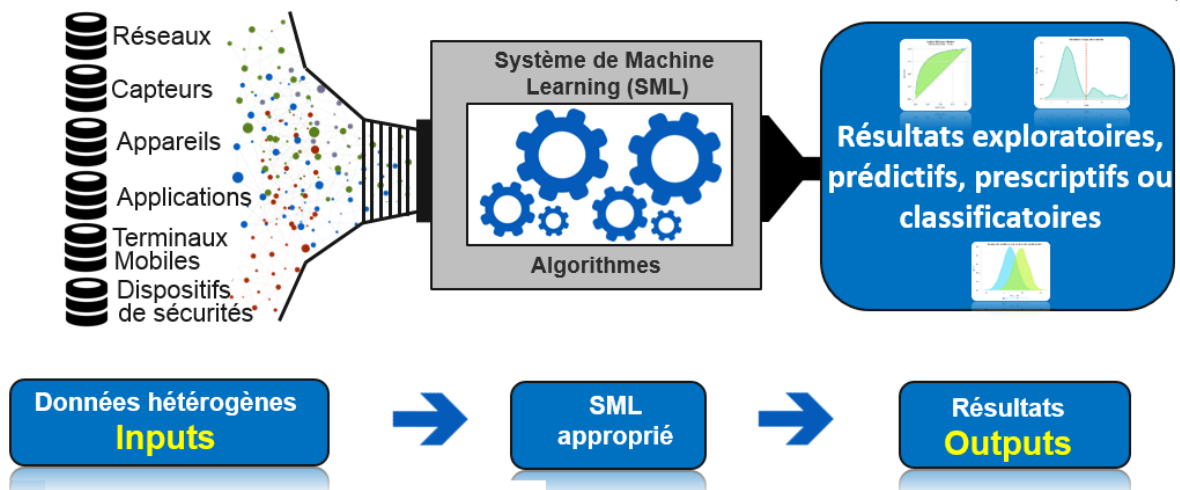


Figure 1 : Système de Machine learning

1) Pourquoi le Machine Learning n'est rien sans Big Data :

Sans le Big Data, le Machine Learning et l'intelligence artificielle ne seraient rien. Les données sont l'instrument qui permet à l'IA de comprendre et d'apprendre à la manière dont les humains pensent. C'est le Big Data qui permet d'accélérer la courbe d'apprentissage et permet l'automatisation des analyses de données. Plus un système Machine Learning reçoit de données, plus il apprend et plus il devient précis.

2) Les secteurs d'application de Machine Learning :

Le Machine Learning est utilisé dans de nombreuses industries, et même dans les domaines créatifs comme la peinture ou le cinéma. Les entreprises ont compris l'avantage compétitif procuré par la capacité de collecter des informations en temps réel à partir de données.

- Services financiers
- Gouvernement
- Santé
- Marketing
- Gaz et pétrole
- Transports

3) Les principales méthodes de Machine Learning :

Les **deux méthodes de Machine Learning les plus couramment utilisées** sont **l'apprentissage supervisé & l'apprentissage non supervisé**.

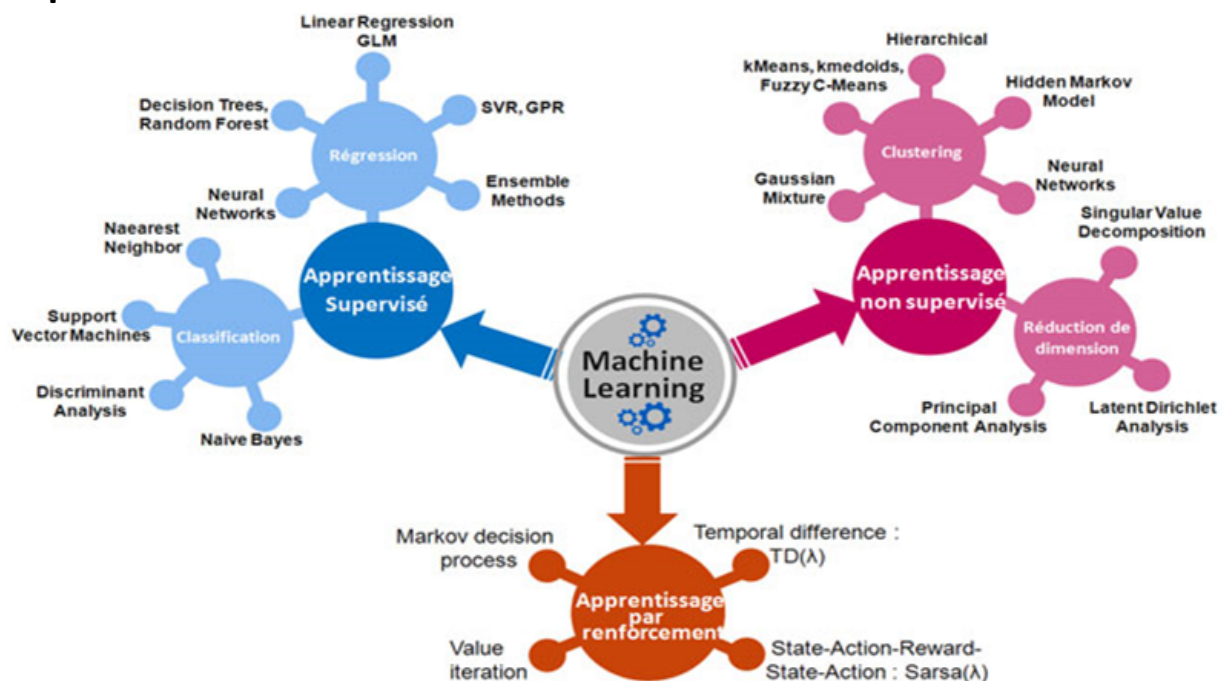


Figure 5 : méthodes de Machine Learning

L'apprentissage supervisé:

Les algorithmes d'apprentissage supervisé sont entraînés à l'aide d'exemples étiquetés. Par L'algorithme reçoit un ensemble d'inputs ainsi que les outputs corrects correspondants, et apprend en comparant les outputs avec les résultats corrects attendus pour détecter les erreurs. Il modifie ensuite son modèle en fonction. Les méthodes comme la classification, la régression, permettent à

l'apprentissage supervisé d'utilisé des patterns pour prédire la valeur d'une étiquette ou d'une donnée additionnelle sans étiquette. Cette méthode d'apprentissage est couramment utilisée dans les applications où les données historiques permettent de prédire les événements futurs.

L'apprentissage supervisé est certainement la forme de machine learning la plus répandue. Elle consiste à entraîner le programme sur des exemples dont on connaît la catégorie. Il en existe deux types :

Régression :

La Target est de type numérique continu. Les sciences exactes sont fondées sur la notion de relations répétables, qui peut s'énoncer ainsi: dans les mêmes conditions, les mêmes causes produisent les mêmes effets.

La régression prédit une valeur numérique basée sur les données observées précédemment.

Par exemple: prévision du prix des maisons, prévision du prix des actions, prédiction taille-poids.

Classification :

La classification consiste à déterminer la catégorie d'un objet possédant certaines caractéristiques. La classification est utilisée pour prédire les réponses discrètes. Dans l'apprentissage supervisé, les algorithmes tirent des enseignements de données étiquetées. Après avoir compris les données, l'algorithme détermine l'étiquette à attribuer aux nouvelles données en fonction du modèle et en associant les modèles aux nouvelles données non étiquetées. La classification prédit la catégorie à laquelle les données appartiennent.

Par exemple: détection de spam, prévision du taux de désabonnement, analyse des sentiments,

- L'apprentissage non supervisé:

Est utilisé pour les données qui n'ont pas d'étiquettes historiques. Le système ne connaît pas la réponse correcte, et l'algorithme doit comprendre par lui-même ce qui lui est présenté. L'objectif est d'explorer les données et de trouver une structure en leur sein. Cette méthode fonctionne bien pour les données de transaction.

Par exemple : identifier les segments des consommateurs, segmenter les textes, recommander des produits.

Il existe deux principales méthodes d'apprentissage non-supervisées

La forme la plus répandue d'apprentissage non supervisé est le clustering, elle est similaire à la classification, mais la base est différente. Dans le clustering, nous ne savons pas ce que nous recherchons et nous essayons d'identifier certains segments ou clusters dans nos données. Lorsque nous utilisons des algorithmes de clustering sur notre jeu de données, des événements inattendus peuvent soudainement apparaître, tels que des structures, des clusters et des groupements auxquels nous n'aurions jamais pensé autrement.

- L'apprentissage de renforcement:

Est souvent utilisé pour la robotique, le jeu vidéo et la navigation.

Grâce à l'apprentissage de renforcement, l'algorithme multiplie les tentatives pour tenter de découvrir quelles actions apportent les plus grandes récompenses.

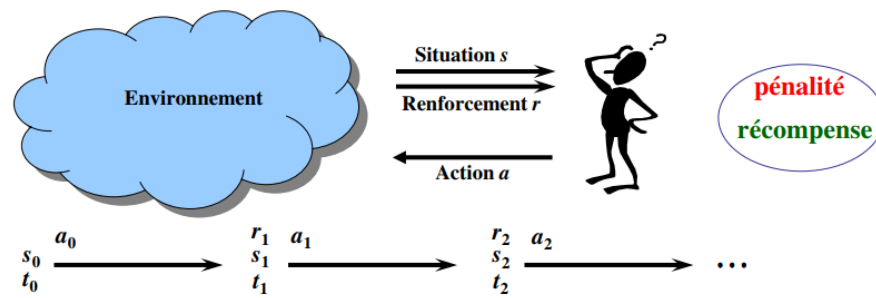


Figure 2 : apprentissage de renforcement

L'agent apprend à se rapprocher d'une stratégie comportementale optimale par des interactions répétitives avec l'environnement, les décisions sont prises séquentiellement à des intervalles de temps discrets.

L'objectif de l'apprentissage de renforcement est donc d'apprendre les meilleures règles.

Conclusion :

Le machine learning est un outil très puissant qui permet d'effectuer de multiples actions comme classifier des données, faire apprendre à un programme à partir d'expérimentations ou encore de créer un programme évolutionnaire qui s'améliore sans cesse.

Dans le prochain chapitre on va parler sur :

L'apprentissage supervisé:

- Régression :
 - ✓ la régression linéaire simple.
 - ✓ la régression linéaire multiple.
 - ✓ la régression linéaire polynomiale.
- Classification :

- ✓ Régression logistique
- ✓ SVM

Chapitre II :

Réalisation du projet

Méthodologie de conduite du projet :

- 1) Les éléments clés d'un projet de Machine Learning réussi :

Pour mener un projet de Machine Learning efficace, voici les 8 étapes à suivre :

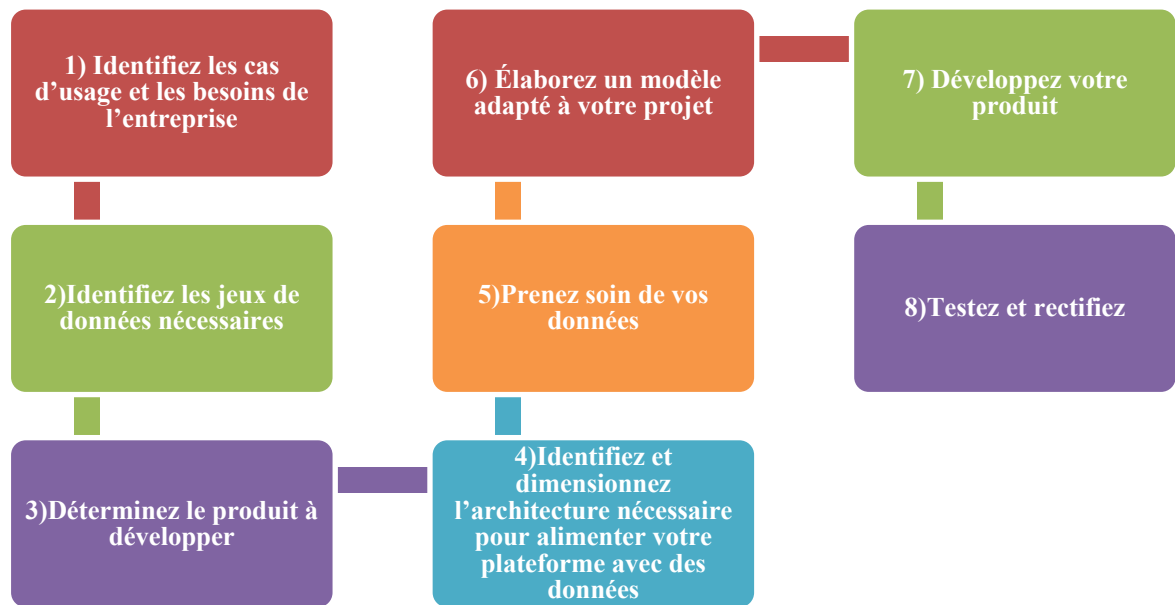


Figure 3 : Les éléments clés d'un projet de Machine Learning

Actuellement, grâce à l'accessibilité des technologies et à l'abondance des données, le Machine Learning est à la portée de toutes les entreprises, toutes tailles et tous secteurs d'activité confondus. Mais le secret de la réussite d'un projet de Machine Learning consiste à adopter une approche judicieuse dès le départ. Une perspective axée sur la dimension métier (plutôt que sur l'aspect technique) permet d'identifier et d'articuler les éléments du projet (compétences, données et réalisation) d'une façon adaptée.

2) Le problématique de notre mini projet de Machine Learning est:

Prédire si un patient est diabétique ou pas

3) Les phases d'un projet de Machine Learning:

Un projet Machine Learning ce n'est pas un projet de développement classique et il a donc ses propres contraintes mais surtout il aura besoin d'une grande souplesse et de réajustements réguliers.

Réussir son projet de Machine Learning revient à respecter les étapes ci-dessous:

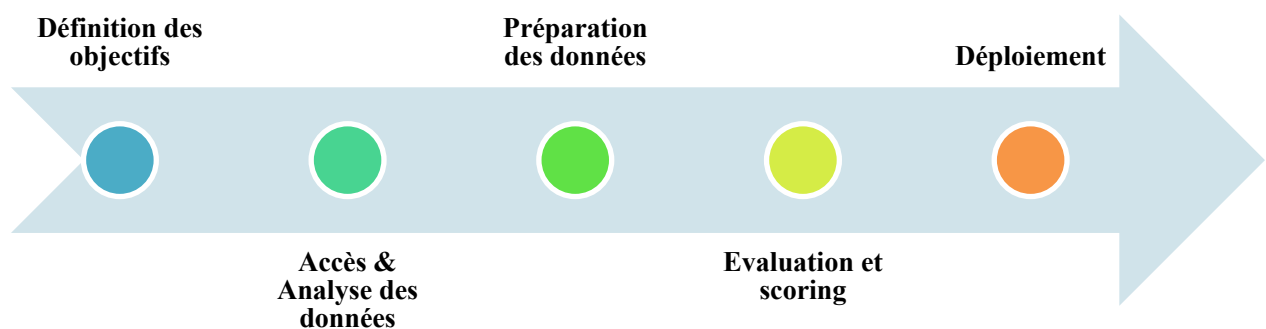


Figure 4 : Les phases d'un projet de Machine Learning

a) Définition des objectifs :

Il s'agit ici de déterminer quelle typologie de problème nous devons résoudre. Pour cela nous devons savoir si nous avons des données d'expérimentation avec résultat ou non, afin de déterminer si nous abordons un problème de type supervisé ou non-supervisé.

Ensuite quelle est la typologie du problème à résoudre (Régression, Classification, Clustering...).

b) Accès au jeu de données :

Une étape cruciale dans laquelle on va retravailler les données (features ou variables). C'est une opération indispensable car les algorithmes de Machine Learning n'acceptent pas tout type de données. C'est une opération nécessaire afin d'affiner les variables pour qu'elles soient mieux gérées par ces mêmes algorithmes.

Importer les librairies :

La première étape consiste à importer les librairies qui nous seront utiles pour la construction de notre modèle. Dans notre cas, nous travaillerons avec trois librairies complètement indispensables pour le déroulement de notre projet, il s'agit de :

- **NUMPY** : qui nous permet d'utiliser des maths dans notre code, et ainsi faire de gros calculs avec Python.
- **PANDAS** : cette librairie est la meilleure concernant la gestion et le traitement des Datasets.
- **MATPLOTLIB** : pour visualiser nos résultats sous forme de graphs (Courbes, Diagrammes, Histogrammes...). Nous utiliserons plus précisément le module « **PYLOT** » qui nous sera très utile pour visualiser des graphs en 2D.

Importer le Dataset:

Pour importer notre Dataset, il faut l'affecter à une variable qui va le représenter tout au long du déroulement du projet.

Pour affecter le Dataset à la variable créée, il nous suffit d'utiliser soit le nom de la librairie (pandas), ou bien la méthode d'importation « `read_csv (..)` ».

c) Préparation des données :

Analyse & préparation des données

L'étape N°2 permettant de faire un état des lieux complet des données dont on dispose,

C'est une étape toute aussi importante dans laquelle nous allons préparer nos features/variables afin qu'elles soient utilisables par des algorithmes de Machine Learning.

Les tâches principales du prétraitement des données sont les suivantes:

1. Nettoyage des données: renseignez les valeurs manquantes, adoucissez les données bruyantes, identifiez ou supprimez les valeurs éloignées et résolvez les incohérences
2. Intégration de données : Intégration de plusieurs bases de données, cubes de données ou fichiers.
3. Transformation de données: normalisation et agrégation.
4. Réduction des données: représentation en volume réduite, mais résultats analytiques identiques ou similaires.
5. Discrétisation des données : fait partie de la réduction des données, mais revêt une importance particulière, en particulier pour les données numériques.

Réalisation du projet :

1) Outils et langages utilisés :



Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets anaconda. La Distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs, et il comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOS.



Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Julia, Python, R, Ruby ou encore Scala. Jupyter permet de réaliser des notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces notebooks sont utilisés en science des données pour explorer et analyser des données.






Un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

2) Réalisation du projet :

Pour notre projet on va travailler pour trouver des solutions aux problématiques cités ci-après :

 Prédire si un patient est diabétique ou pas

Régression Linear :

Qu'est-ce que la régression Linéaire multiple?

Le but de la régression linéaire multiple est le même que celui de la régression simple excepté le fait que plusieurs variables indépendantes entrent en jeu cette fois ci. D'où l'équation :

$$Y = a + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n.$$

Travaux pratiques :

Dataset :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

Dataset: la table contient toutes les valeurs de notre fichier csv.

Pourquoi la régression Linéaire multiple ?

La variable indépendante est de type numérique continue, et
 Nous avons plusieurs variables dépendantes, donc l'algorithme
 convenable est « l'algorithme de régression linéaire multiple ».

Code :

Multivariate Linear Regression

```
In [21]: # To make debugging of linear_regression module easier we enable imported modules autoreloading feature.
# By doing this you may change the code of linear_regression library and all these changes will be available here.
%load_ext autoreload
%autoreload 2
```

The autoreload extension is already loaded. To reload it, use:
`%reload_ext autoreload`

Import Dependencies

- [pandas](#) - library that we will use for loading and displaying the data in a table
- [numpy](#) - library that we will use for linear algebra operations
- [matplotlib](#) - library that we will use for plotting the data
- [plotly](#) - library that we will use for plotting interactive 3D scatters
- [linear_regression](#) - custom implementation of linear regression

```
In [31]: # Import 3rd party dependencies.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Load the Data

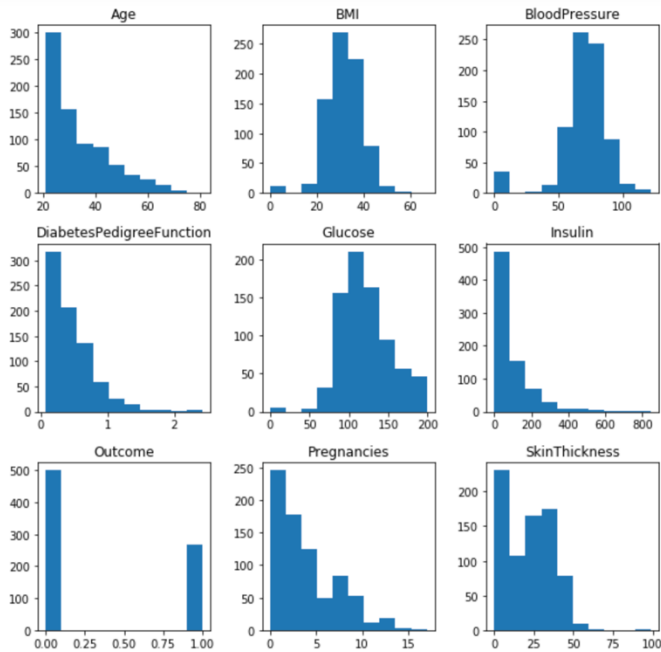
In this demo we will use [World Hapindes Dataset](#) for 2017.

```
In [32]: # Load the data.
data = pd.read_csv('diabetes.csv')
# Print the data table.
data.head(10)
```

Out[32]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

```
In [33]: # Print histograms for each feature to see how they vary.
histohrams = data.hist(grid=False, figsize=(10, 10))
```



```
In [34]: x=data.iloc[:, :-1].values
```

```
In [35]: y=data.iloc[:, 4].values
```

```
In [37]: from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
x[:, 3] = labelencoder.fit_transform(x[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
x = onehotencoder.fit_transform(x).toarray()
x = x[:, 1:]
```

```
In [39]: # splitting the dataset into the training set and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state = 0)
```

```
In [41]: # importing the linear regression class
from sklearn.linear_model import LinearRegression
```

```
In [43]: #creating an object of the linear regression class
regressor = LinearRegression()
#fit the created object to our training set
regressor.fit(x_train, y_train)
```

```
Out[43]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [44]: # predicting the test set results
y_pred = regressor.predict(x_test)
```

```
In [46]: #to see the difference btwn predicted and actual result we will also print the test value
print(y_test)
```

```
In [45]: #printing the predicted values
print(y_pred)
```

b) Classification :

Régression Logistique :

L'utilisateur du réseau social va acheter la voiture ?

Qu'est-ce que la régression logistique ?

La régression logistique ressemble un peu à la régression linéaire, mais elle est utilisée lorsque la variable dépendante n'est pas un nombre, mais quelque chose d'autre (comme une réponse Oui / Non). Elle s'appelle Régression mais effectue la classification en fonction de la régression et classe la variable dépendante dans l'une ou l'autre des classes.

Travaux pratiques :

On a essayé de créer une petite application web codée en python qui va nous permettre de prédire Si un patient donne est diabétique ou non pour cela on va procéder à une classification

Dataset : Diabète

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

Code:

```

import pandas as pd

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from PIL import Image

import streamlit as st

df=pd.read_csv('/Users/mac/Desktop/Bureau - MacBook
Pro/Master IDDL/Machine Learning/Priject
Diabete/diabetes.csv')

st.subheader('Data Information : ')

st.dataframe(df)

st.subheader('Data type : ')

df.dtypes

```

```

st.write(df.describe())

chart = st.bar_chart(df)

x=df.iloc[:,0:8].values

y=df.iloc[:,-1].values

X_train,X_test,Y_train,Y_test=
train_test_split(x,y,test_size=0.25,random_state=0)

def get_user_input():

    Pregnancies =st.sidebar.slider('pregnacies',0,17,3)

    Glucose =st.sidebar.slider('Glucose',0,199,117)

    BloodPressure =st.sidebar.slider('BloodPressure',0,122,72)

    SkinThickness =st.sidebar.slider('SkinThickness',0,99,23)

    Insulin =st.sidebar.slider('Insulin',0.0,846.0,30.0)

    BMI =st.sidebar.slider('BMI',0.0,67.1,32.0)

    DiabetesPedigreeFunction
=st.sidebar.slider('DiabetesPedigreeFunction',0.078,2.42,0.3725
)

    Age =st.sidebar.slider('Age',21,81,29)

    user_data={

        'Pregnancies':Pregnancies,

        'Glucose':Glucose,

        'BloodPressure':BloodPressure,

        'SkinThickness':SkinThickness,

        'Insulin':Insulin,

        'BMI':BMI,

        'DiabetesPedigreeFunction':DiabetesPedigreeFunction,

```

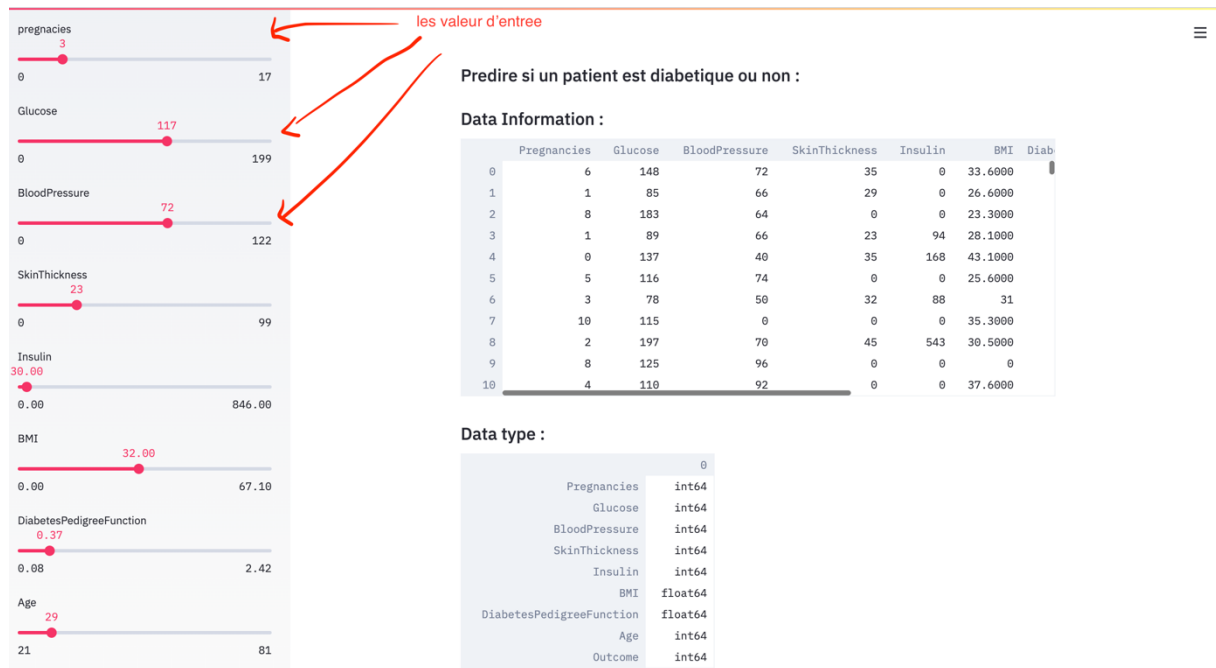
```
'Age':Age
}

features = pd.DataFrame(user_data,index=[0])
return features

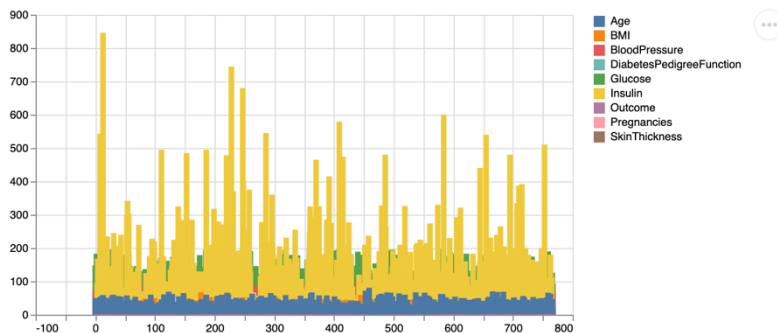
user_input =get_user_input()
st.subheader('User Input : ')
st.write(user_input)

RandomForestClassifier =RandomForestClassifier()
RandomForestClassifier.fit(X_train,Y_train)
st.subheader('Model Test Accuracy score : ')
st.write(str(accuracy_score(Y_test,RandomForestClassifier.predict(X_test))*100)+'%')

prediction =RandomForestClassifier.predict(user_input)
st.subheader('Classification')
st.write(prediction)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
count	768	768	768	768	768	768
mean	3.8451	120.8945	69.1055	20.5365	79.7995	31.9926
std	3.3696	31.9726	19.3558	15.9522	115.2440	7.8842
min	0	0	0	0	0	0
25%	1	99	62	0	0	27.3000
50%	3	117	72	23	30.5000	32
75%	6	140.2500	80	32	127.2500	36.6000
max	17	199	122	99	846	67.1000



User Input :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPer
0	3	117	72	23	30	32	

Model Test Accuracy score :

79.16666666666666%

Classification Accuracy

Classification

0
0 0

0 diabetique 1 non

Conclusion :

Ce projet nous a permis de découvrir l'univers de la science des données et d'approfondir et mettre en pratique nos connaissances dans le domaine de l'apprentissage automatique.

Pour mettre en œuvre ce projet nous avons été amenés à faire une étude sur le machine Learning.

Enfin, l'apprentissage automatique est un domaine qui évolue d'une façon continue. et il faut garder en tête les deux limites cités ci-après:

- Il n'existe pas d'algorithme et modèle "ultime", applicable pour tous les problèmes. donc il faut aborder chaque nouveau problème avec un œil neuf et veiller à tester plusieurs algorithmes afin de le résoudre, en formulant des hypothèses spécifiques à ce problème.
- Il arrive souvent de se trouver confronté à des algorithmes et modèles très puissants... mais trop compliqués pour être utilisés directement. Pour éviter ces situations, il ne faut pas avoir peur d'effectuer des approximations qui permettent de gagner en efficacité.