Diabetics Prediction System Based On Life Style

1 INTRODUCTION

1.1 OVERVIEW

Diabetes is a dangerous and chronic diseases that cause blood sugar to rise. When diabetes remains untreated and unexplained, several complications arise.Machine learning extends to many areas of public health through the rapid progression of machine learning. In recent years, diabetes has become a chronic condition that can cause many complications. There are some signs of diabetes, such as the patient's number of births, their BMI, insulin level , age, etc. So we suggested an Machine Learning based model that would help solve a serious problem which predict whether or not a patient has diabetes, based on certain diagnostic measure included in the dataset. For practical health management of diabetes, this model will be useful

1.2 PURPOSE

The tedious method of recognition results in a patient attending a diagnostic center and consulting a physician. However, the rise of approaches to machine learning resolves this crucial point. The he objective of this study is to design a model that can predict the risk of diabetes in patients with maximum accuracy.The purpose of this project is to develop an end-to-end web application that predicts the probability of females having diabetes. The node-red is used for building application with the machine learning model trained and deployment is done on IBM Watson Studio.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

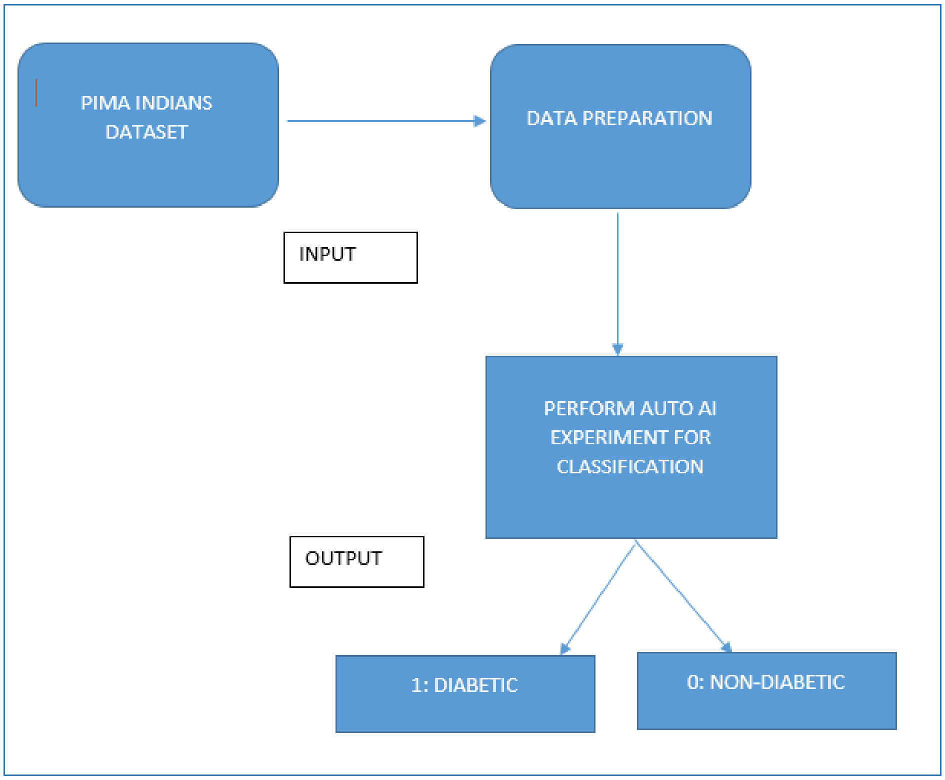
In the healthcare sector, the diabetes predictor machine is a beneficial system. In comparison, they have to spend their resources in vain any time they try to get a diagnostic study. Numerous algorithms, including the conventional approach of machine learning is used for forecasting diabetes.Diabetes mellitus, which can cause multiple complications, is a disorder. It is worth researching how to accurately predict and diagnose this illness by using machine learning

2.2 PROPOSED SOLUTION

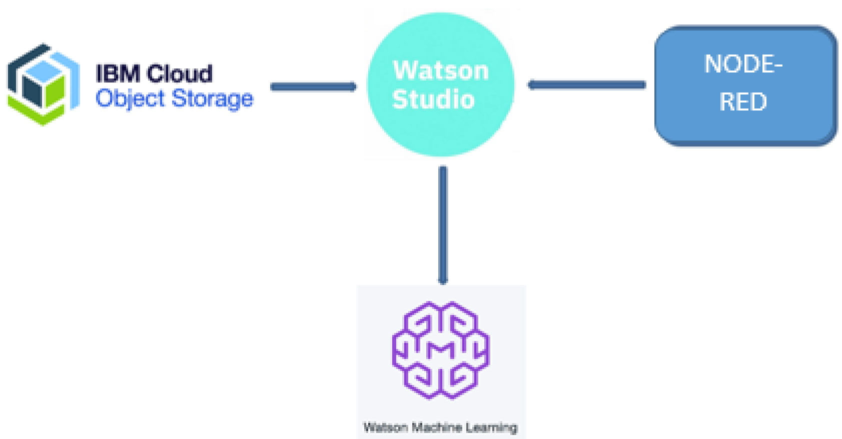
With the rise of approaches in machine learning and data mining, we have been able to find a solution to this problem. One of the main fields of Machine Learning is data mining. The purpose of this study is to build a device that can determine whether the patient is prone to diabetes or not. In the healthcare industry, the diabetics prediction is also a beneficial method for forecasting the disease. In contrast, if they attempt to receive a medical sample, they have to expend their money in vain. In addition , in order to increase precision, more and more researchers have used ensemble approaches.

3.THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.1 ARCHITECTURE



4.EXPERIMENTAL DESIGN

This section discusses the system's general design and the procedure it pursued in order to achieve the prediction.

Dataset collection:

This dataset describes the medical records for Pima Indians and whether or not each patient will have an onset of diabetes.

Fields description follow:

pregnancies = Number of times pregnant
Glucose = Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure = Diastolic blood pressure (mm Hg)
SkinThickness = Triceps skin fold thickness (mm)
Insulin = 2-Hour serum insulin (mu U/ml)
BMI = Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction = Diabetes pedigree function (a function which scores likelihood of diabetes based on family history
Age = Age (years)
Outcome = Class variable (0 if non-diabetic, 1 if diabetic)

Steps in building Diabetics Prediction System based on Life Style:

- Open Watson studio
- Create a Project
- Add Auto AI EXPERIMENT
- Create a machine learning Instance
- Associate ML to the project
- Load the dataset to cloud object storage
- Select the prediction parameter in the dataset
- Train the model (Training-90%, hold out/validation -10%)
    - Step1: Create pipeline for algorithms giving highest accuracy.
    - Step2: Add theses pipeline to a dictionary
    - Step3: Fit the pipelines in training dataset.
    - Step4: Compare accuracy of all pipelines added.
    - Step5: Prediction and identification of the most accurate model based on test data
    - Step 6: Evaluate the prediction results using various evaluation metrics

like classification accuracy, confusion matrix and f1-score.
    - Classification Accuracy- It is the ratio of number of correct predictions to the total number of input samples.
    - Confusion Matrix- It gives us gives us a matrix as output and describes the complete performance of the model.
- Once trained, Deploy the model
- Once deployed, we will get an API
- Build the web application using node_red

Key Findings and Insights:

This is a binary classification experiment with supervised learning. This analysis concentrates an effective binary classifier for identification of diabetes.
Using Gradient Boosting Classifier and XGB classifier. Based on those medical measurements used in the dataset, we need to predict whether a patient has diabetes. Several limits have been imposed on the choosing of these instances from a broader database.
In fact, all the patients here are women of Pima Indian heritage who are at least 21 years old. This model aims in creating a diabetes prediction application. For developing a project. Auto-AI-Experiment is introduced by developing a ML instance and associate the same to the project. Dataset is loaded into the cloud object storage and selected the prediction parameters.

Input:

Input involves pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, Body Mass Index(BMI), DiabetesPedigreeFunction, Age, Outcome

Output:

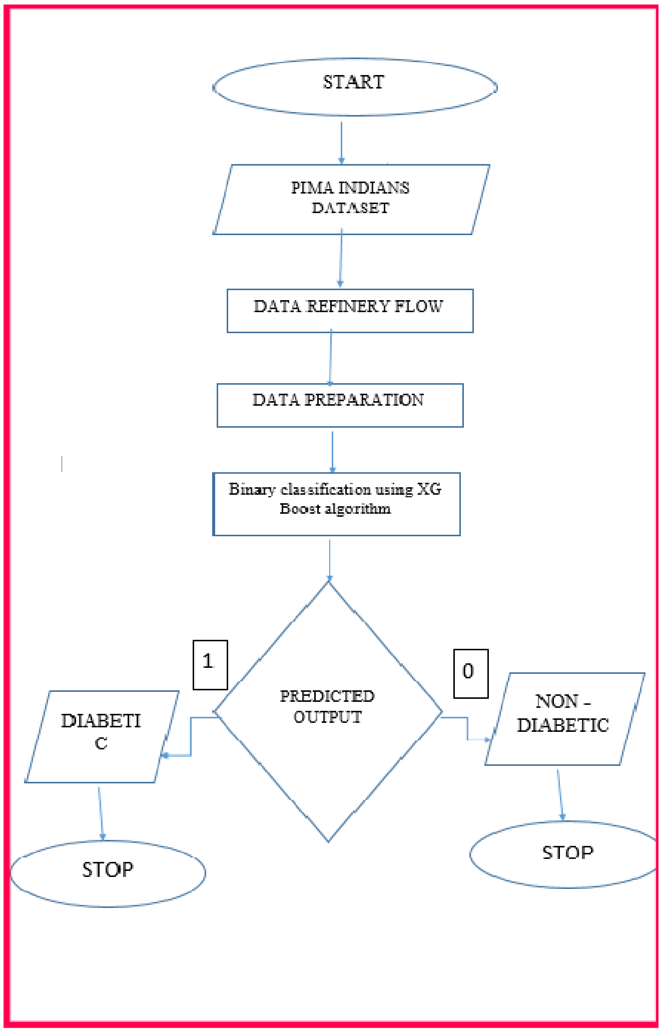
The outcome parameter is taken as output.

Training a model:

(splitting and normalization) Dataset Preparation
While using ML algorithms we need to split our data into a training set and testing set. Here we have used for Training-90% of our dataset and Testing -10% validation or test flow

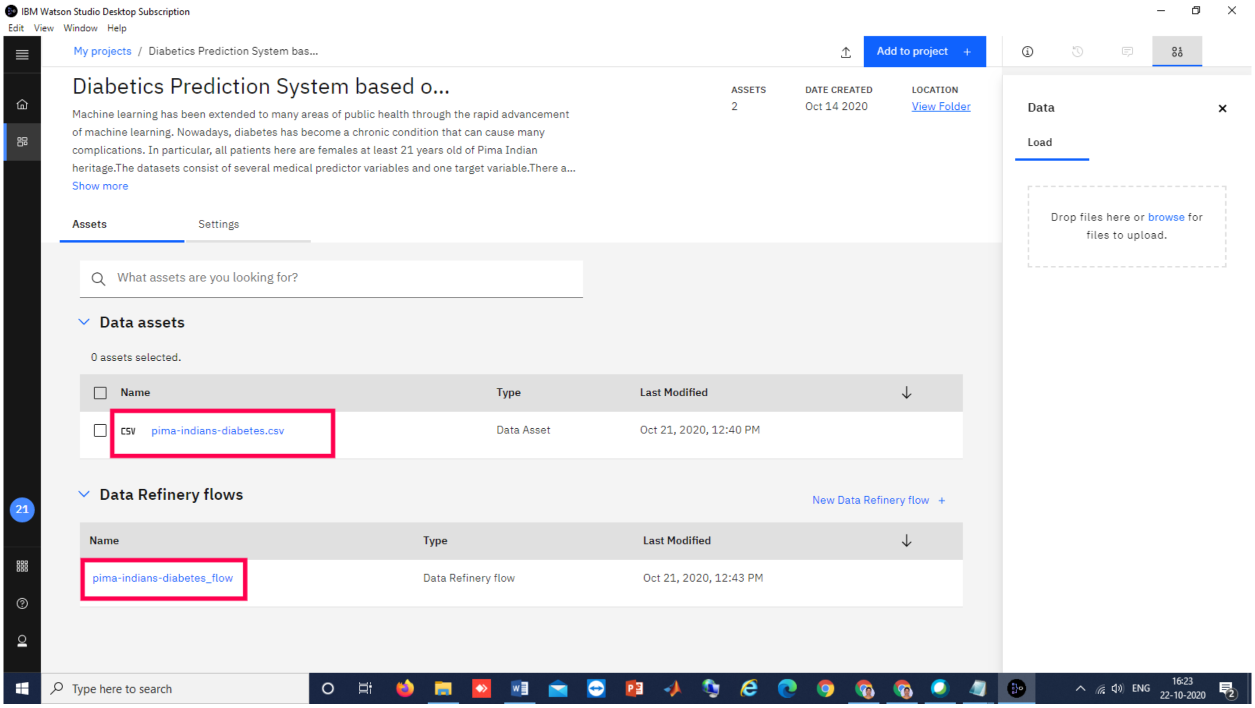
For deploying the model and build web application using NODE-RED.After tuning the hyper parameters, AutoAI experiment have given that XG boost classifier is performing the best by giving 0.77 accuracy.

5. FLOWCHART

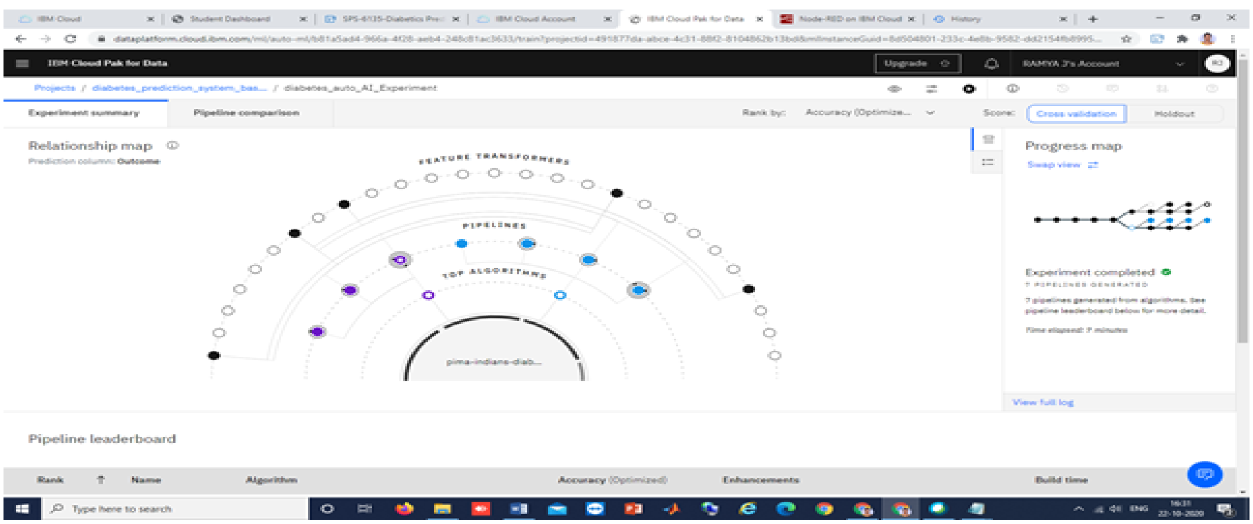


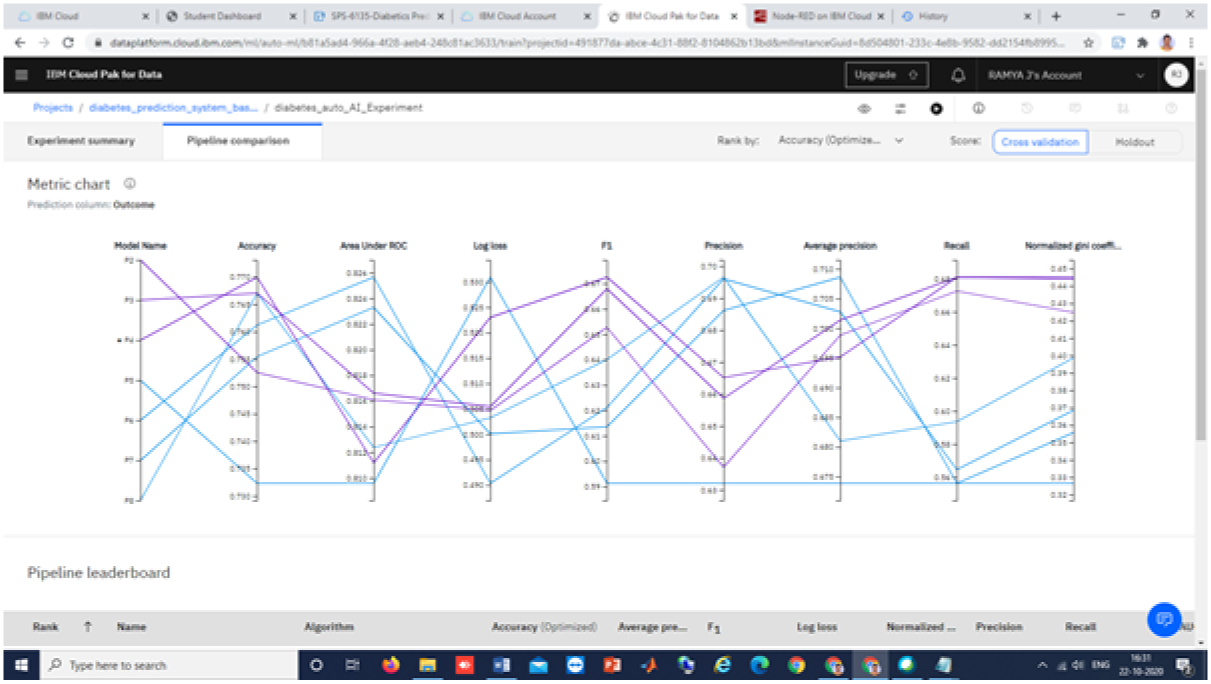
6.RESULT:

6.1 Dataset and data refinery flow

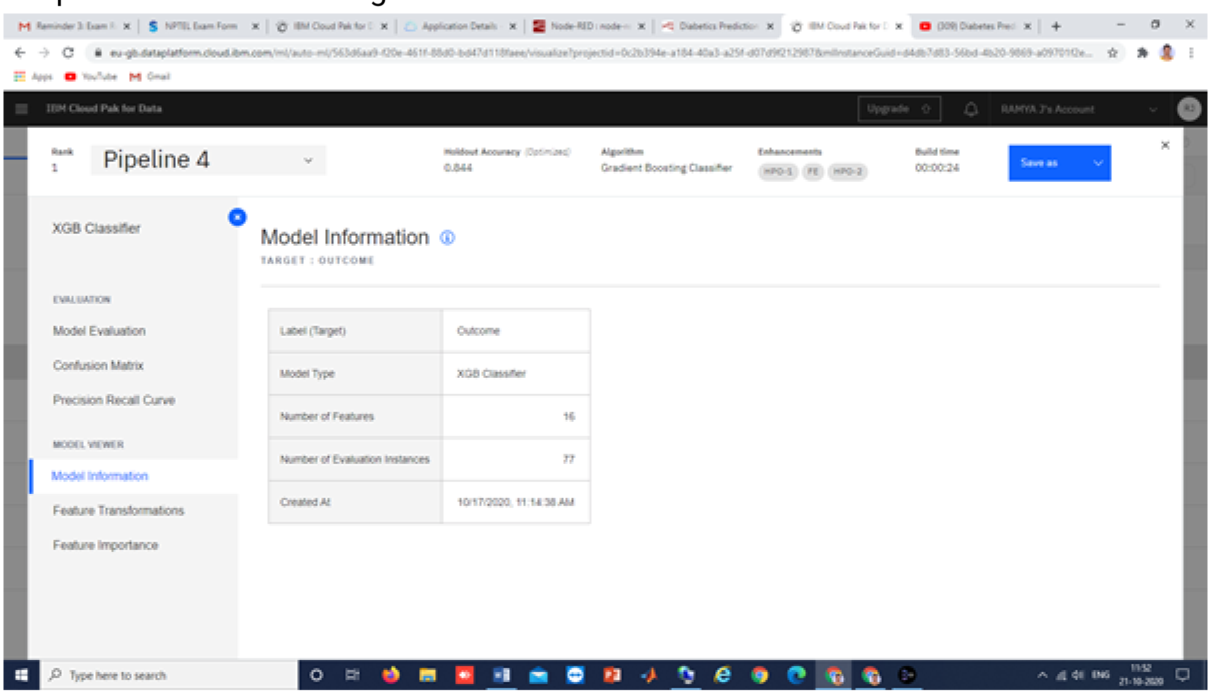


6.2 Auto_AI_OUTPUT

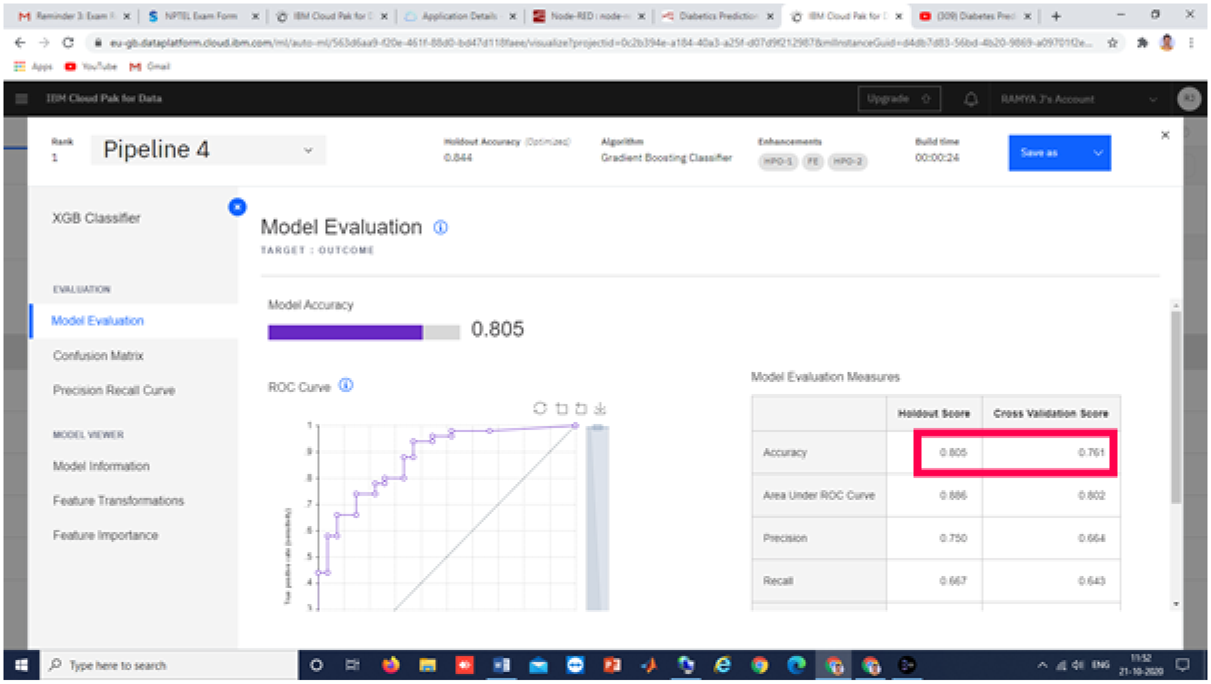




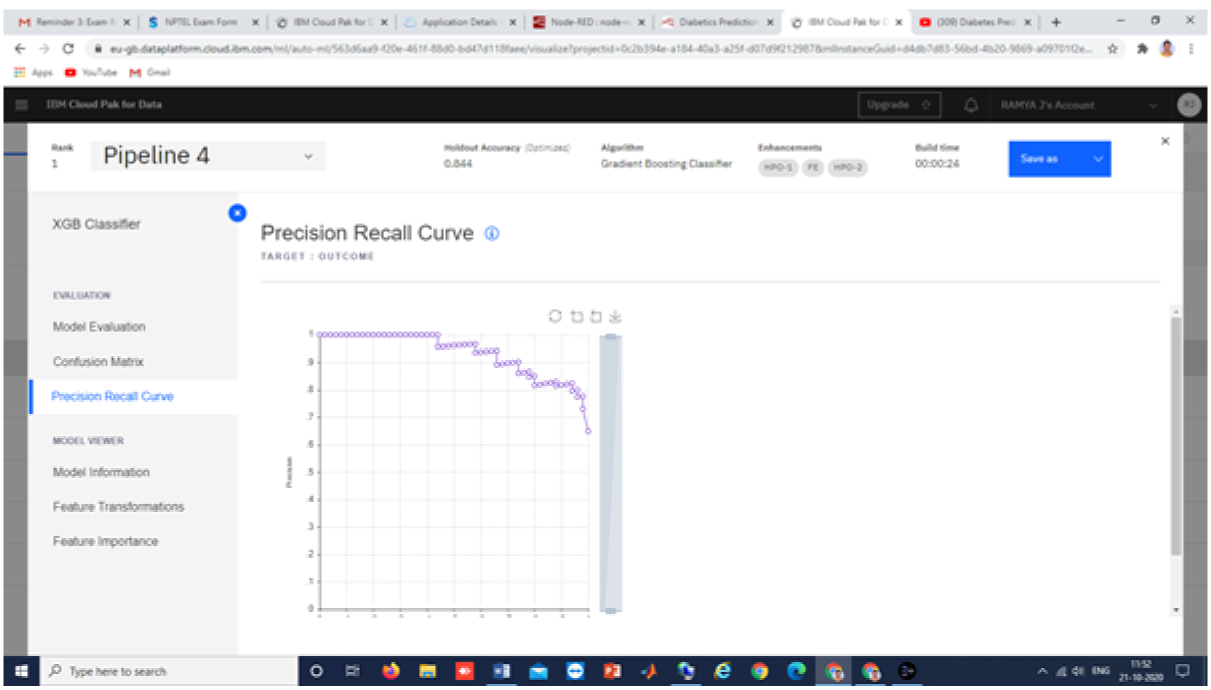
6.3 prediction results using various evaluation metrics



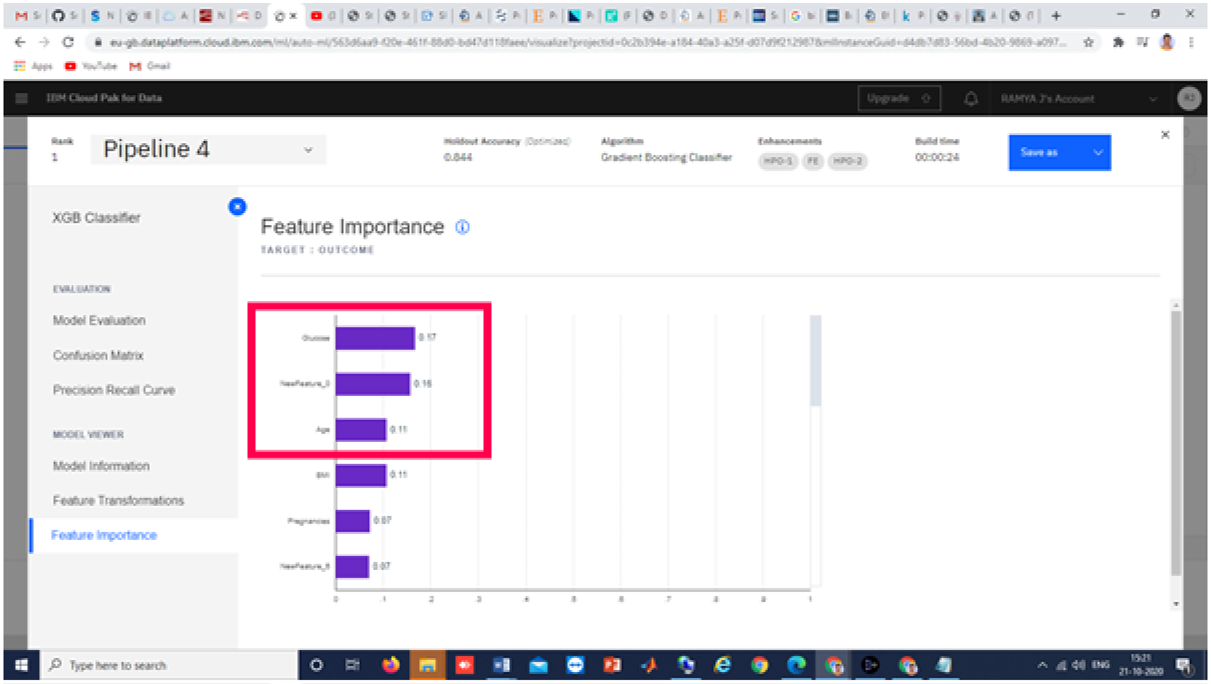
6.4 ROC Curve

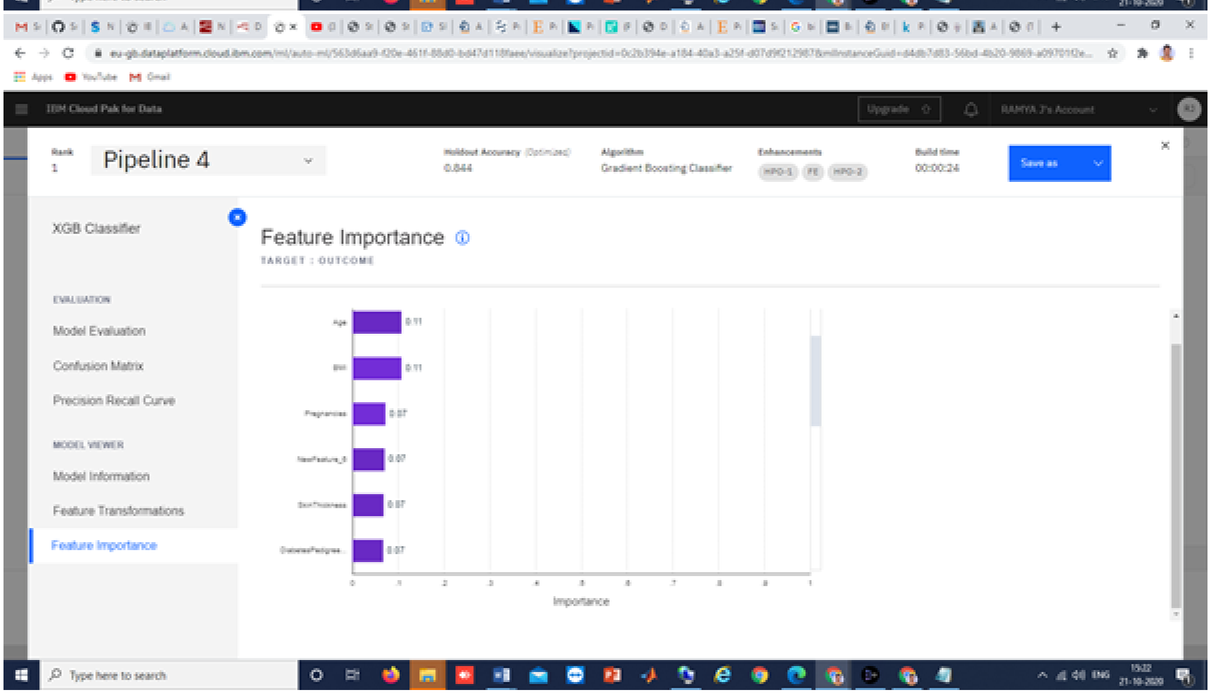


6.5 Precision recall Curve

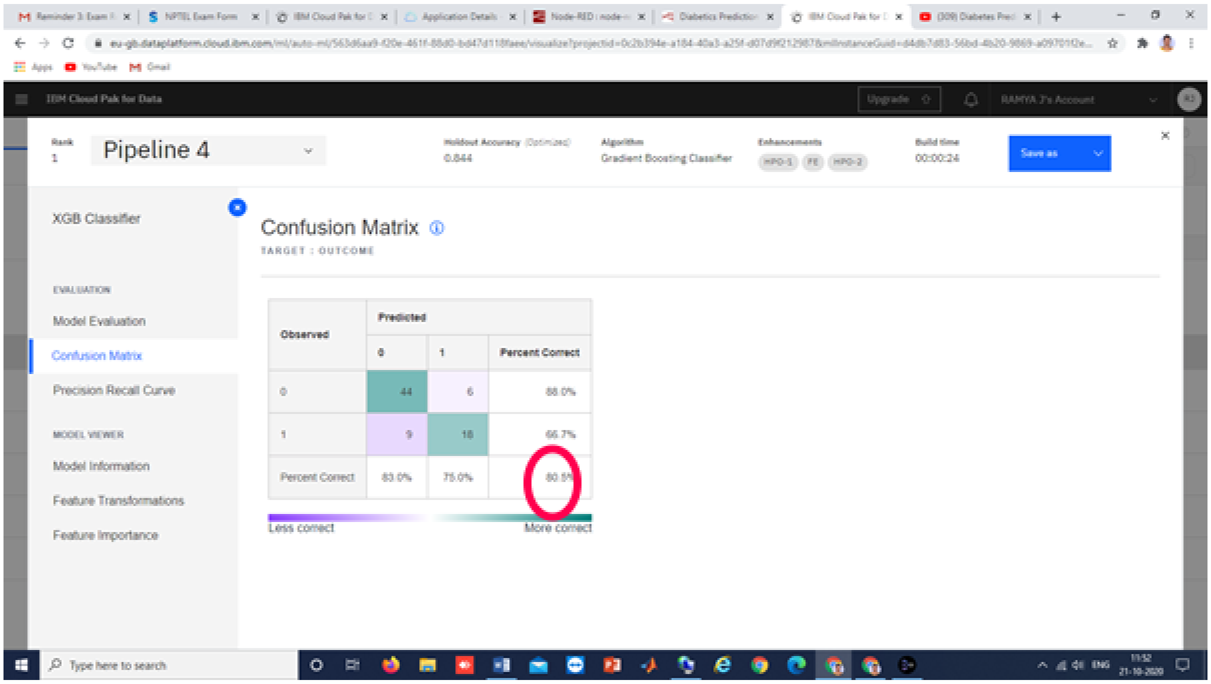


6.6 Feature importance

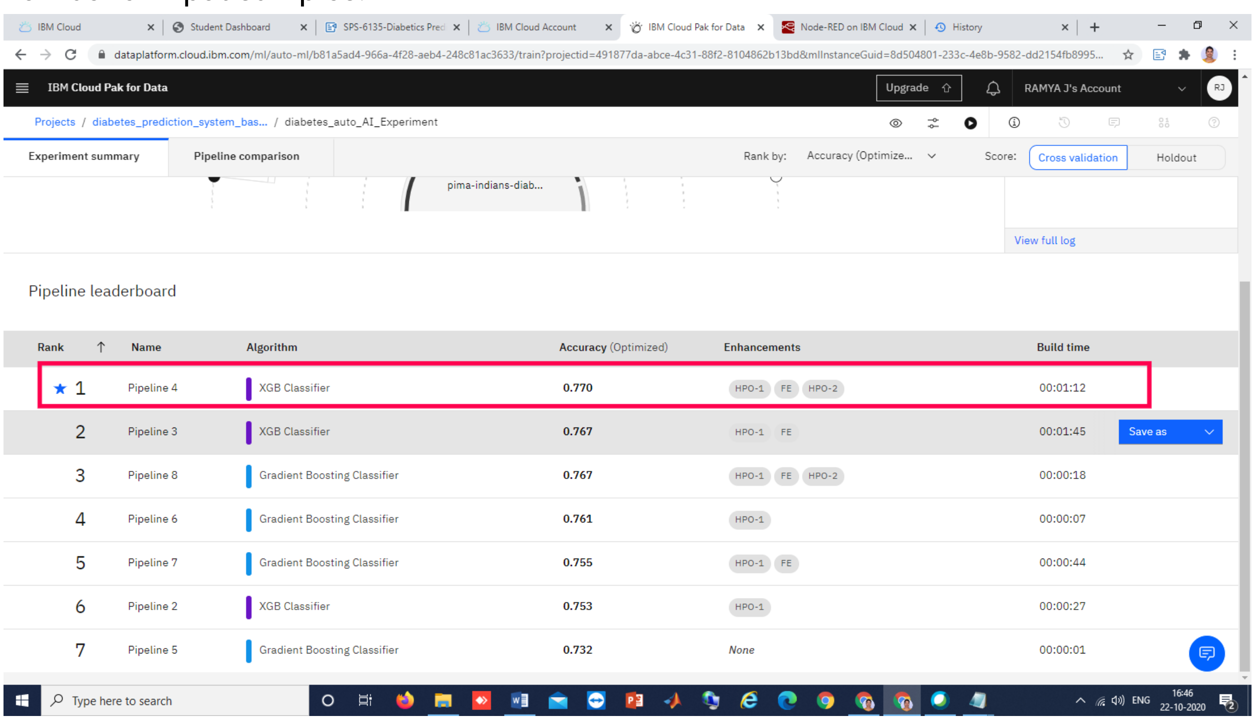




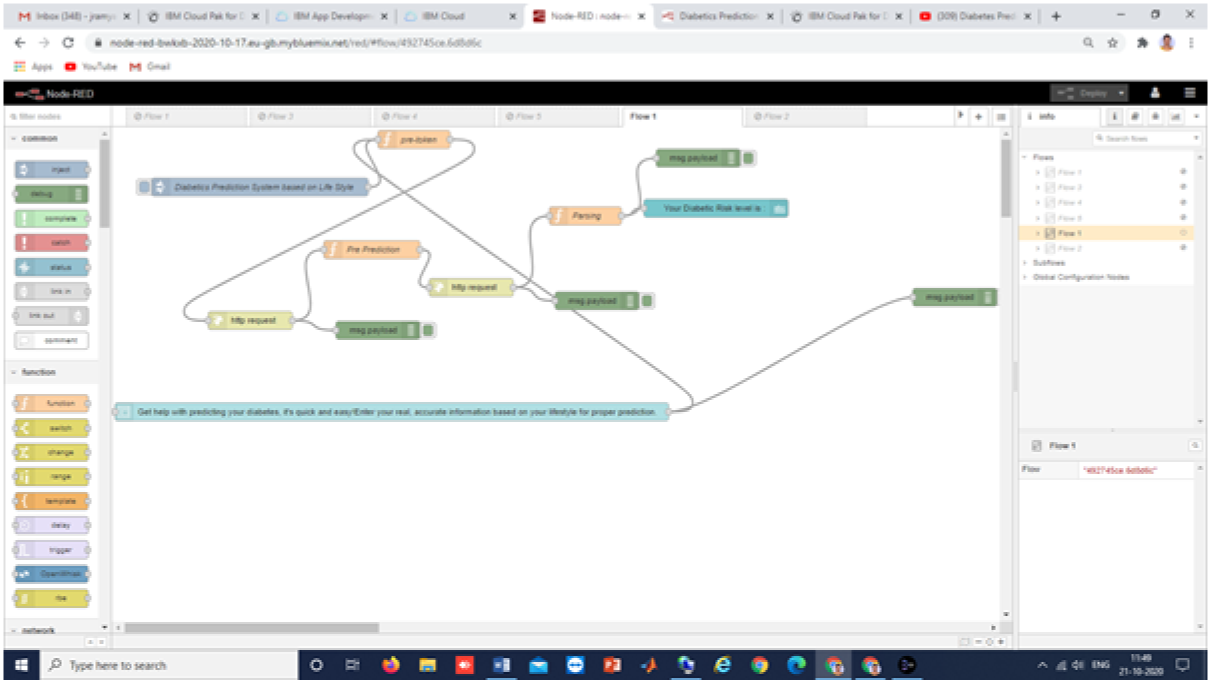
6.7 Confusion Matrix- It gives us gives us a matrix as output and describes the complete performance of the model.



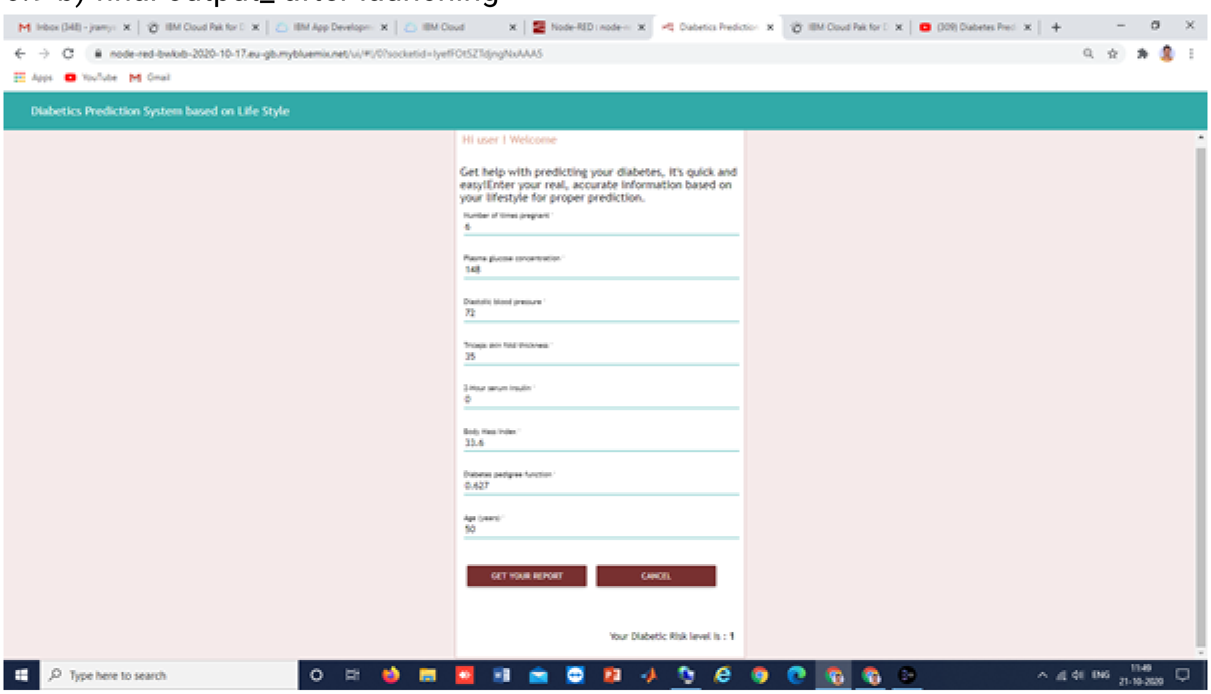
6.8 Classification Accuracy- It is the ratio of number of correct predictions to the total number of input samples.



6.9 a)Building the web application using node_red after deployment



6.9 b) final output_ after launching



7.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Easy to understand and implement
2. Holds good especially when using a graphical representation
3. Works well with numerical and categorical data.

DISADVANTAGES:

1) Harder to tune than different models, since you have so numerous hyper parameters and you can without much of a stretch over fit.
2) Lack of interpretable, contrasted with straight classifiers.

8. APPLICATIONS

- Healthcare
- Patient Monitoring
- Public welfare
- Cost efficiency prediction test

9. CONCLUSION

Machine learning has been extended to many areas of public health through the rapid advancement of machine learning. In recent times, diabetes has become a chronic condition that can cause many complications. There are some signs of diabetes, such as the patient's number of births, their BMI, insulin level, age, etc. Therefore, we suggested an ML-based model that would help solve a serious problem, which predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. For practical health management of diabetes, this model will be useful.Experiments are performed on Pima Indians Diabetes Database. Experimental results determine the adequacy of the designed system with an achieved accuracy of 77 % using the XGBoost classifier by predicting with the diagnostic measurements obtained.

10. FUTURE WORK

The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms.Because of the results, we could not predict the type of diabetes, so we try to predict the type of diabetes in the future and explore the proportion of each feature contribution.

11. BIBILIOGRAPHY

1. Kadhm, Mustafa & Ghindawi, Ikhlas & Enteesha, Duaa. (2018). An Accurate Diabetes Prediction System Based on K-means Clustering and Proposed Classification Approach. International Journal of Applied Engineering Research. 13.
2. Alfian, G.; Syafrudin, M.; Ijaz, M.F.; Syaekhoni, M.A.; Fitriyani, N.L.; Rhee, J. A Personalized Healthcare Monitoring System for Diabetic Patients by Utilizing BLE-Based Sensors and Real-Time Data Processing. Sensors 2018, 18, 2183.
3. De, L & Silva, & Pathirage, Nandana & Jinasena, T & Silva, L & Jinasena, Kasun. (2016). Diabetic Prediction System Using Data Mining
4. Kavakiotis I., Tsave O., Salifoglou A., Maglaveras N., Vlahavas I., Chouvarda I.Machine Learning and Data Mining Methods in Diabetes Research Computational and Structural Biotechnology Journal, 15 (2017), pp. 104-116 doi:10.1016/j.csbj.2016.12.005
5. Joshi, Tejas & Pramila, M & Chawan, Pramila. (2018). Diabetes Prediction Using Machine Learning Techniques. 2248-9622. 10.9790/9622-0801020913.
6. https://www.frontiersin.org/article/10.3389/fgene.2018.00515
7. Jayanthi, N., Babu, B.V. & Rao, N.S. Survey on clinical prediction models for diabetes prediction. J Big Data 4, 26 (2017). https://doi.org/10.1186/s40537-017-0082-7
8. Khan, Nabila & Muaz, Mehedi & Kabir, Anusha & Islam, Muhammad Nazrul. (2017). Diabetes Predicting mHealth Application Using Machine Learning. 237-240. 10.1109/WIECON-ECE.2017.84688
9. P. Suresh Kumar and S. Pranavi "Performance Analysis of Machine Learning Algorithms on Diabetes Dataset using Big Data Analytics",
10. International Conference on Infocom Technologies and Unmanned Systems, 978-1-5386-0514-1, Dec. 18-20, 2017
11. https://www.youtube.com/channel/UCvB8PgOZdb2y7lgToPE-Dfw

APPENDIX

A.SOURCE CODE

Setup

Before you use the sample code in this notebook, you must perform the following setup tasks:

ibm_watson_machine_learning installation

autoai-libs installation/upgrade
lightgbm or xgboost installation/downgrade if they are needed.
!pip install -U ibm-watson-machine-learning
!pip install -U autoai-libs

AutoAI experiment metadata

This cell defines COS credentials required to retrieve AutoAI pipeline.

```
# @hidden_cell
from ibm_watson_machine_learning.helpers import DataConnection, S3Connection,
S3Location

training_data_reference = [DataConnection(
    connection=S3Connection(
        api_key='0sdSWTXm4BcDkotv53Hum4k9fFQrkRzbO2EVgASrmqLA',
        auth_endpoint='https://iam.bluemix.net/oidc/token/',
        endpoint_url='https://s3.eu-geo.objectstorage.softlayer.net'
    ),
        location=S3Location(
        bucket='diabetespredictionbasedonlifestyl-donotdelete-pr-jxmkaptfjhfuwa',
        path='pima-indians-diabetes.csv'
    ))
]
```

```
training_result_reference = DataConnection(
    connection=S3Connection(
        api_key='0sdSWTXm4BcDkotv53Hum4k9fFQrkRzbO2EVgASrmqLA',
        auth_endpoint='https://iam.bluemix.net/oidc/token/',
        endpoint_url='https://s3.eu-geo.objectstorage.softlayer.net'
    ),
    location=S3Location(
        bucket='diabetespredictionbasedonlifestyl-donotdelete-pr-jxmkaptfjhfuwa',

path='auto_ml/563d6aa9-f20e-461f-88d0-bd47d118faee/wml_data/e76d03aa-4d09-4
ae1-8540-d2b618d9aad5/data/automl',

model_location='auto_ml/563d6aa9-f20e-461f-88d0-bd47d118faee/wml_data/e76d03
aa-4d09-4ae1-8540-d2b618d9aad5/data/automl/hpo_c_output/Pipeline1/model.pickl
e',

training_status='auto_ml/563d6aa9-f20e-461f-88d0-bd47d118faee/wml_data/e76d03
aa-4d09-4ae1-8540-d2b618d9aad5/training-status.json'
    ))
```

Following cell contains input parameters provided to run the AutoAI experiment in Watson Studio

```
experiment_metadata = dict(
   prediction_type='classification',
   prediction_column='Outcome',
   test_size=0.1,
   scoring='accuracy',
   project_id='0c2b394e-a184-40a3-a25f-d07d9f212987',
   csv_separator=',',
   random_state=33,
   max_number_of_estimators=2,
   training_data_reference = training_data_reference,
   training_result_reference = training_result_reference,
   deployment_url='https://eu-gb.ml.cloud.ibm.com')

pipeline_name='Pipeline_4'
```

Pipeline inspection
In this section you will get the trained pipeline model from the AutoAI experiment and inspect it.
You will see pipeline as a pythone code, graphically visualized and at the end, you will perform a local test.

Get historical optimizer instance

The next cell contains code for retrieving fitted optimizer.

```
from ibm_watson_machine_learning.experiment import AutoAI

optimizer = AutoAI().runs.get_optimizer(metadata=experiment_metadata)
```

Get pipeline model

The following cell loads selected AutoAI pipeline model. If you want to get pure scikit-learn pipeline specify as_type='sklearn' parameter. By default enriched scikit-learn pipeline is returned as_type='lale'.

```
pipeline_model = optimizer.get_pipeline(pipeline_name=pipeline_name)
```

Preview pipeline model as python code

In the next cell, downloaded pipeline model could be previewed as a python code. You will be able to see what exact steps are involved in model creation.

```
pipeline_model.pretty_print(combinators=False, ipython_display=True)
```

Visualize pipeline model

Preview pipeline model stages as graph. Each node's name links to detailed description of the stage.

```
pipeline_model.visualize()
```

Read training data

Retrieve training dataset from AutoAI experiment as pandas DataFrame.

```
train_df = optimizer.get_data_connections()[0].read()
test_df = train_df.sample(n=5).drop([experiment_metadata['prediction_column']],
axis=1)
```

Test pipeline model locally
You can predict target value using trained AutoAI model by calling predict().

```
y_pred = pipeline_model.predict(test_df.values)
print(y_pred)
```

Pipeline refinery and testing (optional)

In this section you will learn how to refine and retrain the best pipeline returned by AutoAI. It can be performed by:

modifying pipeline definition source code

using lale library for semi-automated data science
Note: In order to run this section change following cells to 'code' cell.

Pipeline definition source code

Following cell lets you experiment with pipeline definition in python, e.g. change steps parameters.

It will inject pipeline definition to the next cell.

```
pipeline_model.pretty_print(combinators=False, ipython_display='input')
```

Connection to WML

Authenticate the Watson Machine Learning service on IBM Cloud.

Action: Enter your api_key in the following cell.

```
api_key = "PUT_YOUR_API_KEY_HERE"

wml_credentials = {
  "apikey": api_key,
  "url": experiment_metadata["deployment_url"]
}
```

Create deployment

Action: If you want to deploy refined pipeline please change the pipeline_name to new_pipeline. If you prefer you can also change the deployment_name.

Action: To perform deployment please specify target_space_id.

```
target_space_id = "PUT_YOUR_TARGET_SPACE_ID_HERE"

from ibm_watson_machine_learning.deployment import WebService
service = WebService(source_wml_credentials=wml_credentials,
                     target_wml_credentials=wml_credentials,
                     source_project_id=experiment_metadata['project_id'],
                     target_space_id=target_space_id)
service.create(
model=pipeline_name,
metadata=experiment_metadata,
deployment_name=f'{pipeline_name}_webservice'
)
```

Deployment object could be printed to show basic information:

```
print(service)
```

To be able to show all available information about deployment use .get_params() method:

service.get_params()

Score webservice
You can make scoring request by calling score() on deployed pipeline.

predictions = service.score(payload=test_df)
predictions
If you want to work with the webservice in external Python application you can retrieve the service object by:

initialize service by:

service = WebService(target_wml_credentials=wml_credentials,
 target_space_id=target_space_id)
get deployment_id by service.list() method
get webservice object by service.get('deployment_id') method
After that you can call service.score() method.

Delete deployment

You can delete an existing deployment by calling service.delete().

flow.json

[{"id":"903a8a75.c814a8","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"c3077698.1f60b8","type":"ui_form","z":"903a8a75.c814a8","name":"","label":"","group":"de4835dd.3d6508","order":1,"width":0,"height":0,"options":[{"label":"preg","value":"p","type":"number","required":true,"rows":null},{"label":"plas","value":"pl","type":"number","required":true,"rows":null},{"label":"pres","value":"pr","type":"number","required":true,"rows":null},{"label":"Skin ","value":"s","type":"number","required":true,"rows":null},{"label":"test","value":"t","type":"number","required":true,"rows":null},{"label":"mass","value":"m","type":"number","required":true,"rows":null},{"label":"pedi","value":"pe","type":"number","required":true,"rows":null},{"label":"age","value":"a","type":"number","required":true,"rows":null}],"formValue":{"p":"","pl":"","pr":"","s":"","t":"","m":"","pe":"","a":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":70,"y":280,"wires":[["3a7f8360.5298ac","2770dfbc.e93ca"]]},{"id":"3a7f8360.5298ac","ty

pe":"debug","z":"903a8a75.c814a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":250,"y":360,"wires":[]},{"id":"2770dfbc.e93ca","type":"function","z":"903a8a75.c814a8","name":"pre-token","func":"global.set(\"p\",msg.payload.p)\nglobal.set(\"pl\",msg.payload.pl)\nglobal.set(\"pr\",msg.payload.pr)\nglobal.set(\"s\",msg.payload.s)\nglobal.set(\"t\",msg.payload.t)\nglobal.set(\"m\",msg.payload.m)\nglobal.set(\"pe\",msg.payload.pe)\nglobal.set(\"a\",msg.payload.a)\nvar
apikey=\"wzJfMjpLlvM1Ta31pnzmzetCK9oo8duUP7N4Q2yc6gR4\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn
msg;\n","outputs":1,"noerr":0,"initialize":"","finalize":"","x":260,"y":220,"wires":[["32c483d.430857c"]]},{"id":"32c483d.430857c","type":"http
request","z":"903a8a75.c814a8","name":"","method":"POST","ret":"obj","paytoqs":"ignore","url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"","x":412.00000381469727,"y":264.00000286102295,"wires":[["e099eb89.aa4258","aba60e90.7c137"]]},{"id":"e099eb89.aa4258","type":"debug","z":"903a8a75.c814a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":600,"y":320,"wires":[]},{"id":"aba60e90.7c137","type":"function","z":"903a8a75.c814a8","name":"Pre Prediction","func":"var p = global.get(\"p\")\nvar
pl = global.get(\"pl\")\nvar pr = global.get(\"pr\")\nvar s = global.get(\"s\")\nvar t =
global.get(\"t\")\nvar m = global.get(\"m\")\nvar pe = global.get(\"pe\")\nvar a =
global.get(\"a\")\nvar
token=msg.payload.access_token\nmsg.headers={'Content-Type':
'application/json',\"Authorization\":\"Bearer
\"+token,\"Accept\":\"application/json\"}\nmsg.payload={\"input_data\":[{\"fields\":
[[\"preg\", \" plas\", \"pres\", \"Skin\", \"test\", \"mass\",\"pedi\", \"age\"]],\"values\":
[[p,pl,pr,s,t,m,p,a]]}]}\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":604.0000076293945,"y":216.0000021457672,"wires":[["10c118b8.ecef17"]]},{"id":"10c118b8.ecef17","type":"http
request","z":"903a8a75.c814a8","name":"","method":"POST","ret":"obj","paytoqs":"ignore","url":"https://us-south.ml.cloud.ibm.com/ml/v4/deployments/117ed57c-184d-483e-a5b8-2bc5c83e1374/predictions?version=2020-09-01","tls":"","persist":false,"proxy":"","authType":"","x":830,"y":240,"wires":[["9ff10849.8c96a8","115cea0c.d78446"]]},{"id":"9ff10849.8c96a8","type":"debug","z":"903a8a75.c814a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":982.9999914169312,"y":324.00000381469727,"wires":[]},{"id":"6ae67dcc.416e94","type":"inject","z":"903a8a75.c814a8","name":"","props":[{"p":"payload"},{"p":"topic","vt":"str"}],"repeat"

:"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":120,"wires":[["2770dfbc.e93ca"]]},{"id":"7e4eed17.357514","type":"debug","z":"903a8a75.c814a8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","statusVal":"","statusType":"auto","x":813.9999885559082,"y":107.0000057220459,"wires":[]},{"id":"115cea0c.d78446","type":"function","z":"903a8a75.c814a8","name":"Parsing","func":"msg.payload=msg.payload.predictions[0].values[0][0]\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","x":659.6000595092773,"y":148.00000190734863,"wires":[["7e4eed17.357514","c93e1b4c.8f1998"]]},{"id":"c93e1b4c.8f1998","type":"ui_text","z":"903a8a75.c814a8","group":"de4835dd.3d6508","order":2,"width":0,"height":0,"name":"","label":"Prediction","format":"{{msg.payload}}","layout":"row-spread","x":777.6000366210938,"y":381.20001220703125,"wires":[]},{"id":"de4835dd.3d6508","type":"ui_group","z":"","name":"New","tab":"56be333f.3550fc","order":1,"disp":true,"width":"6","collapse":false},{"id":"56be333f.3550fc","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

source code ends here.