**Predict heart failure using IBM auto AI service**

**Introduction:-**

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide.

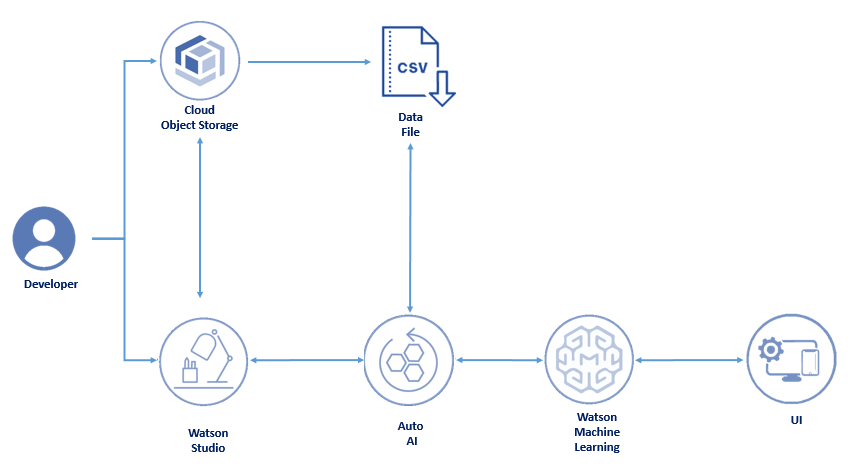
Heart failure is a common event caused by CVDs and this dataset contains 9 features that can be used to predict mortality by heart failure.

In this project, we need to build a model using Auto AI and build a web application where we can showcase the prediction of heart failure.

**Services Used:**

* IBM Watson Studio
* IBM Watson Machine Learning
* Node-RED
* IBM Cloud Object Storage

**Technical Architecture:**



**Project Objectives**

After completing this project, we have learn how to

* Work with Watson Studio
* Create a project in Watson Studio
* Use Auto Ai experiment to create a model
* Deploy the ML model as a webserver
* Integrating Model and Node-RED Service
* Build an Application using Node-RED which takes inputs from the user and showcases the prediction on UI

**Project Flow**

**This is the project flow to complete the project**

1. Log in to IBM account
2. Create IBM Watson Studio and Node-RED Service
3. Create a Watson studio project
4. ADD Auto AI Experiment
5. Run the Auto AI Experiment to build a Machine learning model on the desired dataset
6. Save the model
7. Deploy the model as a web server and generate scoring End Point
8. Create a WEB application Using Node-RED to take user input and showcase Prediction on UI

**Create IBM Services**

In this activity, we have created the Necessary IBM services. The following are the services that we have to create.

* Watson studio
* Node-RED
* Cloud Object Storage service (COS)
* Machine Learning service (ML)

COS and ML services have e-created while creating a Watson Studio Project.

**Build Machine learning Model**

In this activity, we have learn how we can save a pipeline as a Watson Machine Learning model, deploy the model, and score it to view a prediction.

This Activity  contains the Following Tasks

* Collect the data set
* Create Watson Studio project
* Add Auto AI experiment
* Run AI Experiment
* Save the model
* Deploy the model

**Collect the data set**

In this Activity, We gonna build a machine learning model that predicts  heart failure  based on the following parameters

* AVGHEARTBEATSPERMIN
* PALPITATIONSPERDAY
* CHOLESTEROL
* BMI
* AGE
* SEX
* FAMILY HISTORY
* SMOKERLAST5YRS
* EXERCISEMINPERWEEK

**Add Auto Ai Experiment**

The AutoAI graphical tool in Watson Studio automatically analyzes wer data and generates candidate model pipelines customized for wer predictive modeling problem.  These model pipelines are created iteratively as AutoAI analyzes wer dataset and discovers data transformations, algorithms, and parameter settings that work best for wer problem setting.  Results are displayed on a leaderboard, showing the automatically generated model pipelines ranked according to wer problem optimization objective.

To add the project

* Click on Add Project from Watson Studio project
* Select Auto AI
* Specify a name and description for wer experiment.
* Select a machine learning service instance and click Create.

### Run Auto Ai Experiment

Upload the heart failure dataset that we have downloaded and select HEART FAILURE as the prediction column

### Save The model

Once the pipeline creation is complete, we can view and compare the ranked pipelines in a leaderboard.

Choose Save model from the action menu for the pipeline with the highest accuracy or low error rate This saves the pipeline as a Machine Learning asset in wer project. A notification gives we the link to view the saved model in wer project.

**Deploy the Model**

Before we can use wer trained model to make predictions on new data, we must deploy the model.

We can deploy the model from the model details page. We can access the model details page in one of these ways:

* Click on the model name in the notification displayed when we save the model.
* Open the Assets page for the project containing the model and click the model name in the Machine Learning Model section.

From the model details page:

* Click the Promote to deployment space.
* Choose an existing deployment space or create a new one.
* Click Add Deployment.
* In the page that opens, fill in the fields:
* Specify a name for the deployment.
* Select “Web service” as the Deployment type.
* Click Save.

After we save the deployment, click on the deployment name to view the deployment details page

### Build Node-RED Application

Let's build a User interface which takes inputs from the user. The Model Analyses the Inputs and returns the Prediction that is showcased on the User interface

### Integrate Node-RED With Auto Ai Model

In this task, we have be creating a Node-RED flow that integrated with the model built

Create a form node which has the Input values according to the dataset that we have taken

CODE Node-Red:

GLOBAL VARIABLE NODE:-

global.set("avgheartratepermin",msg.payload.a)

global.set("palpitationperday",msg.payload.p)

global.set("cholestrol",msg.payload.c)

global.set("bmi",msg.payload.b)

global.set("age",msg.payload.a1)

global.set("exerciseperweek",msg.payload.e)

var apikey="TAOngRQepl84-JegfNDKX96z69IAM1tF3dIdJbM5cZcJ"

msg.headers={"content-type":"application/x-www-form-urlencoded"}

msg.payload={"grant\_type":"urn:ibm:params:oauth:grant-type:apikey","apikey":apikey}

return msg;

**NODE http request:**

URL: <https://iam.cloud.ibm.com/identity/token>

Method: POST

Return parsed JSON object

**NODE FUNCTION CODE:-**

var age=global.get("age")

var avgheartratepermin=global.get("avgheartratepermin")

var palpitationperday=global.get("palpitationperday")

var cholestrol=global.get("cholestrol")

var bmi=global.get("bmi")

var exerciseperweek=global.get("exerciseperweek")

var sex=global.get("sex")

var familyhistory=global.get("familyhistory")

var somker=global.set("smoker")

var token=msg.payload.access\_token

msg.headers={'Content-Type':'application/json',"Authorization":"Bearer"+token,"Accept":"application/json"}

msg.payload={

"input\_data": [

{"fields": [

"AVGHEARTBEATSPERMIN",

"PALPITATIONSPERDAY",

"CHOLESTEROL",

"BMI",

"AGE",

"SEX",

"FAMILYHISTORY",

"SMOKERLAST5YRS",

"EXERCISEMINPERWEEK"

],"values": [

[avgheartratepermin,

palpitationperday,

cholestrol,

bmi,

age,

sex,

familyhistory,

somker,

exerciseperweek

]

]}]}

return msg;

**NODE http request AUTO AI experiment:**

Method:- POST

URL:  [https://us-south.ml.cloud.ibm.com/ml/v4/deployments/73d5ad42-1e60-456f-894c-18bc4001aa5c/predictions?version=2021-07-09](%20https://us-south.ml.cloud.ibm.com/ml/v4/deployments/73d5ad42-1e60-456f-894c-18bc4001aa5c/predictions?version=2021-07-09)

Return: parsed JSON object

**NODE heartrisk:**

msg.payload=msg.payload.predictions[0].values[0][0]

return msg;

**NODE Prediction:-**

msg.payload=msg.payload.predictions[0].values[0][1][0]\*100

return msg;

**NODE Probability:-**

msg.payload=msg.payload.predictions[0].values[0][1][1]\*100

return msg;