Breast Cancer Risk Prediction System

1)INTRODUCTION:

1.1 overview :

Breast cancer is one of the main causes of cancer death worldwide. Early diagnostics significantly increases the chances of correct treatment and survival, but this process is tedious and often leads to a disagreement between pathologists. Computer-aided diagnosis systems showed the potential for improving diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible.

1.2 purpose :

We will be building a model in Watson Studio and deploying the model in IBM Watson Machine Learning. To interact with the model we will be using Node-Red and scoring Endpoint. Develop a model that is capable of detecting the Breast Cancer in early stages. The Machine learning model is trained and deployed on IBM Watson Studio and an endpoint is created. The web application is built using IBM Node-Red.

2) LITERATURE SURVEY

2.1 Existing problem :

Cancer is clearly the most deadly disease in the developed world as one in three people develop cancer during their lifetime. The cure for cancer is like the Holy Grail since most of the existing treatments are not effective enough to provide full protection from this disease.

Cancer is clearly the most deadly disease in the developed world as one in three people develop cancer during their lifetime. The cure for cancer is like the Holy Grail since most of the existing treatments are not effective enough to provide full protection from this disease

For instance the treatments of leukemia and lymphoma have been established and proved to be satisfactory. Despite occasional successes the treatment for most cancers is still a long way from reality

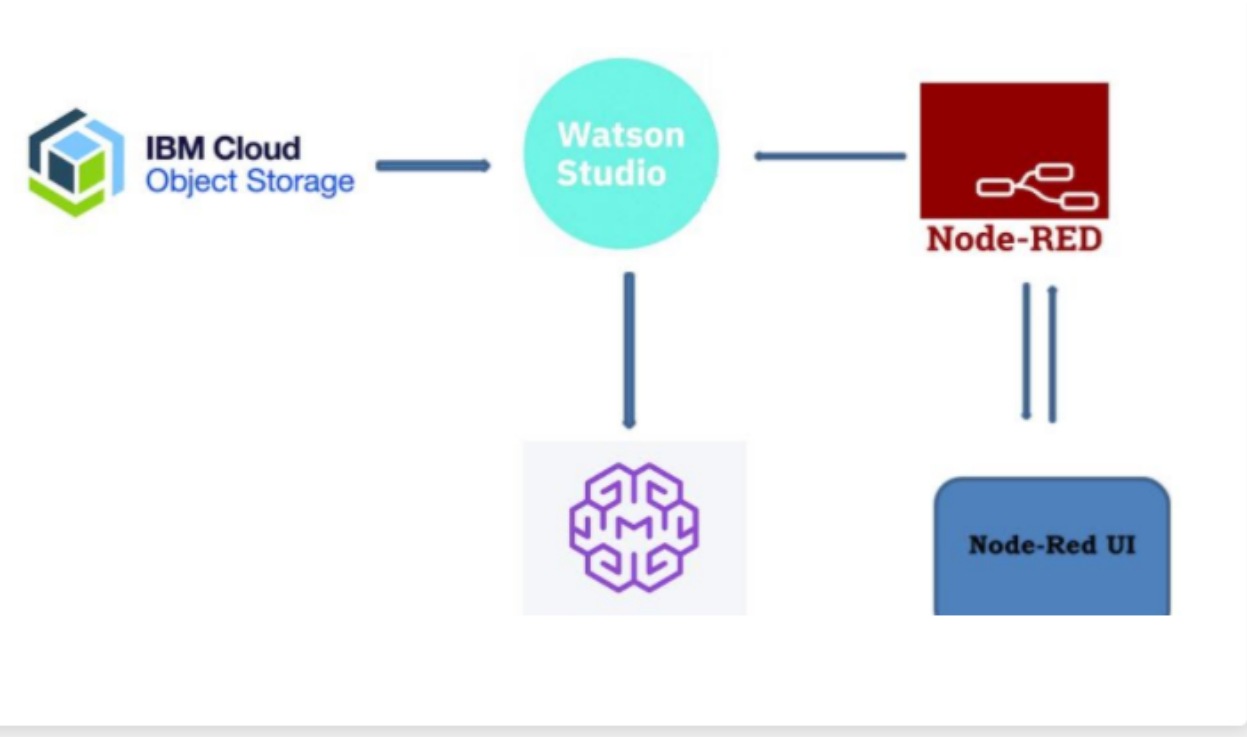
2.2 Proposed solution :

Cancer has been characterized as a heterogeneous disease consisting of many different subtypes. The early diagnosis and prognosis of a cancer type

have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients. The importance of classifying cancer patients into high or low risk groups has led many research teams, from the biomedical and the bioinformatics field, to study the application of machine learning (ML) methods. Therefore, these techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance.
Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice

3)THEORITICAL ANALYSIS :

3.1 Block diagram :



3.2 Hardware / Software designing :

in this projrect we used several software which helped us achieve our result :

IBM Cloud Object Storage :is a service offered by **IBM** for **storing** and accessing unstructured data. ... The offering can store any type of **object** which allows for uses like data archiving and backup, web and mobile applications, and as scalable, persistent **storage** for analytics.

Watson Studio : formerly Data Science Experience or DSX, is **IBM's** software platform for data science. ... In **Watson Studio**, a data scientist can create a project with a group of collaborators, all having access to various analytics models and using various languages (R/Python/Scala).

Node-Red:is a flow-based development tool for visual programming developed originally by **IBM** for wiring together hardware devices, APIs and online services as part of the Internet of Things. **Node**-**RED** provides a web browser-based flow editor, which can be used to create JavaScript functions.
Software genre: Visual programming language

Languages used: JavaScript

Machine Learning sevice : s a form of AI that enables a system to learn from data rather than through explicit programming. ... Then, when you provide the predictive model with data, you will receive a prediction based on the data that trained the model.

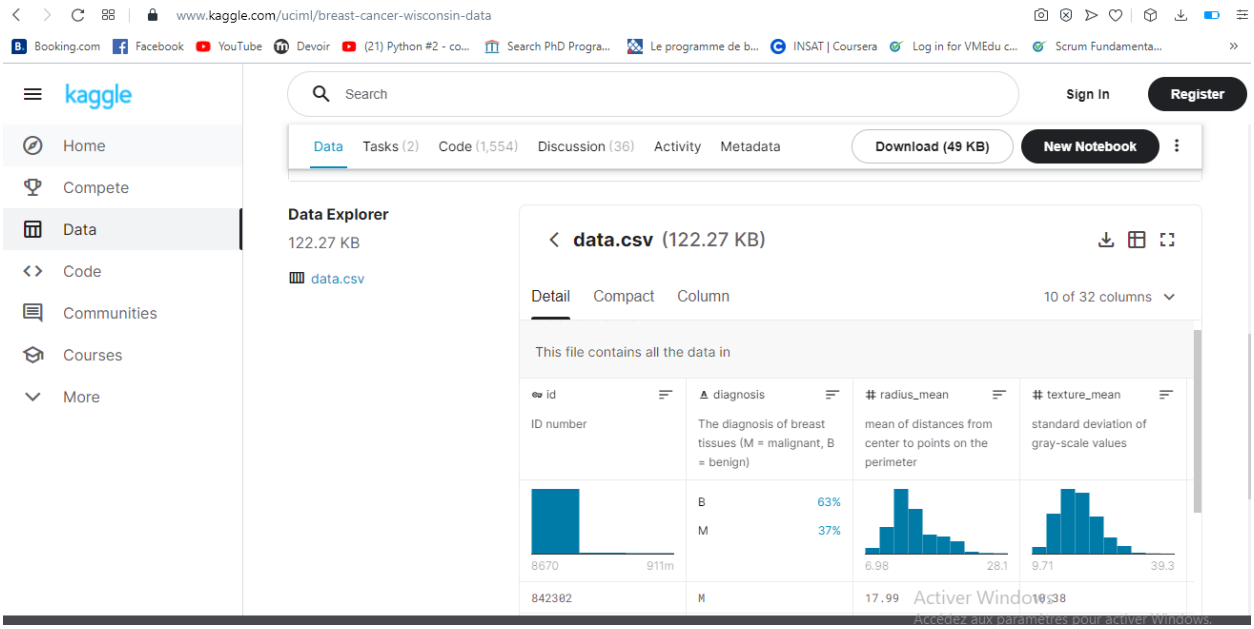
4)EXPERIMENTAL INVESTIGATIONS:

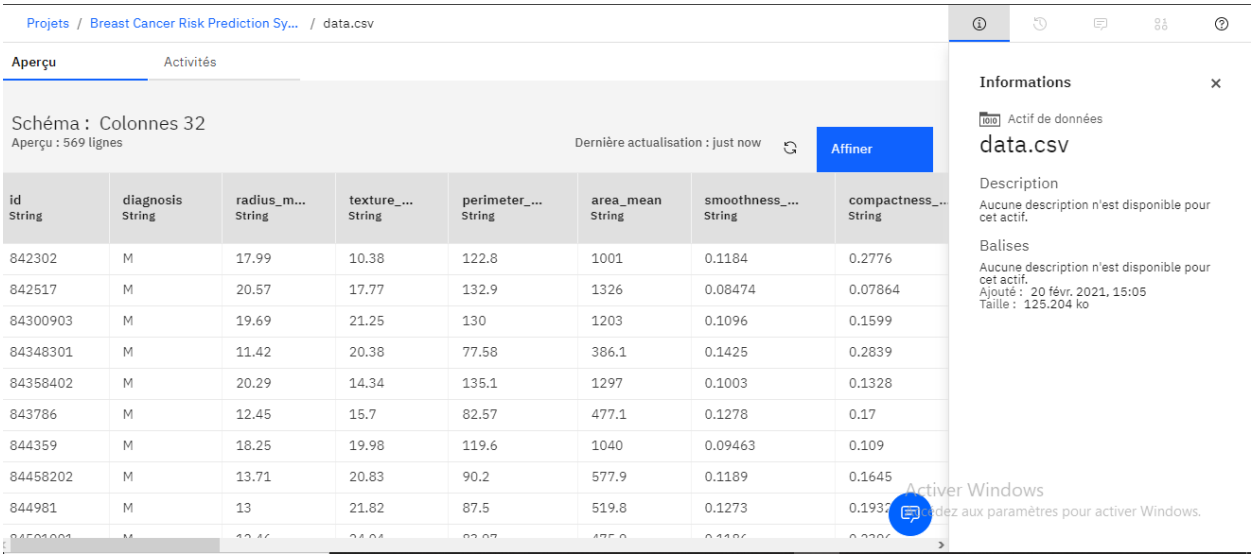
Unnecessary delays in clinical trials or poor protocol decisions can make developing new treatments more expensive. IBM technology can help you manage the complexity of oncology trials while reducing trial costs.
Our team at IBM Research – Haifa hypothesized that a model combining machine learning and deep learning could be applied to assess breast cancer at a level both comparable to radiologists and with the capabilities to be accepted into clinical practice as a second reader.
ogether with IBM Watson Machine Learning, IBM Watson Studio is a leading data science and machine learning platform built from the ground up for an AI-powered business. It helps

enterprises scale data science operations across the lifecycle—simplifying the process of experimentation to deployment, speeding up data exploration and preparation, as well as model development and training.

5)FLOWCHART:

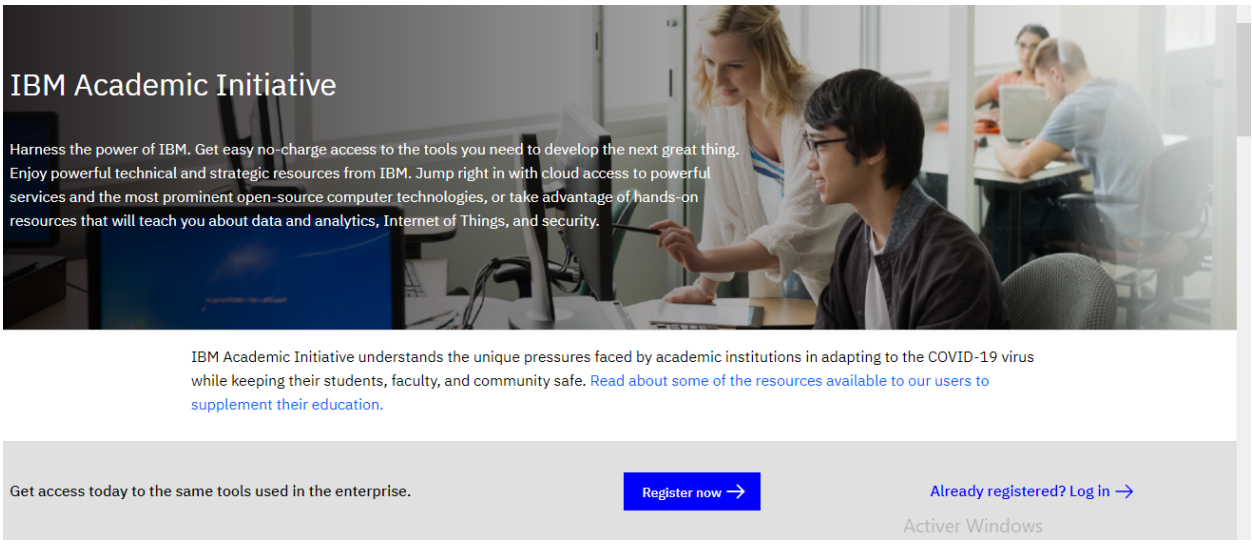
***collecting data :**



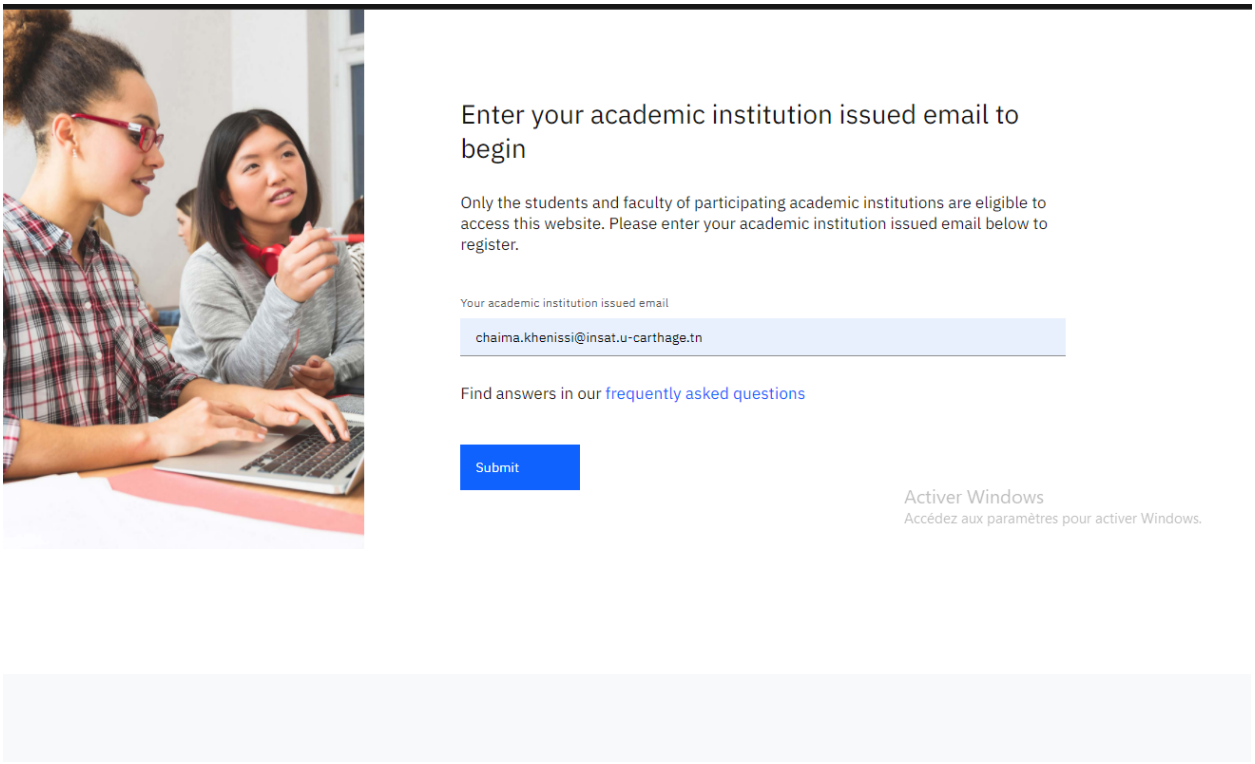


***create IBM academic initiative account :**

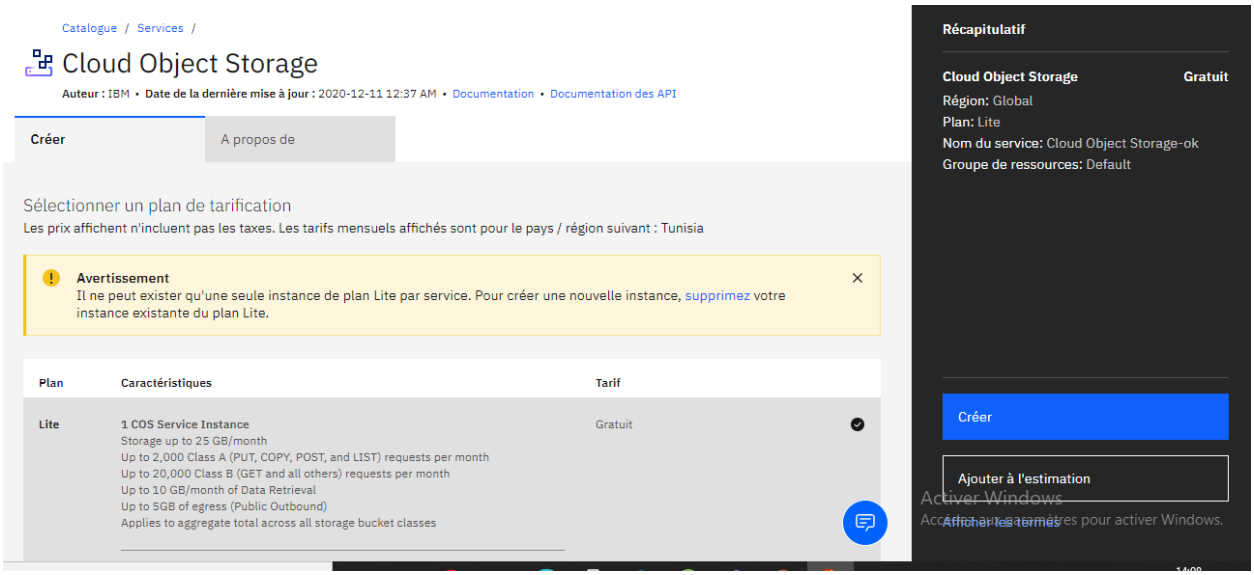
****)create IBM academic initiative account :**



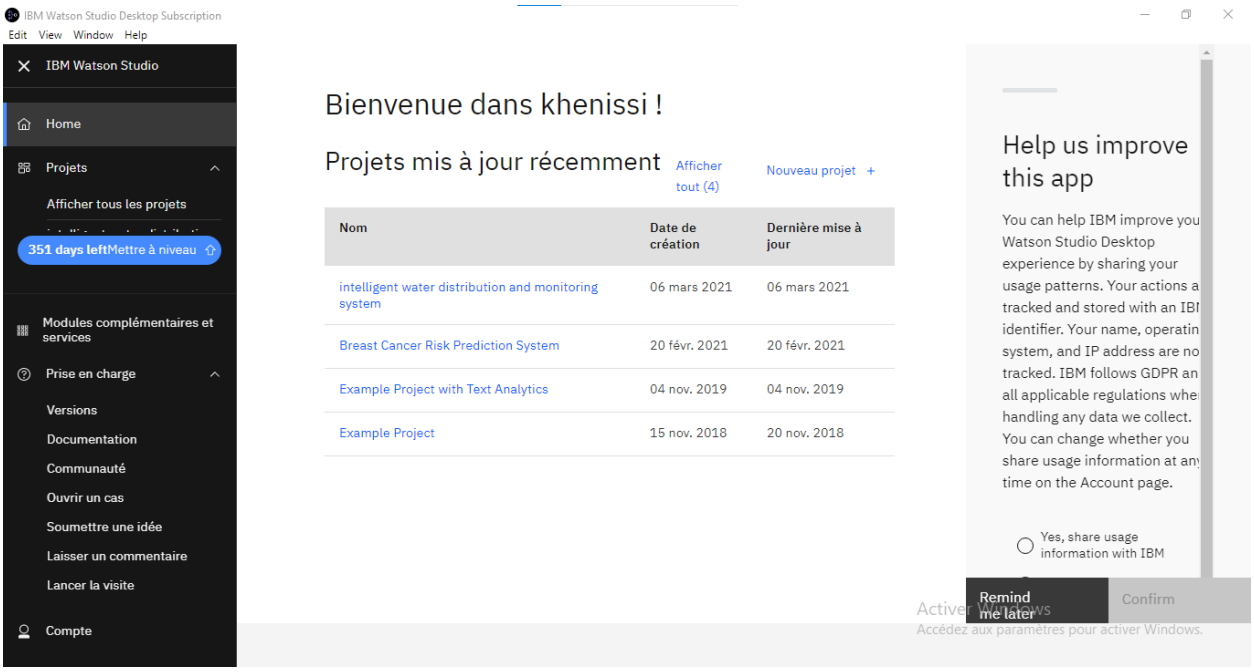
**)Login to IBM Cloud :



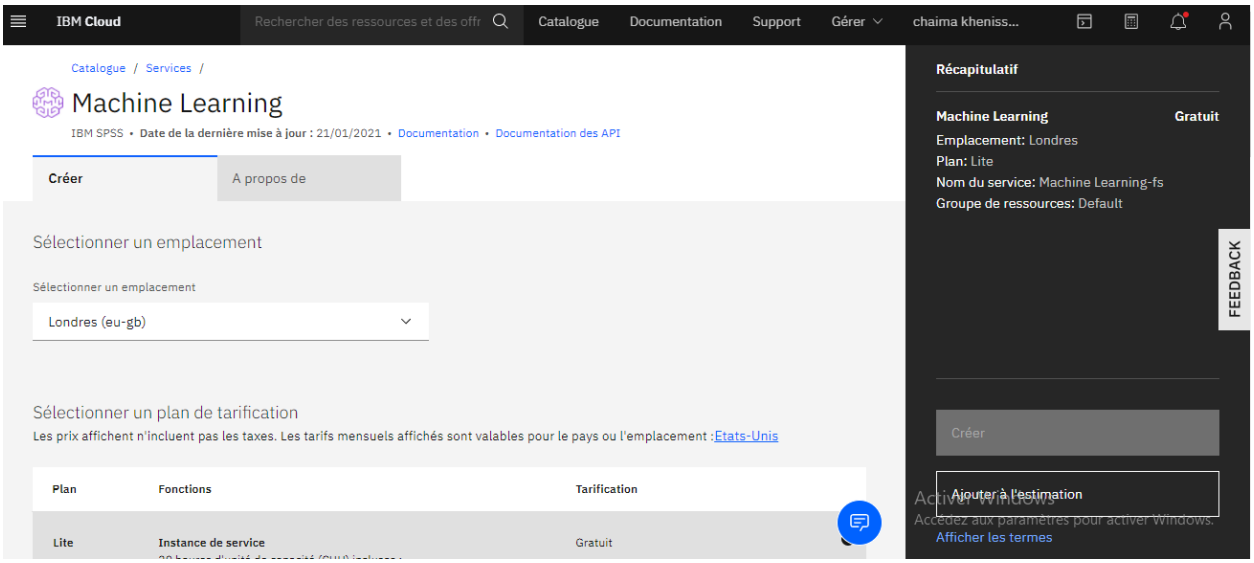
***)Create Cloud Object Storage Service:



****)Download Watson Studio Desktop and Create Watson Studio Platform

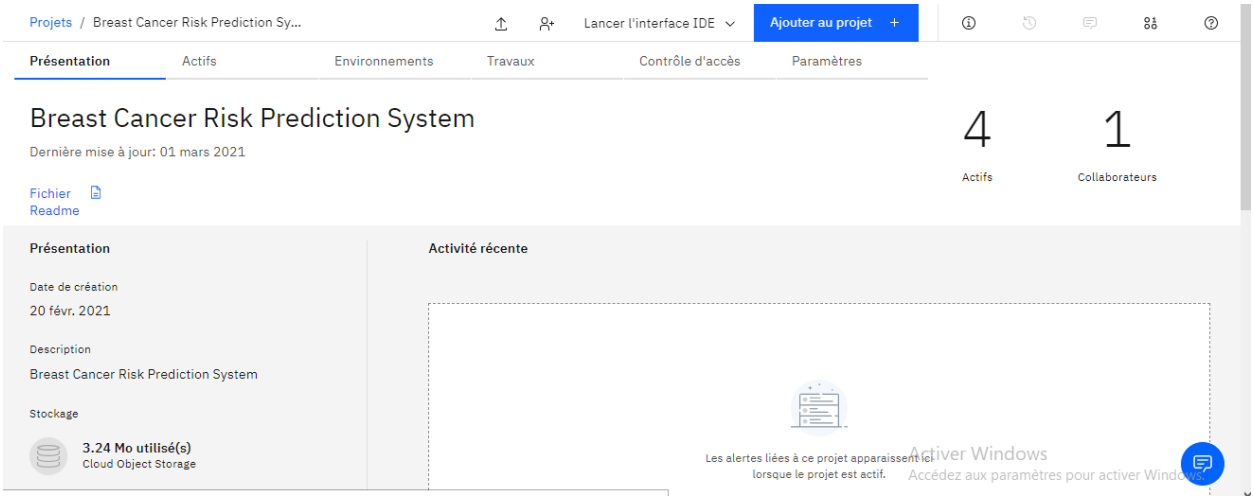


*****)Create Machine Learning Service :

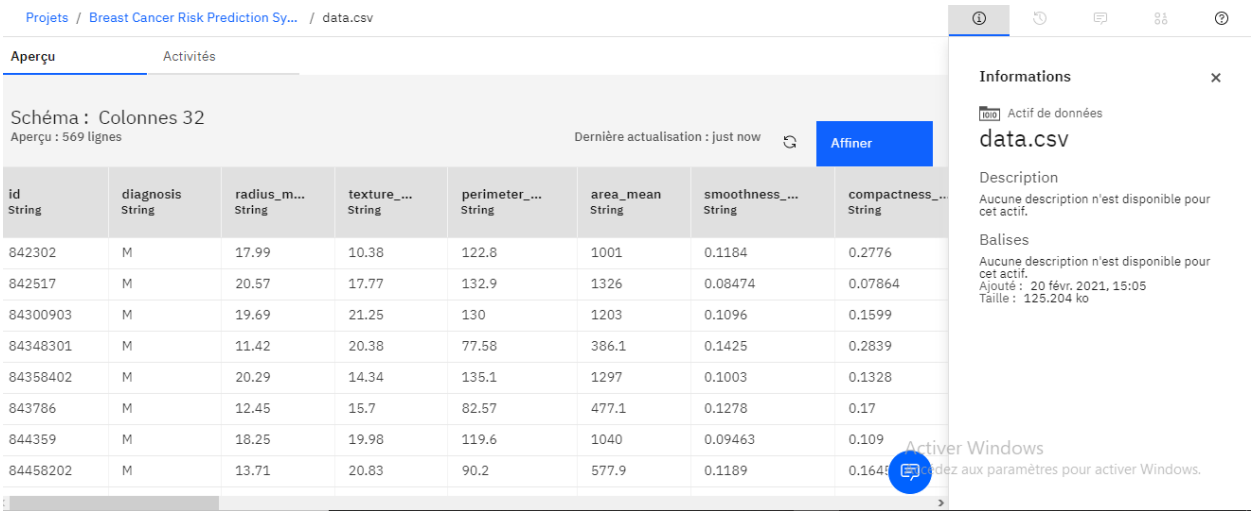


***)Train A Model In Watson Studio:

*)Create A Project In Watson Studio :

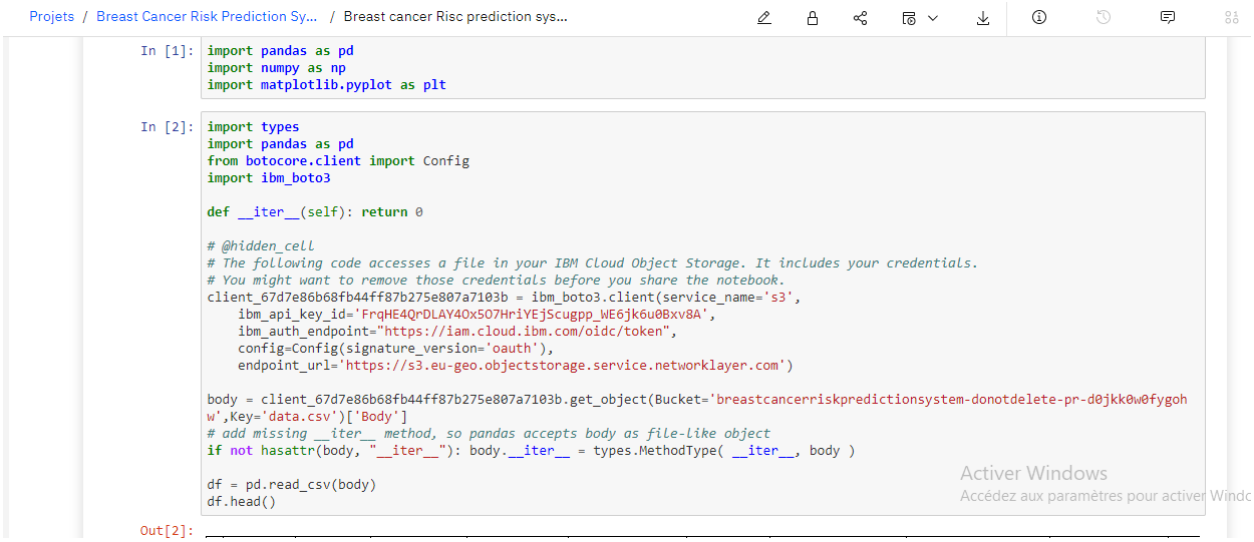


**) Upload The Dataset :



***)Create Notebook Instance :

****)Importing Libraries and Impoting Dataset From IBM COS:



Projets / Breast Cancer Risk Prediction Sy... / Breast cancer Risc prediction sys...

```
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body = client_67d7e86b68fb44ff87b275e807a7103b.get_object(Bucket='breastcancerriskpredictionsystem-donotdelete-pr-d0jkk0w0fygoh
w',Key='data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head()
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	poir
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10

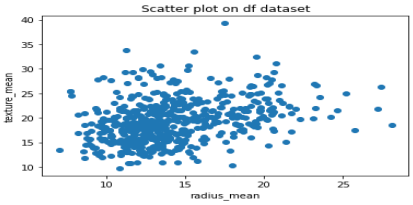
5 rows × 33 columns

In [3]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```


*****)Data

In [4]:
```
plt.scatter(df['radius_mean'],df['texture_mean'])
plt.xlabel('radius_mean')
plt.ylabel('texture_mean')
plt.title('Scatter plot on df dataset')
```

Out[4]: `Text(0.5, 1.0, 'Scatter plot on df dataset')`



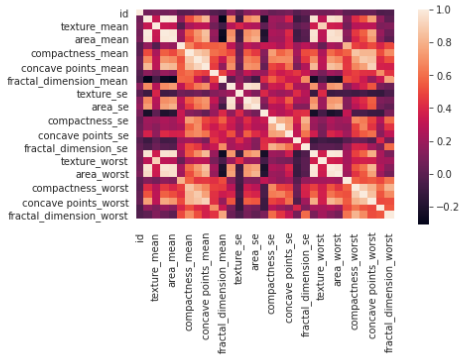
In [5]:
```
sns.set_style("whitegrid")
sns.FacetGrid(df, hue="id", size=4) \
   .map(plt.scatter, "perimeter_mean", "smoothness_mean") \
   .add_legend()
```


Visualization :

Projets / Breast Cancer Risk Prediction Sy... / Breast cancer Risc prediction sys...

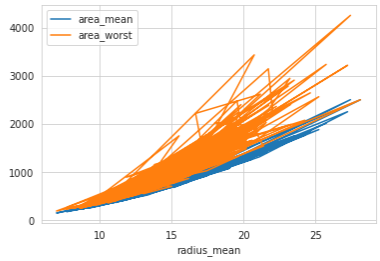
In [6]:
```
#HEATMAP FROM SEABORNI
sns.heatmap(df.corr())
plt.show
```

Out[6]: `<function matplotlib.pyplot.show(*args, **kw)>`



In [7]:
```
df.plot(y=['area_mean','area_worst'],x='radius_mean')
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f53c083a8d0>
```

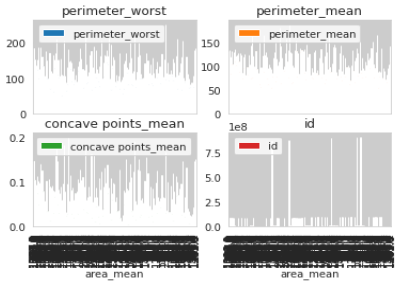


In [8]:
```
df.plot.bar(y=['perimeter_worst','perimeter_mean','concave points_mean','id'],x='area_mean',subplots=True,layout=(2,2))
```

Out[8]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc3642d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc383710>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc352990>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc303d90>]],
```


In [8]:
```
df.plot.bar(y=['perimeter_worst','perimeter_mean','concave points_mean','id'],x='area_mean',subplots=True,layout=(2,2))
```

Out[8]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc3642d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc383710>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc352990>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f53bc303d90>]],
      dtype=object)
```



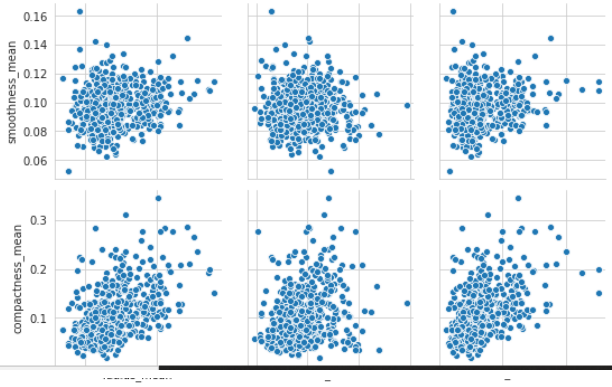
Activer Windows
Accédez aux paramètres pour activer Window

In [9]:
```
sns.pairplot(
```

area_mean area_mean

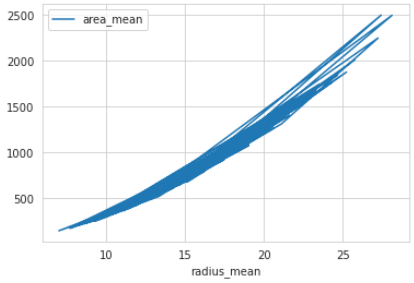
```
In [9]: sns.pairplot(
            df,
            x_vars=["radius_mean", "texture_mean", "area_mean"],
            y_vars=["smoothness_mean", "compactness_mean"],
        )
```

```
Out[9]: <seaborn.axisgrid.PairGrid at 0x7f53c07b76d0>
```




```
In [10]: df.plot(x='radius_mean',y='area_mean')
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f53b4245a10>
```




```
In [11]: sns.boxplot(x='radius_mean',y='area_mean',data=df)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f53b41a9890>
```

******)Taking Care Of Missing Data:

```
fractal_dimension_mean     False
radius_se                  False
texture_se                 False
perimeter_se               False
area_se                    False
smoothness_se              False
compactness_se             False
concavity_se               False
concave points_se          False
symmetry_se                False
fractal_dimension_se       False
radius_worst               False
texture_worst              False
perimeter_worst            False
area_worst                 False
smoothness_worst           False
compactness_worst          False
concavity_worst            False
concave points_worst       False
symmetry_worst             False
fractal_dimension_worst    False
Unnamed: 32                 True
dtype: bool
```

```
In [13]: df.fillna(df.mean(),inplace=True)
```

```
In [14]: df['Unnamed: 32'].fillna(df['Unnamed: 32'].mean(),inplace=True)
```

*******)One Hot Encoding and Feature Scaling and Splitting Data Into Train And Test and :

```
In [13]: df.fillna(df.mean(),inplace=True)

In [14]: df['Unnamed: 32'].fillna(df['Unnamed: 32'].mean(),inplace=True)

In [15]: from sklearn.preprocessing import LabelEncoder
         labelencoder_y=LabelEncoder()
         df['diagnosis']=labelencoder_y.fit_transform(df['diagnosis'])

In [16]: x = df.iloc[:, 0:3].values
         y = df.iloc[:, 3].values

In [17]: from sklearn.preprocessing import OneHotEncoder
         onehotencoder = OneHotEncoder()

In [18]: x= onehotencoder.fit_transform(x).toarray()
         x= x[:, 1:]

In [19]: 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 [20]: from sklearn.preprocessing import StandardScaler
         sc1 = StandardScaler()
         x_train = sc1.fit_transform(x_train)
         x_test = sc1.transform(x_test)
```


****)Deploying The Model :

*)Deploy The Model In Watson Machine Learning:

```
In [21]: pip install watson-machine-learning-client
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.389-py3-none-any.whl (538 kB)
     |████████████████████████████████| 538 kB 11.9 MB/s eta 0:00:01
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-learni
ng-client) (1.13.11)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-lear
ning-client) (1.25.9)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-learn
ing-client) (0.3.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-lea
rning-client) (2.24.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-lear
ning-client) (2020.12.5)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-learnin
g-client) (4.47.0)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-
learning-client) (2.7.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-learn
ing-client) (1.0.5)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from watson-machine-lea
rning-client) (0.8.3)
Requirement already satisfied: botocore<1.17.0,>=1.16.11 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from b
oto3->watson-machine-learning-client) (1.16.11)
Requirement already satisfied: s3transfer<0.4.0,>=0.3.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from bo
to3->watson-machine-learning-client) (0.3.3)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from boto
3->watson-machine-learning-client) (0.9.4)

In [ ]: from watson_machine_learning_client import WatsonMachineLearningAPIClient

In [23]: online_deployment = client.deployments.create(model_uid, 'Deployment X', 'Online deployment of XYZ model.')
         virtual_deployment = client.deployments.create(model_uid, 'Deployment A', 'Virtual deployment of XYZ model.', deployment_type='v
         irtual')
```

```
In [24]: !pip install -U ibm-watson-machine-learning

Requirement already up-to-date: ibm-watson-machine-learning in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (1.
0.53)
Requirement already satisfied, skipping upgrade: packaging in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from
ibm-watson-machine-learning) (20.4)
Requirement already satisfied, skipping upgrade: ibm-cos-sdk==2.7.* in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packa
ges (from ibm-watson-machine-learning) (2.7.0)
Requirement already satisfied, skipping upgrade: urllib3 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from i
bm-watson-machine-learning) (1.25.9)
Requirement already satisfied, skipping upgrade: requests in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from
ibm-watson-machine-learning) (2.24.0)
Requirement already satisfied, skipping upgrade: tabulate in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from
ibm-watson-machine-learning) (0.8.3)
Requirement already satisfied, skipping upgrade: lomond in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from ib
m-watson-machine-learning) (0.3.3)
Requirement already satisfied, skipping upgrade: pandas<=1.0.5 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages
```

```
In [25]: from ibm_watson_machine_learning import APIClient
         import json
         import numpy as np
```

```
In [26]: wml_credentials ={
             "apikey":"h0vmr-9dXRvfb1wTRLxQdfcDMQQSv5uRmTCkUm1U7AIw",
             "url":"https://eu-gb.ml.cloud.ibm.com"

         }
```

```
In [27]: wml_client = APIClient(wml_credentials)
         wml_client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  --------------------  ------------------------
ID                                    NAME                  CREATED
94905796-16f6-4b8b-842c-52bc6eadbf20  DEPLOYMENT            2021-03-06T17:30:10.342Z
25d58cc5-07c6-48c6-8325-15a26d691023  mon_espace-deployment 2021-03-05T23:07:15.485Z
------------------------------------  --------------------  ------------------------
```

```
In [28]: SPACE_ID="94905796-16f6-4b8b-842c-52bc6eadbf20"
```

```
In [29]: wml_client.set.default_space(SPACE_ID)
Out[29]: 'SUCCESS'
```



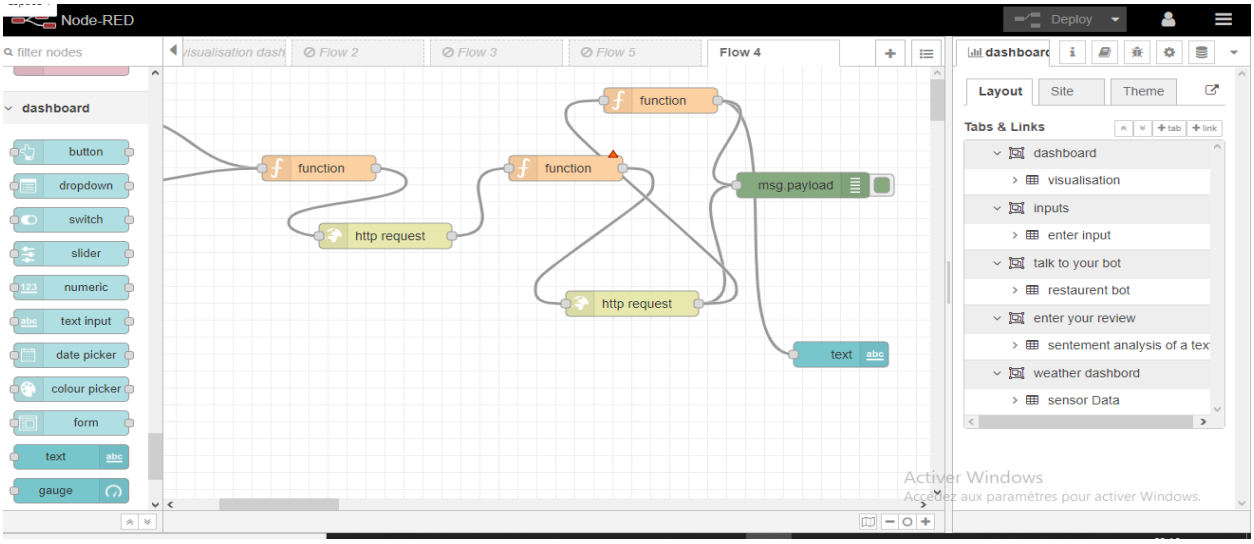
```
In [28]: SPACE_ID="94905796-16f6-4b8b-842c-52bc6eadbf20"

In [29]: wml_client.set.default_space(SPACE_ID)
Out[29]: 'SUCCESS'
```


```
In [30]: MODEL_NAME = 'sklearn Forecast'
         DEPLOYMENT_NAME = 'sklearn Deployment'
         BEST_MODEL = best_model
```

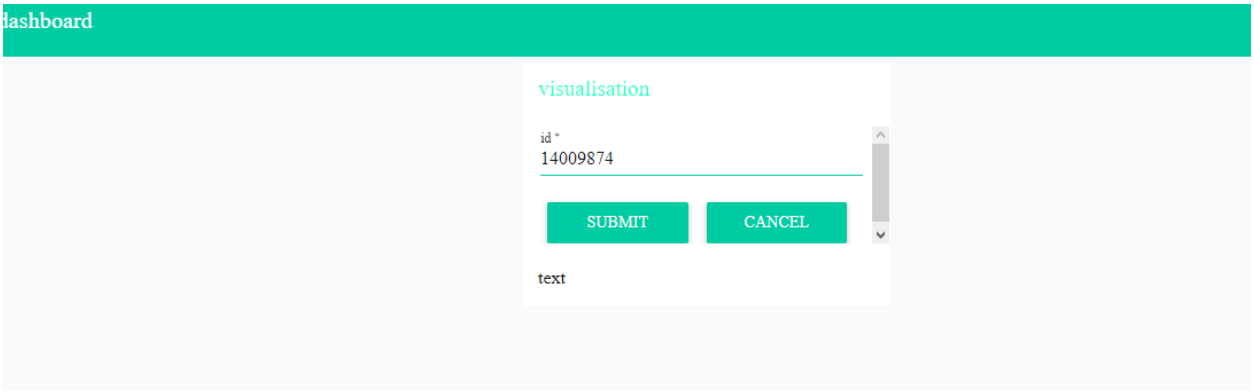
```
In [46]: from sklearn.matrics import r2_score,mean_absolute_error
```

*****)Node-RED UI :

*) Create Node-Red Application and uild UI With Node-RED



6)RESULT:



7.ADVANTAGES & DISADVANTAGES :

.)advantages : -)easy model building with less formal statical knowledge required .

-) capable of capturing interactions between predictors

-)capable of capturing nonlinearities between predictors and outcomes .

-)the existence of this application which is accessible to everyone and free, encourages everyone to monitor their health from their places without needing a doctor.

-)early recognition reduces the risk of this disease.

.) Disadvantages : -) the uncertainty of this application .

-)This algorithm needs large amount of data to attain good outcomes.

It is lazy as they store entire the training examples

8) Application :

Machine learning is not new to cancer research. Artificial neural networks (ANNs) and

decision trees (DTs) have been used in cancer detection and diagnosis for nearly 20 years (Simes 1985; Maclin et al. 1991; Ciccheti 1992). Today machine learning methods are being used in a wide range of applications ranging from detecting and classifying tumors via X-ray and CRT images .The applications of AI in medicine are developing quickly. In 2016, AI projects coupled with medicine drew in more speculation from the global economy than other projects [4]. In medicine, AI refers to the utilization of automated diagnosis processes and the treatment of patients who require care. Increased AI utilization in prescription will allow a considerable amount of the role to be automated, opening up medicinal experts' time to be used in performing different obligations, ones that cannot be automated. As such, this technology promises progressively significant utilization in the field of human resources (HR).

9) CONCLUSION :

to conclude the application of IBM Cloud (machine learning ,watson studio ..) helped us to realize this application which allows to predict cancer and consequently to reduce the risk and to help people to make the diagnosis at home and for free. we need such applications in our modern world to make life easier

Today, **machine learning** is helping to streamline administrative processes in hospitals, map and treat infectious diseases and personalize **medical** treatments. ... "It can also be **used** to demonstrate and educate patients on potential disease pathways and outcomes given different treatment options.

10)FUTURE SCOPE :

We have come across many studies reporting the potential of Machine Learning (ML) for big data analysis, especially in the medical field . In healthcare, ML applications may offer better indications of the risks and implications of the correlation between diagnosis and therapies; data that may later be confirmed by randomized controlled trials in a sample of patients . ML has been widely applied in the diagnosis and prognosis of certain diseases, mainly cancer patients

11) BIBILOGRAPHY :

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data
https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf
https://wml-api-pyclient.mybluemix.net/

https://youtu.be/apFbFikesjA

APPENDIX

https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/d4b3fddb-e988-4d90-bd4a-58c2abd0182e/view?access_token=e4436e73a56ae3d31cdcdce9eeb15c3d6b71743d3eb7b6299bad2a6b8798804a