**AWS Build-a-thon Project Report**

ON

**“DIABETICS PREDICTION MODELFOR INDIAN WOMEN** **”**

Submitted for the aws build-a-thon of

**Machine Learning**

BY

**NAME: RAHUL P.BHERE**

**PROJECT ID: SPS\_PRO\_101**

**A project report submitted to:**

Smart Bridge



**ACKNOWLEDGMENT:**

The success and outcome of this project required guidance and assistance from **Smart Bridge**. I am extremely privileged to have got this all along with the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them. I thank **Smart Bridge**, for providing me an opportunity to do the project work in the AWS build-a-thon **Diabetes Prediction Model For Indian Women** and giving me all support and guidance which made me complete the project duly. I am extremely thankful to smart bridge for providing such a nice support and guidance.

**1. INTRODUCTION**

**1.1 Web Summarization**

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. However, early predict on of diabetes is quite challenging task for medical 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 prediction on medical data.

This project also aims to propose an effective technique for earlier detection of the diabetes disease.

**1.2 Purpose**

In this, we need to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

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.

Develop an end-to-end web application that predicts the probability of females having diabetes. The application must be built with Python-Flask or Django framework with the machine learning model trained & deployed on AWS Sagemaker. Create an API Endpoint for the model with the help of API Gateway and AWS Lambda Service.

# 2. LITERATURE SURVEY

# 2.1 Existing problem

In this, we need to diagnosis predict whether or not a parent has diabetes, based on certain diagnosis c measurements included in the dataset. Several constraints were placed on the select on of these instances from a larger database. In particular, all parents here are females at least 21 years old of Pima

Indian heritage.

The datasets consist of several medical predictor variables and one target variable, Diabetes. Predictor variables include the number of pregnancies the parent has had, their BMI, insulin level, age, and so on.

# 2.2 Proposed solution

Develop an end-to-end web application that predicts the probability of females having diabetes. The application must be built with Python-Flask or Django framework with the machine learning model trained & deployed on AWS sagemaker.

Create an API Endpoint for the model with the help of API Gateway and AWS Lambda Service.

# 3.THEORETICAL ANALYSIS

# 3.1 PROJECT SCOPE:

In this, we need to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

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.

## **3.2 Block Digram**

***Fig. 2.****Block Digram*

**3.3 Software designing**

**1. AWS Cloud**

**2. Amazon S3**

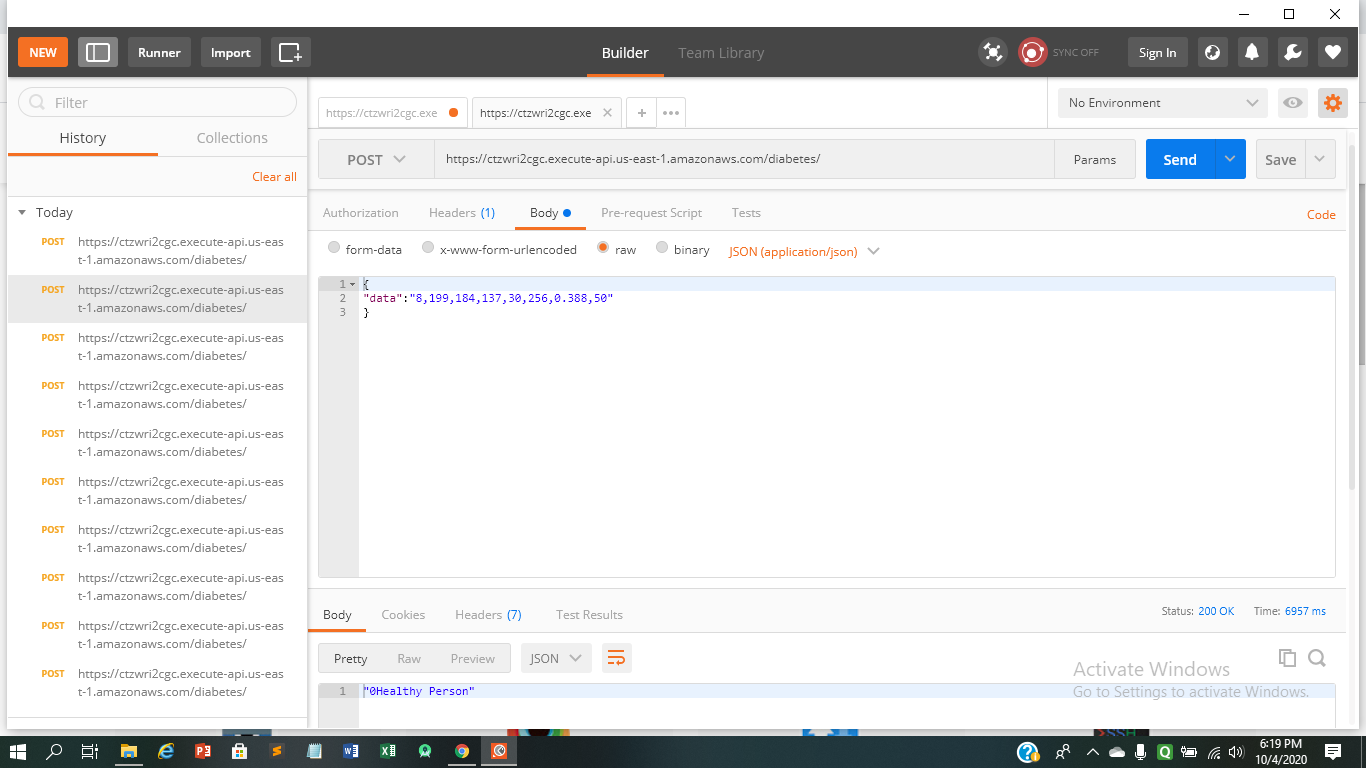
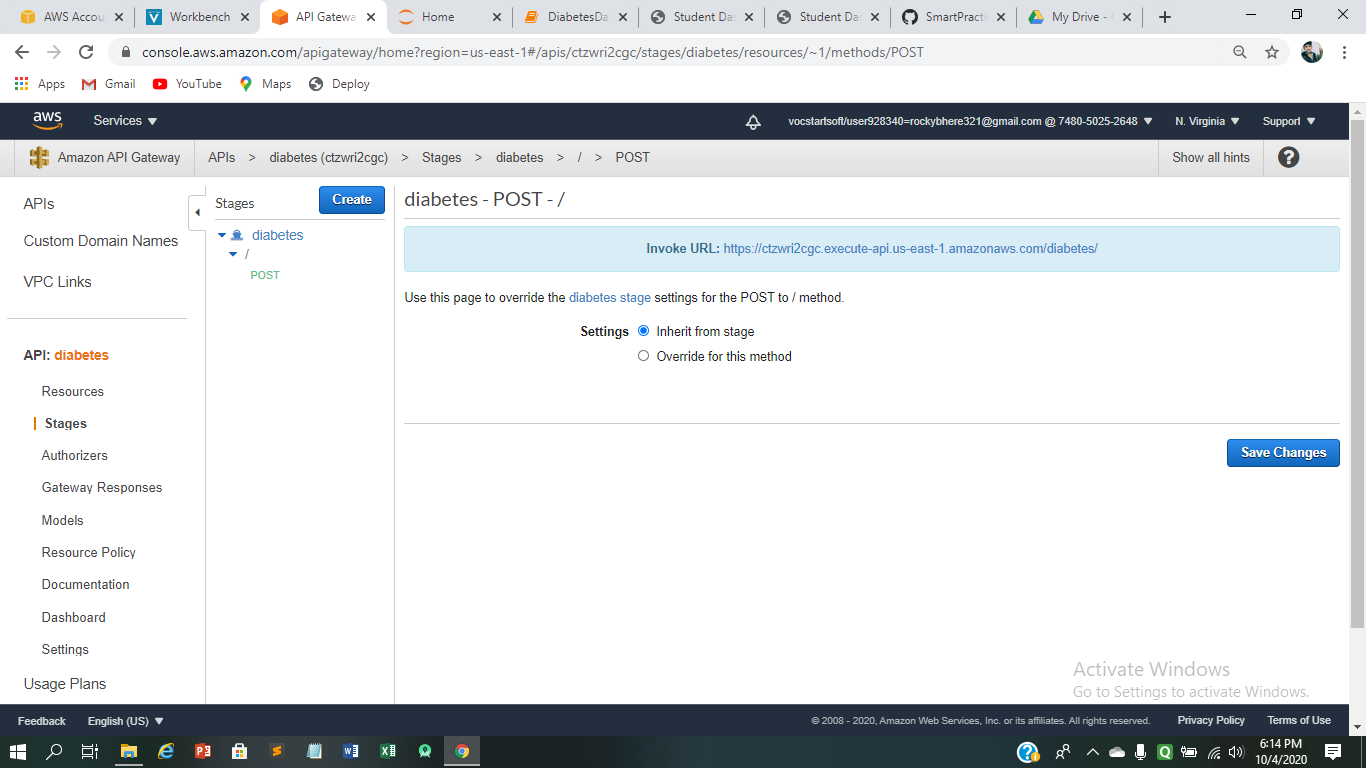
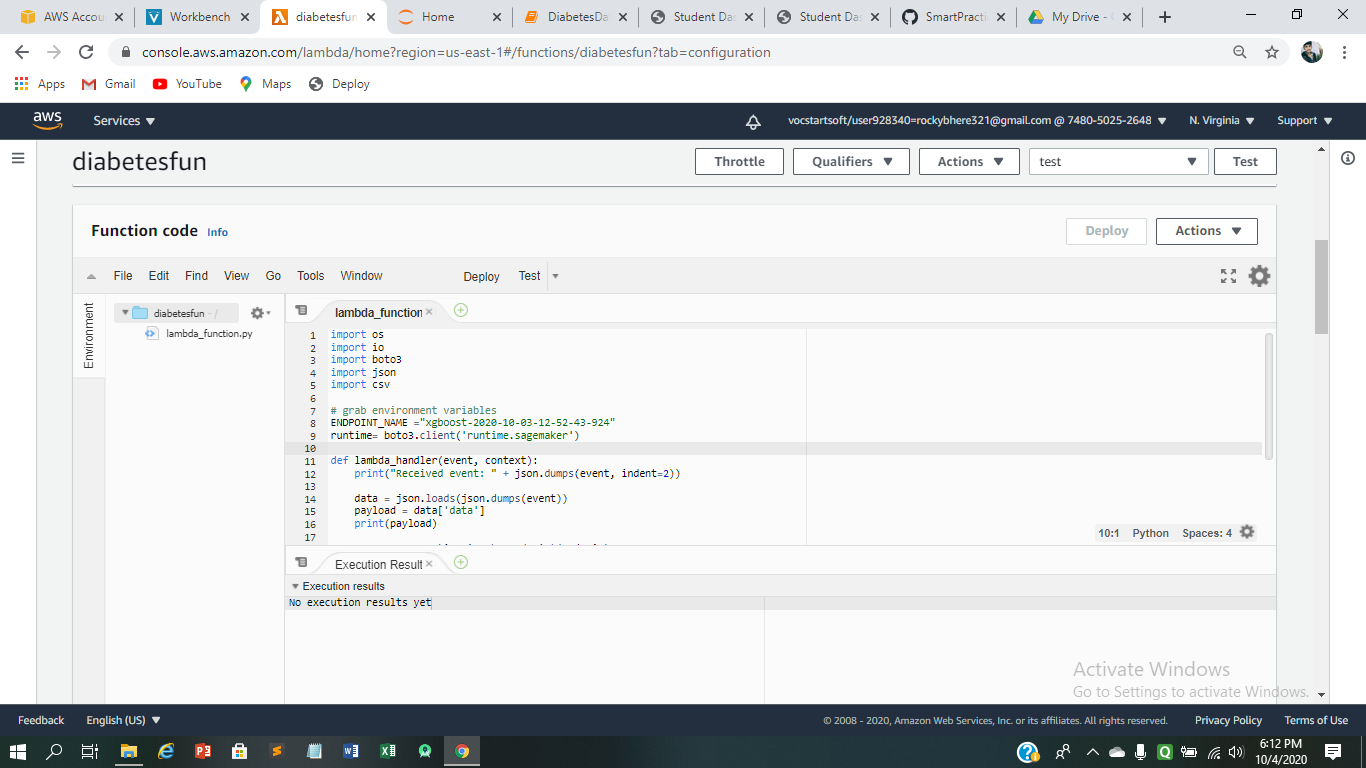
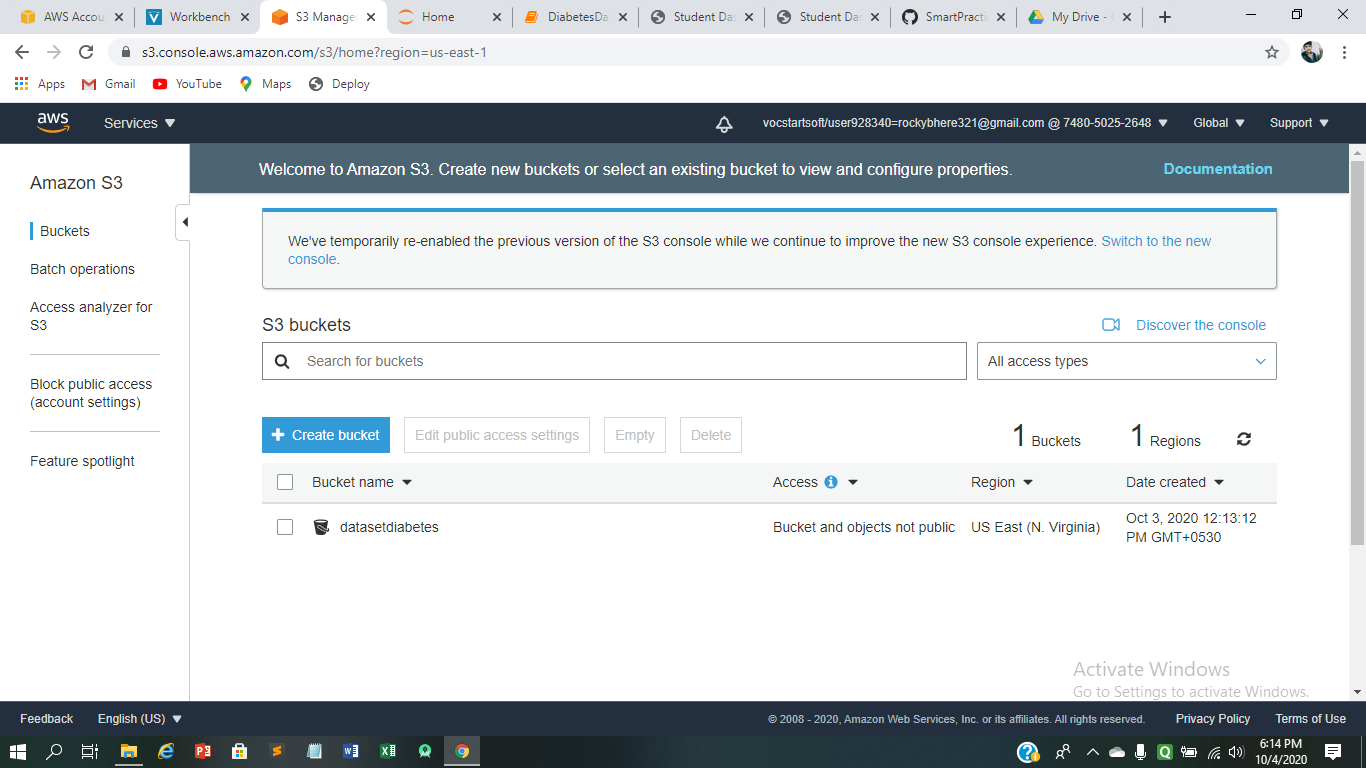
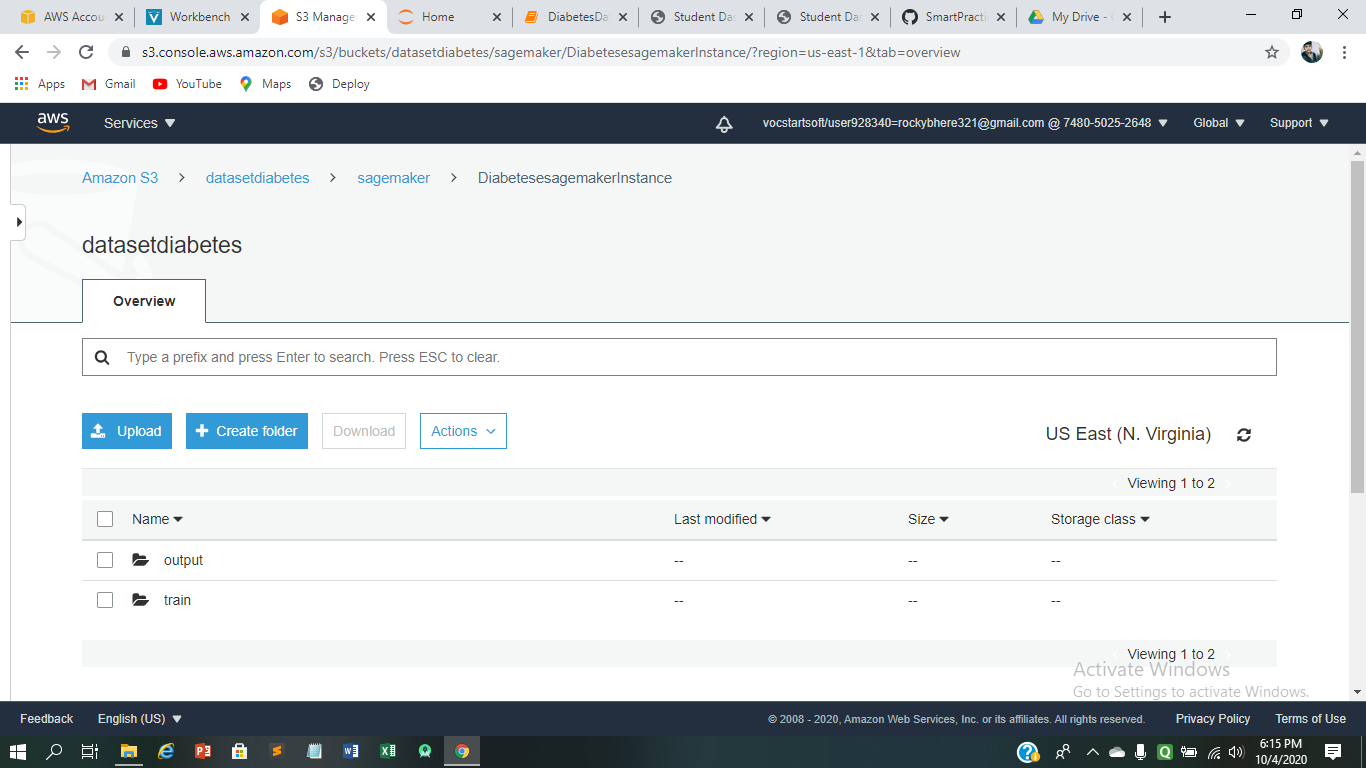
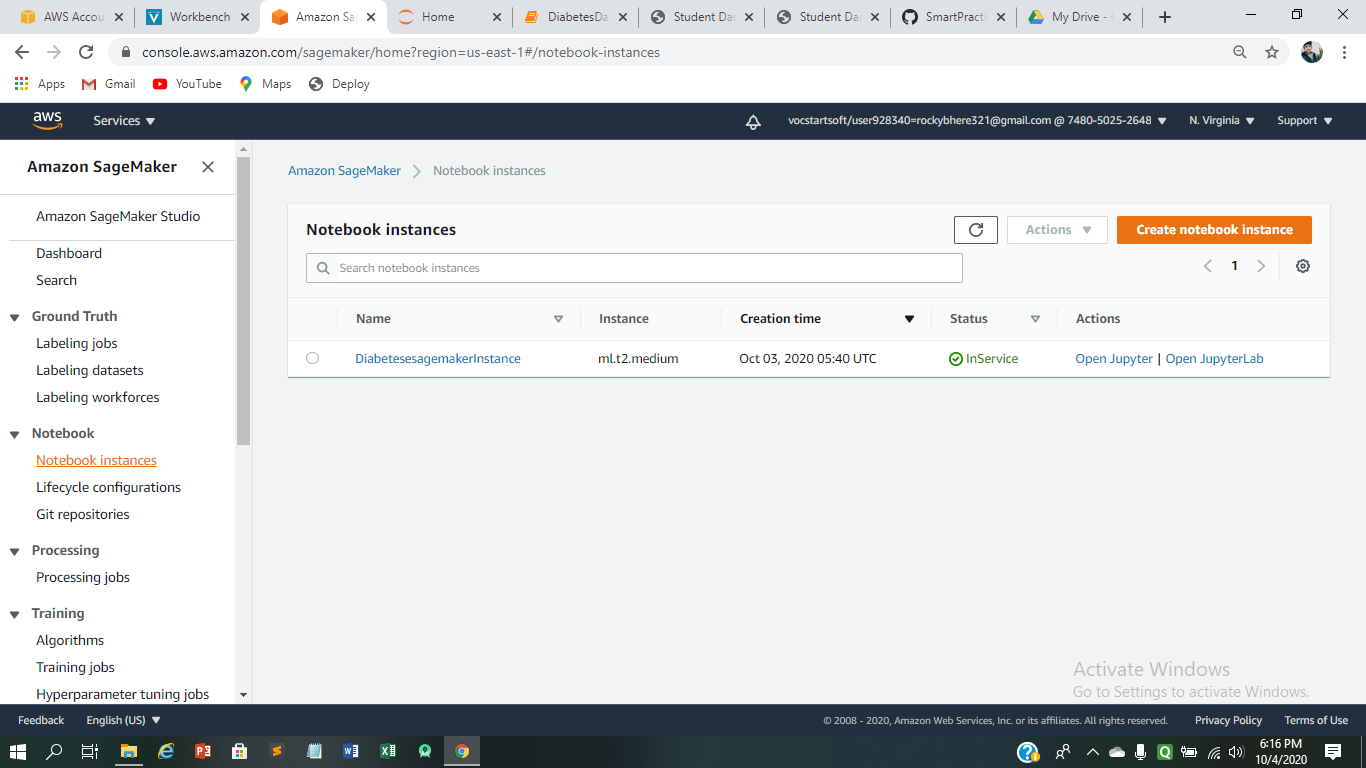
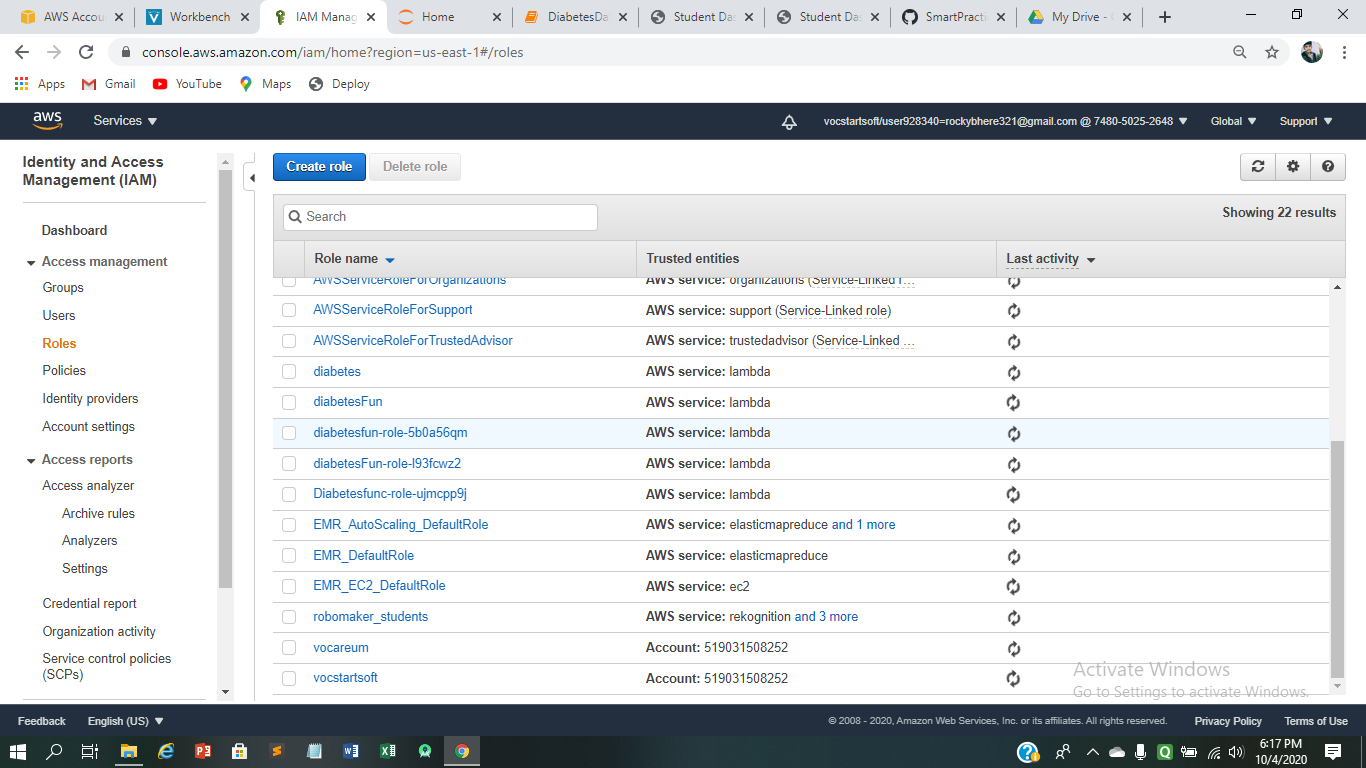
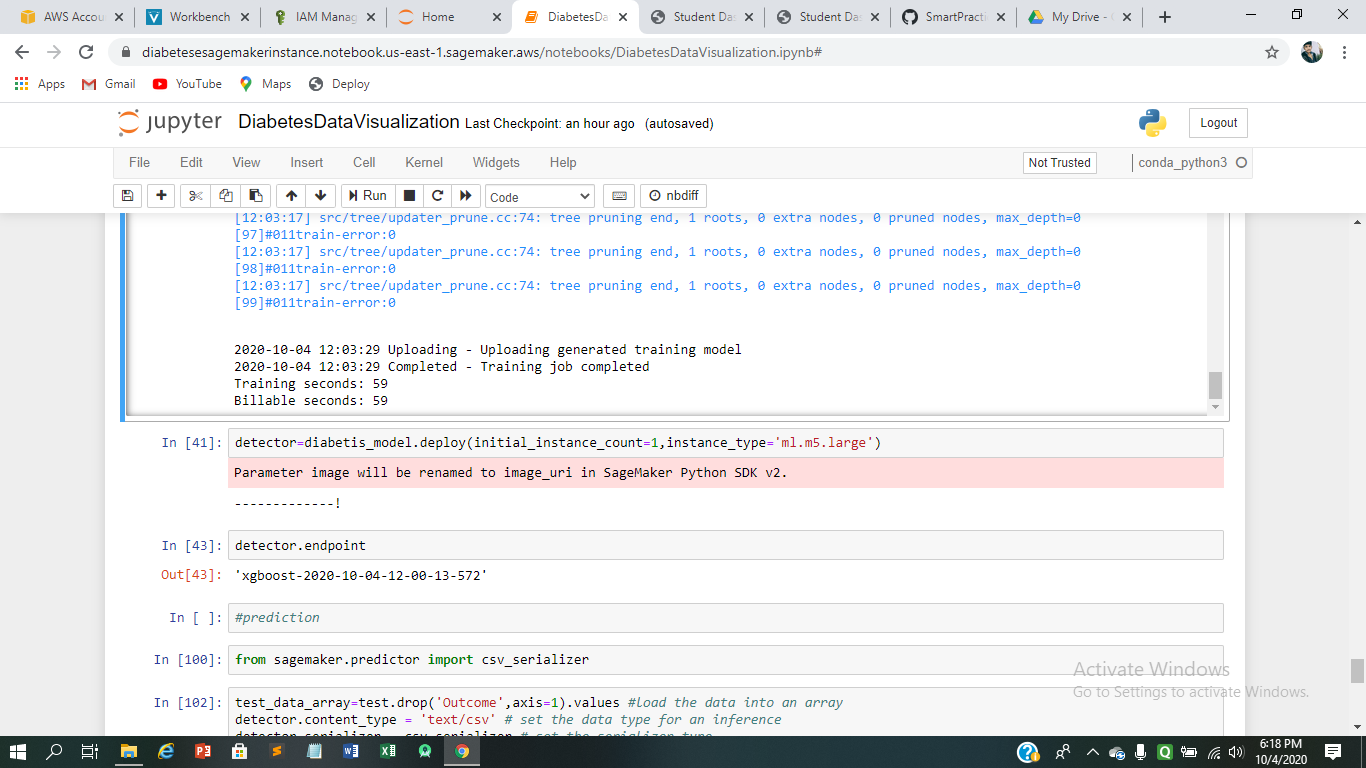
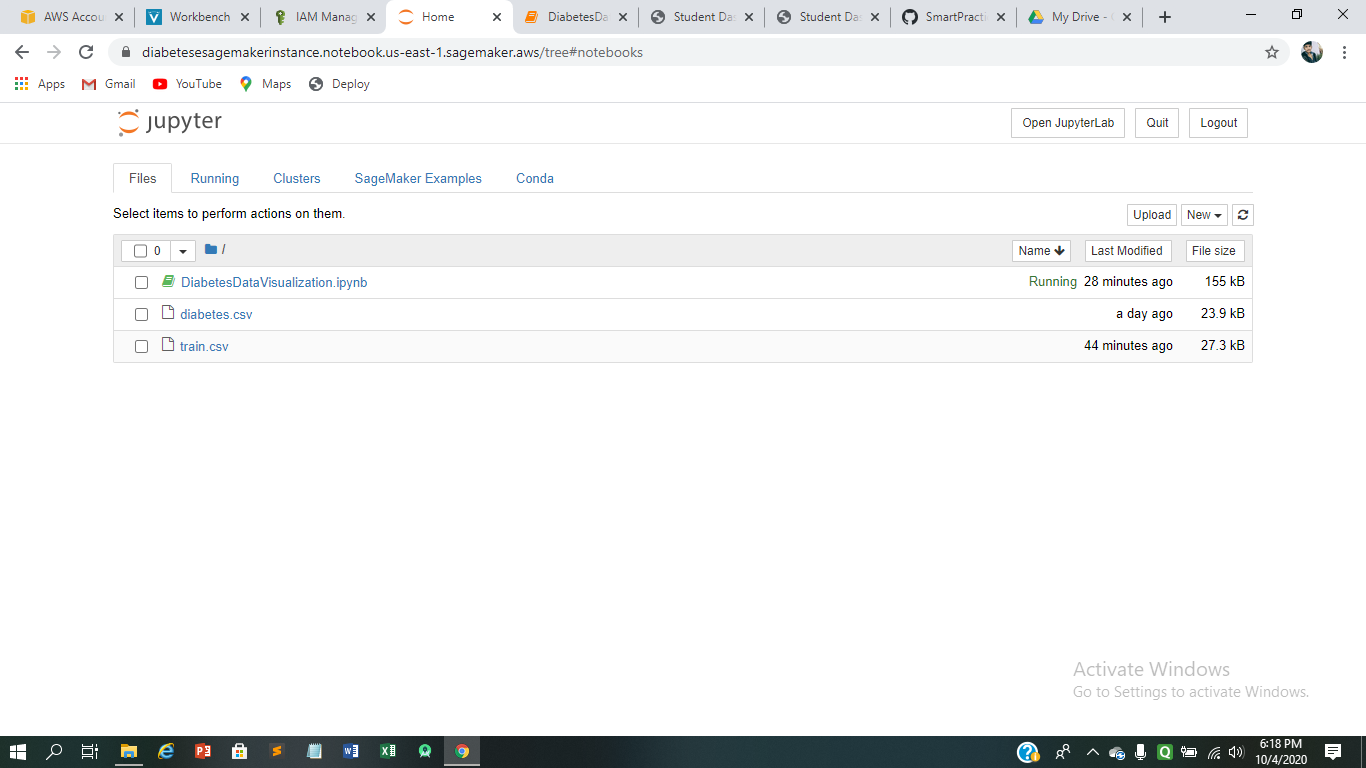
**3. AWS API Gateway**

**4. AWS Lambda**

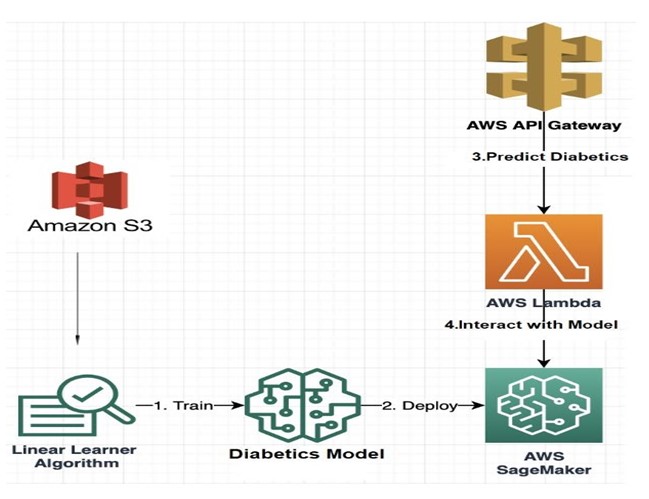
**5. Amazon Sagemaker**

**Experimental Investigations:**

Develop an end-to-end web application that predicts the probability of females having diabetes. The application must be built with Python-Flask or Django framework with the machine learning model trained & deployed on AWS Sagemaker. Create an API Endpoint for the model with the help of API Gateway and AWS Lambda Service.



## **Flowchart**



***Fig. 2.****Flowchart*

**Result:**

Develop an end-to-end web application that predicts the probability of females having diabetes. The application must be built with Python-Flask or Django framework with the machine learning model trained & deployed on AWS Sagemaker. Create an API Endpoint for the model with the help of API Gateway and AWS Lambda Service.

**Advantages & Disadvantages:**

**Advantage:**

* It helps to detect the diabetics in the earlier stages
* It helps to easily detects the diabetic

**Disadvantage:**

* maybe inaccurate sometimes
* For prediction result needs internet connectivity

**Applications:**

* Used in early prediction for diabetes for women.

* We can also use it for predicting heart disease, phenomena, kidney disease
* We can also use it to predict the medical health condition of the people.

**Conclusion**

Machine learning has the great ability to revolutionize the diabetes risk prediction with the help of advanced computational methods and availability of large amount of epidemiological and genetic diabetes risk dataset. Detection of diabetes in its early stages is the key for treatment. This work has described a machine learning approach to predicting diabetes levels. The technique may also help researchers to develop an accurate and effective tool that will reach at the table of clinicians to help them make better decision about the disease status.

**Future Scope:**

This research study has only targeted parents with diabetes. Readmission prediction model needs to be generated for other key health conditions also such as Heart disease, Phenomena, kidney disease etc. in Indian Healthcare system. In the future studies, planned and unplanned(emergency) readmission needs to be considered.

**Bibliography:**

### This video is a sample from Skillsoft's video course catalog. After watching this video, you will be able to get data into and out of an S3 bucket.

### <https://www.youtube.com/watch?v=VDdz1H18-0g>

### Data visualization using Python

<https://towardsdatascience.com/data-visualization-for-machine-learning-and-data-science-a45178970be7>

### how to build, train, and deploy a machine learning model with Amazon SageMaker? Learn how to build, train, and deploy a machine learning model with Amazon SageMaker.

<https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker/>

### Build Model By Applying Training Data To Estimators And Deploy The Model

<https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-> learning-model-sagemaker/

### You can run Python code in AWS Lambda. Lambda provides runtimes for Python that run your code to process events. Your code runs in an environment that includes the SDK for Python (Boto3), with credentials from an AWS Identity and Access Management (IAM) role that you manage.

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-python.html>

### How to use AWS Lambda with Amazon API Gateway.

<https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway.html>

**Appendix:**

**Source Code:**

**Diabetes data visualization in Jupyter notebook:**



**Lambda Function :**

import os

import io

import boto3

import json

import csv

# grab environment variables

ENDPOINT\_NAME = os.environ['ENDPOINT\_NAME']

runtime= boto3.client('runtime.sagemaker')

def lambda\_handler(event, context):

print("Received event: " + json.dumps(event, indent=2))

data = json.loads(json.dumps(event))

payload = data['data']

print(payload)

response = runtime.invoke\_endpoint(EndpointName=ENDPOINT\_NAME,

ContentType='text/csv',

Body=payload)

print(response)

result = json.loads(response['Body'].read().decode())

print("Result is:"+str(result))

pred = round(result)

print("Pred is:"+str(pred))

predicted\_label = (str(pred)+' Diabetic Person') if pred == 1 else (str(pred)+'Healthy Person')

return predicted\_label