Diabetic Prediction System based on Life Style

INTRODUCTION: Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes mellitus or simply diabetes is a disease caused due to the increase level of blood glucose. Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc. Data science methods have the potential to benefit other scientific fields by shedding new light on common questions. One such task is to help make predictions on medical data. Machine learning is an emerging scientific field in data science dealing with the ways in which machines learn from experience. The aim of this project is to develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by combining the results of different machine learning techniques. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

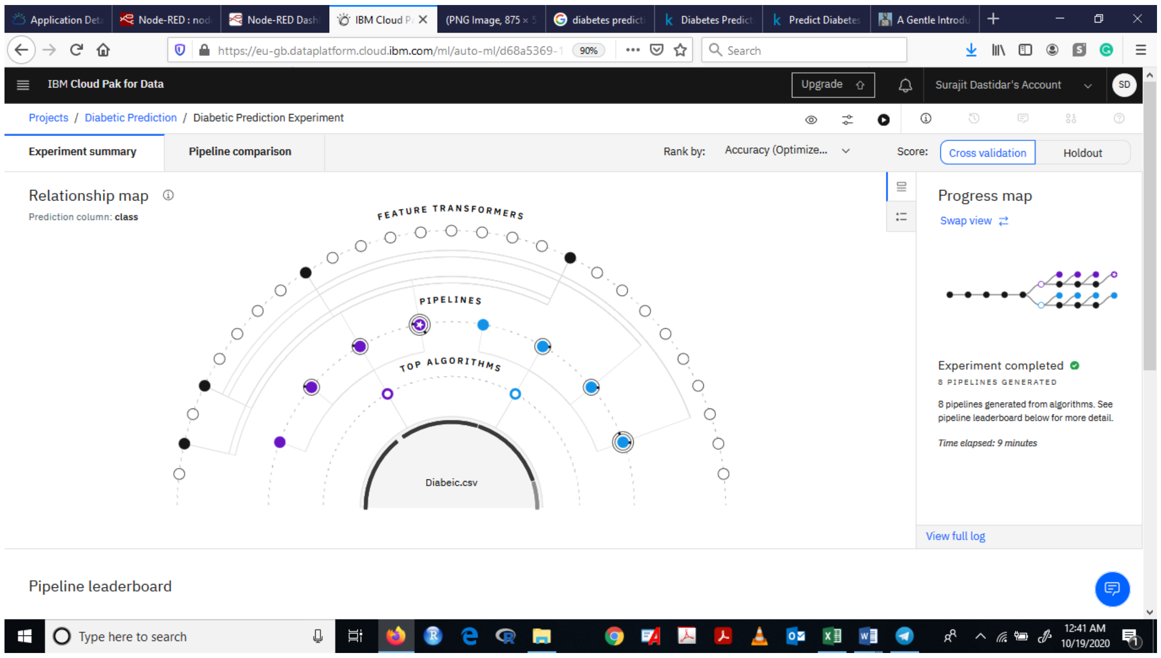
Solution Requirements:

Develop an end-to-end web application that predicts the probability of females having diabetes. The application must be built with the machine learning model trained & deployed on IBM Watson Studio and node-red web application

Data: The data file for the project was downloaded from the source as given. The modified data file named diabeic.csv has been uploaded in the GitHub. I took out the id column from the original data file manually. The datasets consist of several medical predictor variables and one target variable, Diabetes. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Methodology: The best prediction model came out to be **Gradient Boosting Classifier** model as found by the Auto AI experiment with Accuracy as the performance criteria. (Refer to Figure 1 and 2). The variable “class” was chosen as the target variable. The class variable was a binary variable and can take numeric values 0 and 1 only. The value of 1 indicates that the patient is predicted to be diabetic indicated by 1 and 0 otherwise.

Figure - 1



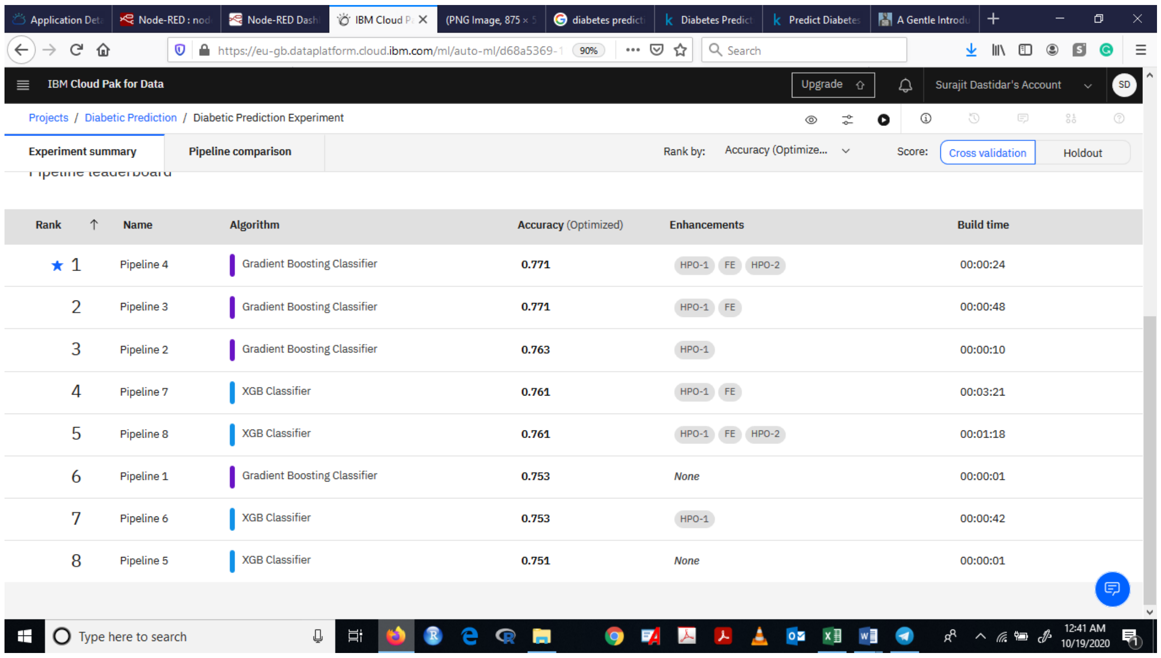
Experiment completed

8 PIPELINES GENERATED

8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 9 minutes

Figure - 2


IBM Cloud Pak for Data
Upgrade
Surajit Dastidar's Account
Projects / Diabetic Prediction / Diabetic Prediction Experiment
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimize...
Score: Cross validation Holdout

Rank	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time
★ 1	Pipeline 4	Gradient Boosting Classifier	0.771	HPO-1 FE HPO-2	00:00:24
2	Pipeline 3	Gradient Boosting Classifier	0.771	HPO-1 FE	00:00:48
3	Pipeline 2	Gradient Boosting Classifier	0.763	HPO-1	00:00:10
4	Pipeline 7	XGB Classifier	0.761	HPO-1 FE	00:03:21
5	Pipeline 8	XGB Classifier	0.761	HPO-1 FE HPO-2	00:01:18
6	Pipeline 1	Gradient Boosting Classifier	0.753	*None*	00:00:01
7	Pipeline 6	XGB Classifier	0.753	HPO-1	00:00:42
8	Pipeline 5	XGB Classifier	0.751	*None*	00:00:01

How Gradient Boosting Works?

Gradient boosting involves three elements:

1. A loss function to be optimized.
2. A weak learner to make predictions.
3. An additive model to add weak learners to minimize the loss function.

1. Loss Function

The loss function used depends on the type of problem being solved.

It must be differentiable, but many standard loss functions are supported and you can define your own.

For example, regression may use a squared error and classification may use logarithmic loss.

A benefit of the gradient boosting framework is that a new boosting algorithm does not have to be derived for each loss function that may want to be used, instead, it is a generic enough framework that any differentiable loss function can be used.

2. Weak Learner

Decision trees are used as the weak learner in gradient boosting. Specifically, regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and "correct" the residuals in the predictions.

Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss.

Initially, such as in the case of AdaBoost, very short decision trees were used that only had a single split, called a decision stump. Larger trees can be used generally with 4-to-8 levels.

It is common to constrain the weak learners in specific ways, such as a maximum number of layers, nodes, splits or leaf nodes.

This is to ensure that the learners remain weak, but can still be constructed in a greedy manner.

3. Additive Model

Trees are added one at a time, and existing trees in the model are not changed.

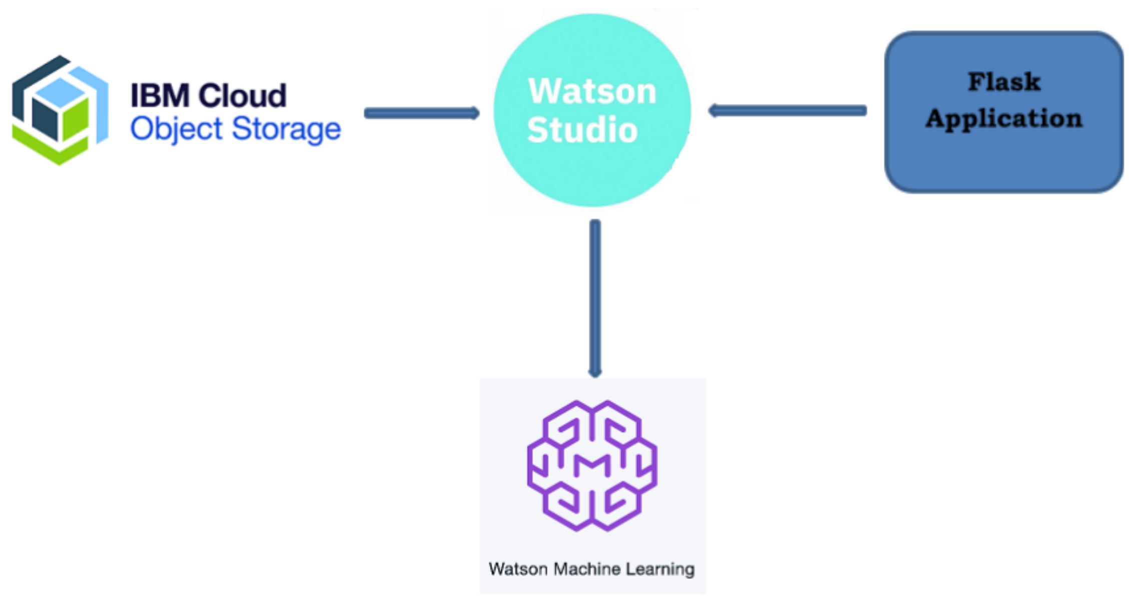
A gradient descent procedure is used to minimize the loss when adding trees.

Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error.

Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure, we must add a tree to the model that reduces the loss (i.e. follow the gradient). We do this by parameterizing the tree, then modify the parameters of the tree and move in the right direction by (reducing the residual loss.

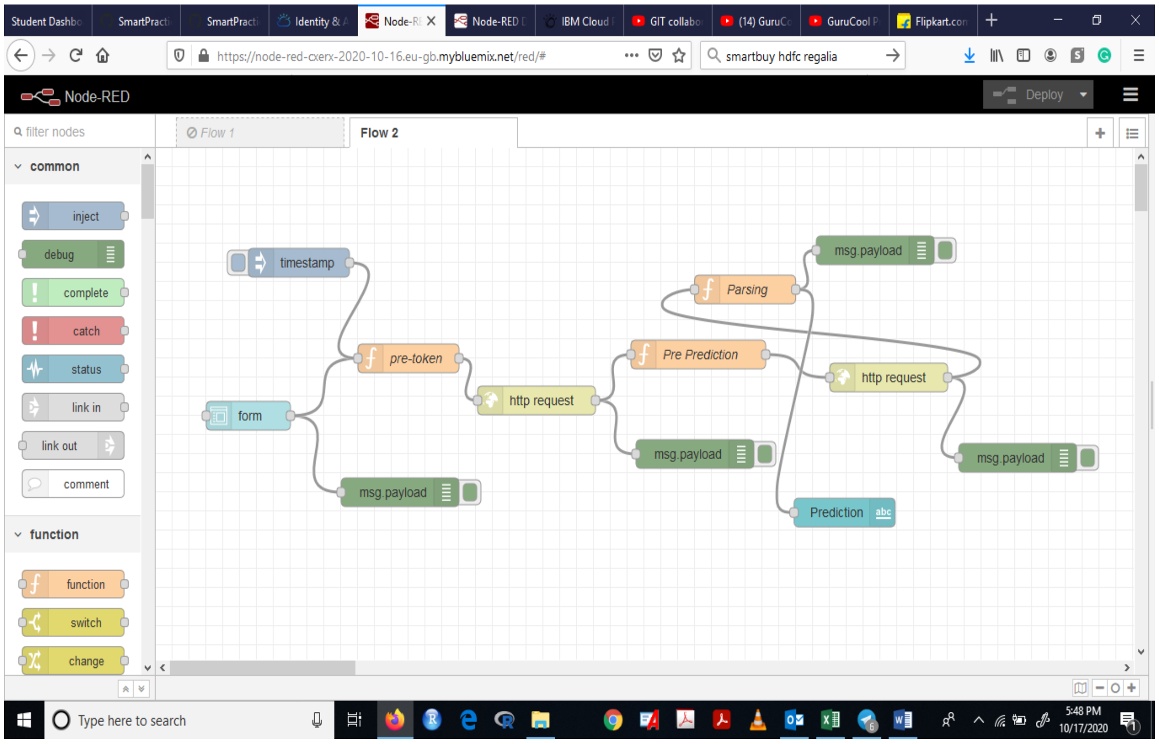
Generally this approach is called functional gradient descent or gradient descent with functions.

ARCHITECTURE:



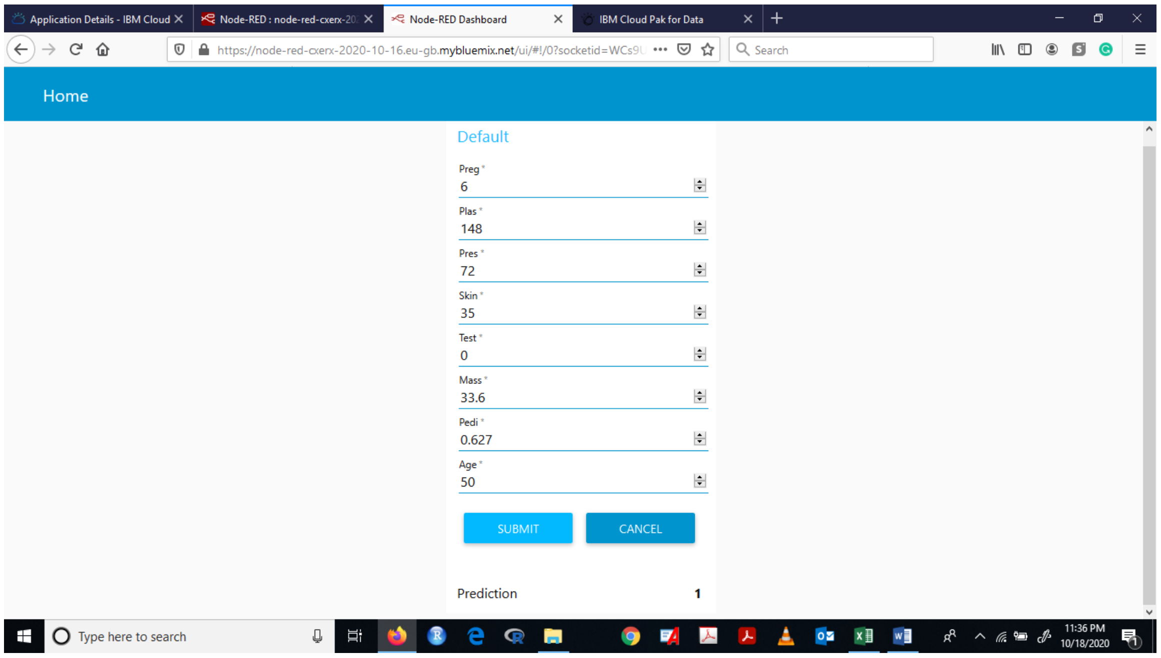
FLOW DIAGRAM:

Please find the NODE RED FLOW DIAGRAM below:



USER INTERFACE USING NODE-RED:

Case-1: Predicting Diabetic Patients correctly (class = 1) with the given data

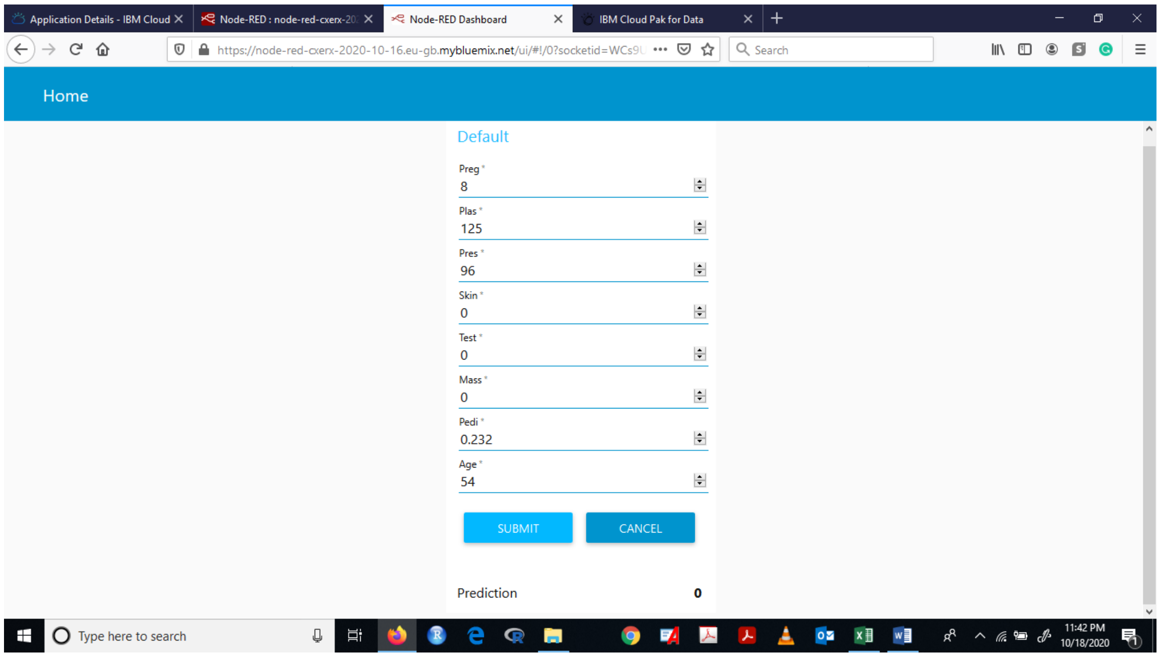

Default

Preg * 6
Plas * 148
Pres * 72
Skin * 35
Test * 0
Mass * 33.6
Pedi * 0.627
Age * 50

Prediction 1

URL:https://node-red-cxerx-2020-10-16.eu-gb.mybluemix.net/ui/#!/0?socketid=WCs9Ud1sU7LdmE8gAAAM

Case-II: Predicting Non-Diabetic Patients correctly (class=0) with the given data


Default

Preg * 8
Plas * 125
Pres * 96
Skin * 0
Test * 0
Mass * 0
Pedi * 0.232
Age * 54

Prediction 0

DISCUSSION:

A Gradient Boosting Classifier model was built using IBM Watson that proved to be the best model for diabetic prediction based on the given data with an accuracy of 77.1%.. The web interface was deployed using a NODE-RED application in which users are expected to enter the required data. Similar kind of prediction models can be built in various application areas like medicine, revenue forecasting, fraud detection, marketing, operations and finance.