PROJECT: DIABETICS MELLITUS CLASSIFICATION ANALYSIS USING AWS SAGE MAKER AND HOSTING THE JUPYTER NOTEBOOK AS A STATIC WEBPAGE.

Overview

In this presentation, we will guide you through the process of training a Support Vector Machine (SVM) model for Diabetes Mellitus classification on Amazon Web Services (AWS) using SageMaker and hosting the Jupyter Notebook in HTML format as a static webpage. The steps include creating an AWS account, launching a SageMaker Notebook Instance, deploying the trained model, and hosting the static webpage on S3.

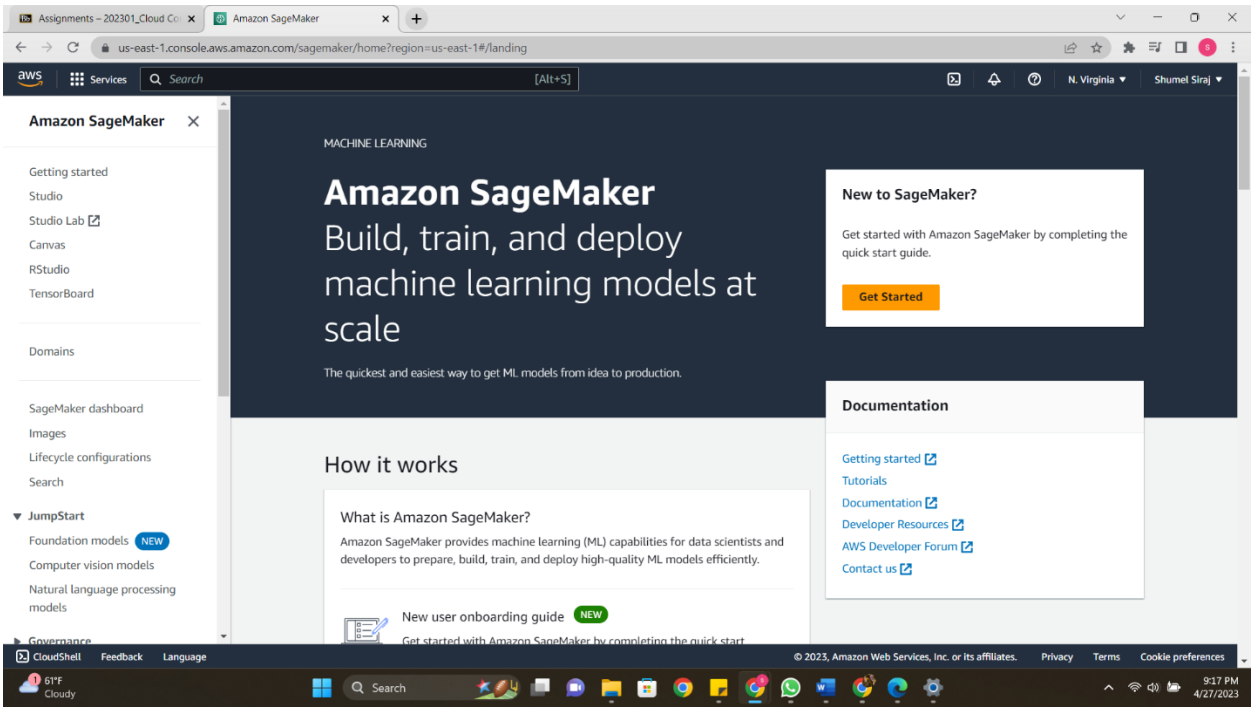
- **Steps to Train a Diabetes Mellitus Classification Model on AWS SageMaker**

1. Create an AWS Account and Sign into the Management Console

   - Create an AWS account by visiting https://aws.amazon.com/ and clicking "Create an AWS Account".
   - Follow the prompts, supplying your name, email address, and payment information.
   - Sign into the AWS Management Console at https://console.aws.amazon.com/ using your AWS account credentials.

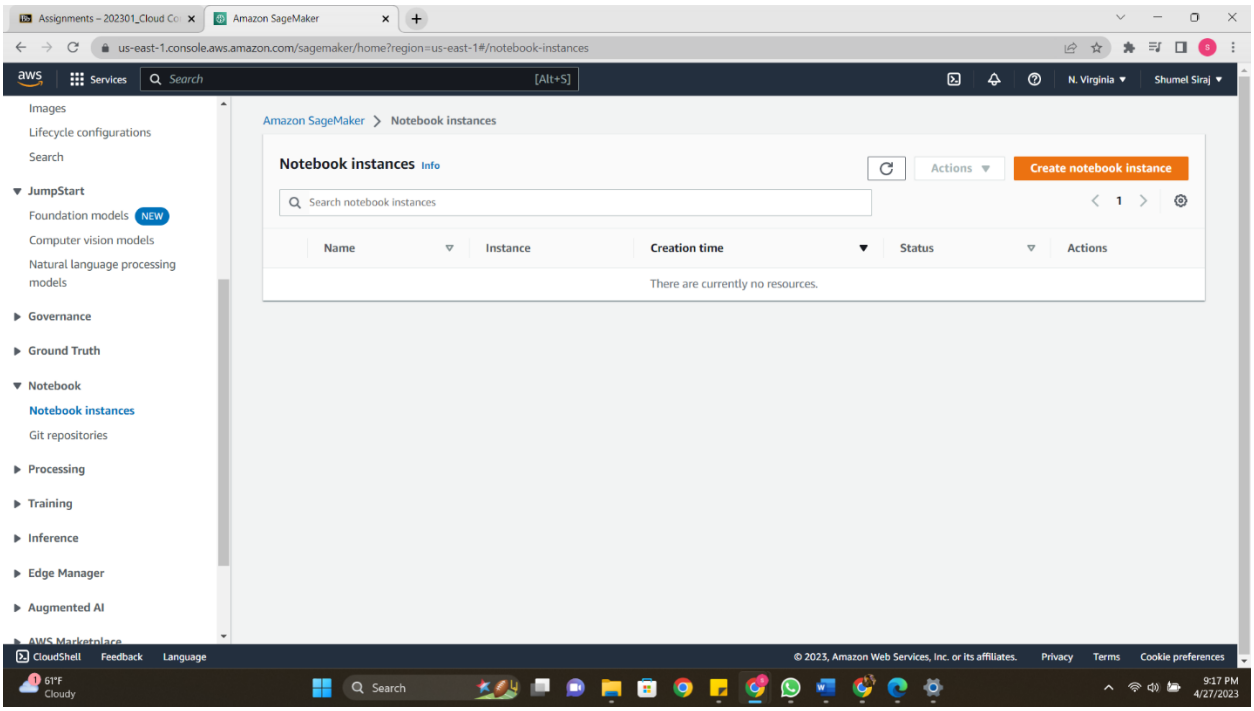
2. Navigate to the Amazon SageMaker Console

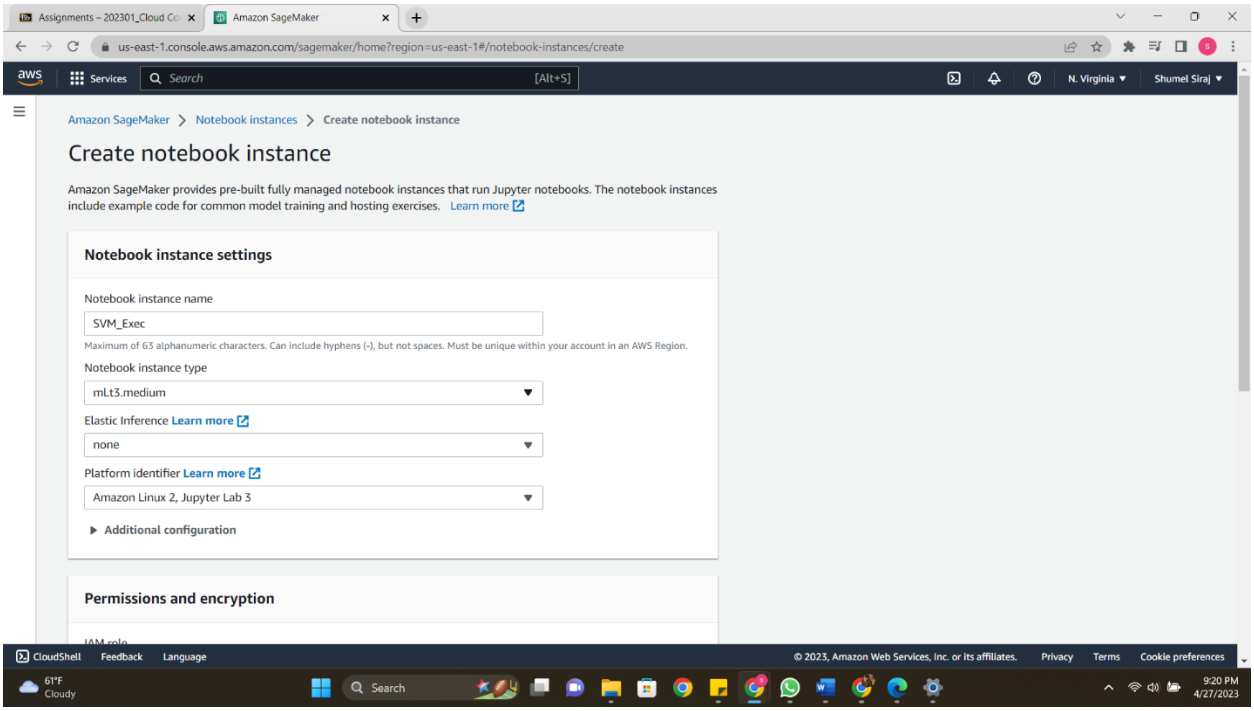
   - Search for "SageMaker" in the search bar at the top of the AWS Management Console.
   - Click on "Amazon SageMaker" in the search results.

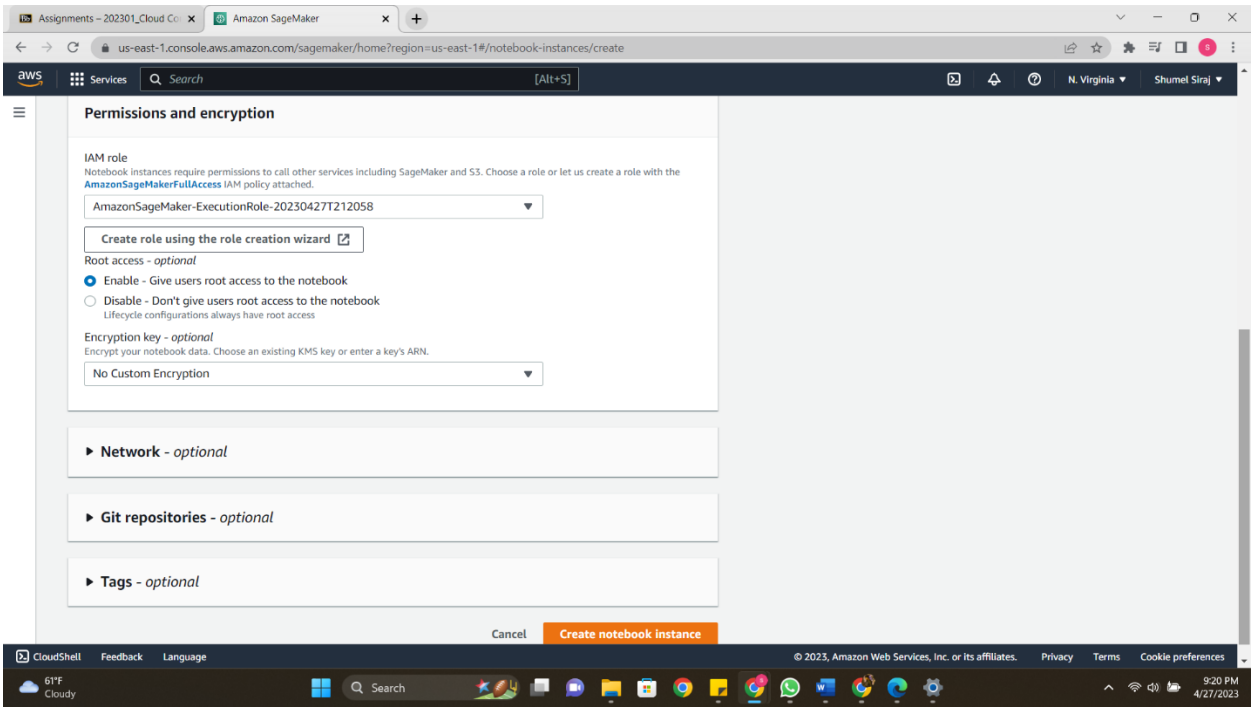


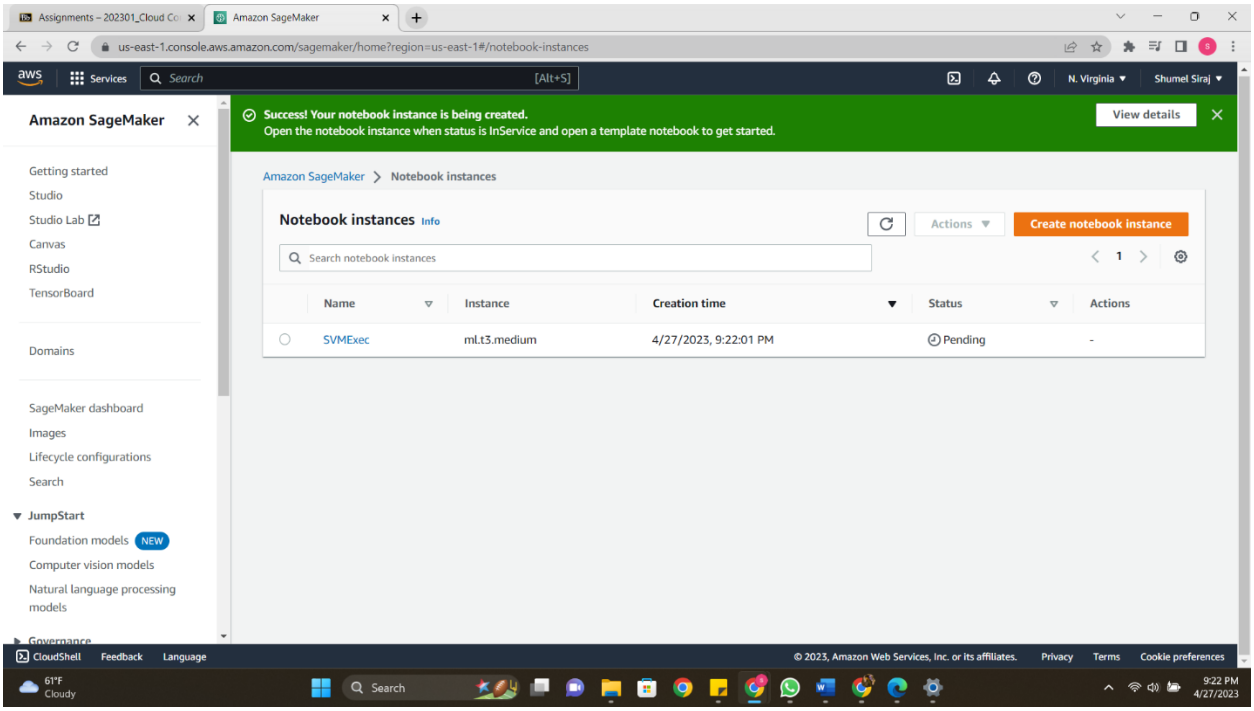
3. Launch an Amazon SageMaker Notebook Instance

- Click on "Notebook instances" in the SageMaker console.
- Click on the "Create notebook instance" button.
- Provide a name for your instance, select an instance type, and choose the IAM role with necessary permissions.
- Click on "Create notebook instance".


Assignments – 202301_Cloud Co
Amazon SageMaker
us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/notebook-instances
Services
Search
[Alt+S]
N. Virginia
Shumel Siraj
Images
Lifecycle configurations
Search
JumpStart
Foundation models
NEW
Computer vision models
Natural language processing models
Governance
Ground Truth
Notebook
Notebook instances
Git repositories
Processing
Training
Inference
Edge Manager
Augmented AI
Amazon SageMaker > Notebook instances
Notebook instances Info
Actions
Create notebook instance
Search notebook instances
Name
Instance
Creation time
Status
Actions
There are currently no resources.
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
61°F
Cloudy
Search
9:17 PM
4/27/2023

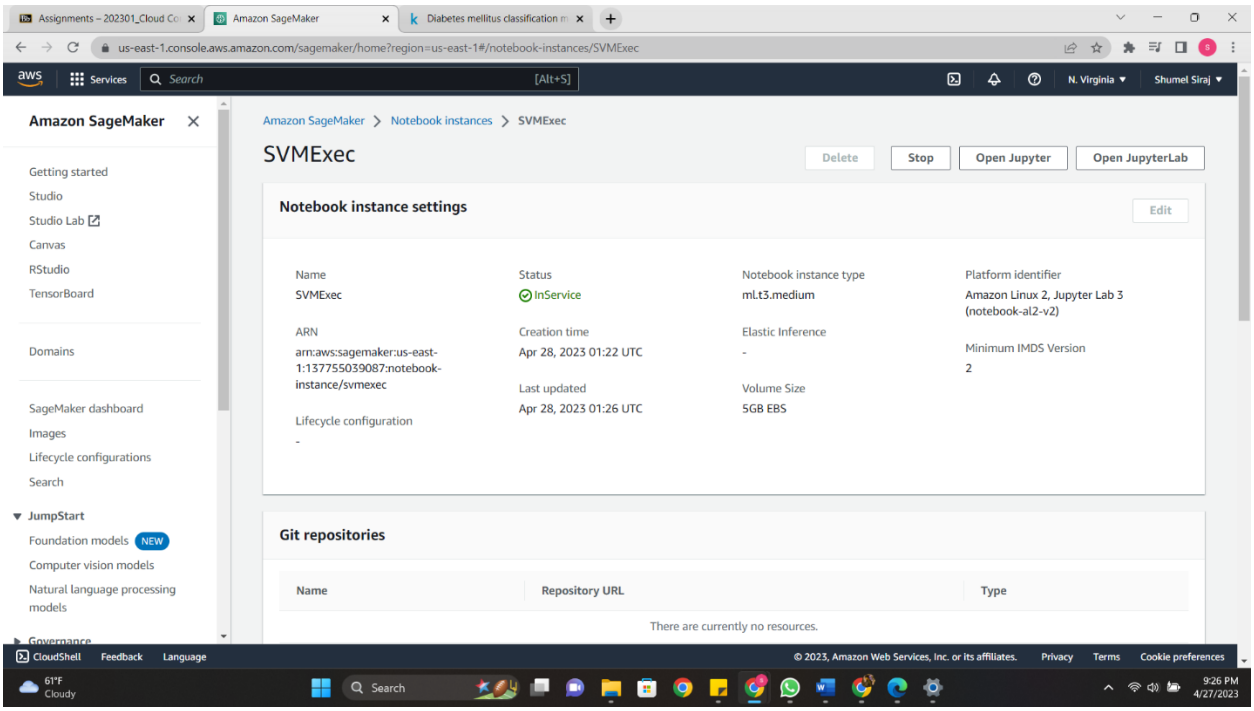

Assignments – 202301_Cloud Co
Amazon SageMaker
us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/notebook-instances/create
Services
Search
[Alt+S]
N. Virginia
Shumel Siraj
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. Learn more
Notebook instance settings
Notebook instance name
SVM_Exec
Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Notebook instance type
ml.t3.medium
Elastic Inference Learn more
none
Platform identifier Learn more
Amazon Linux 2, Jupyter Lab 3
Additional configuration
Permissions and encryption
IAM role
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
61°F
Cloudy
Search
9:20 PM
4/27/2023





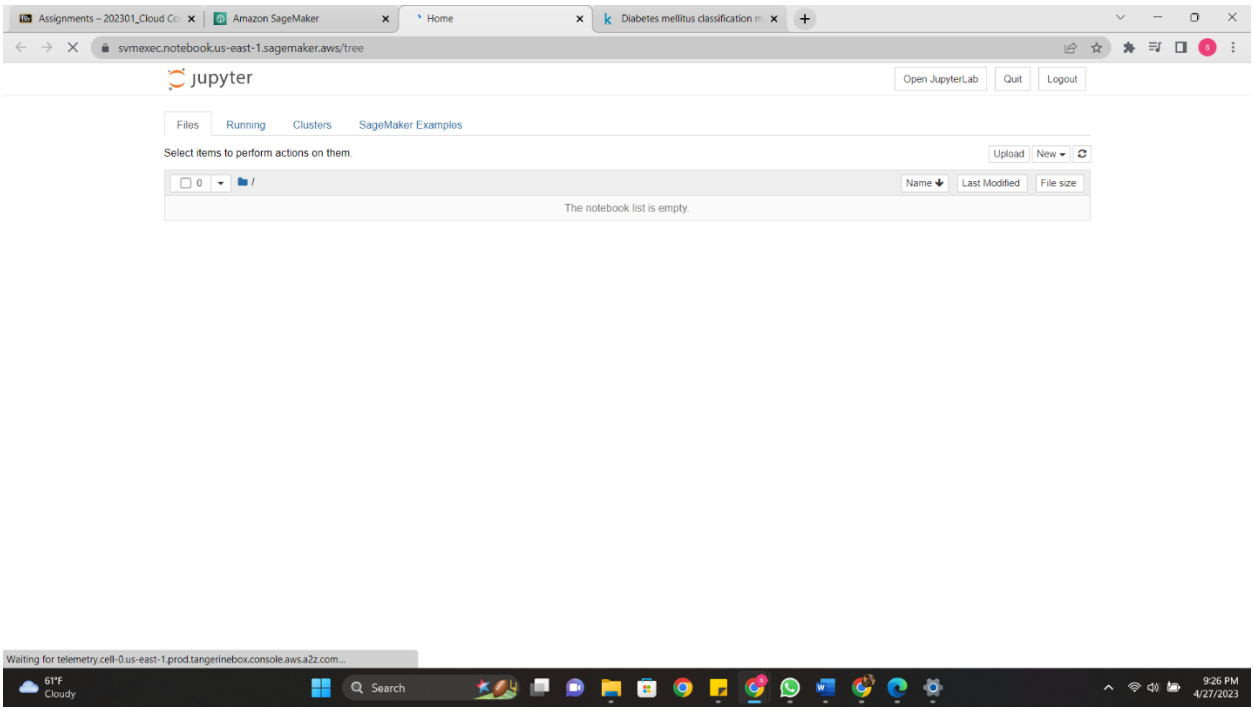
4. Open Jupyter Notebook

- Once the Notebook Instance has started, click on "Open JupyterLab".
- JupyterLab will open in a new browser tab.

- 

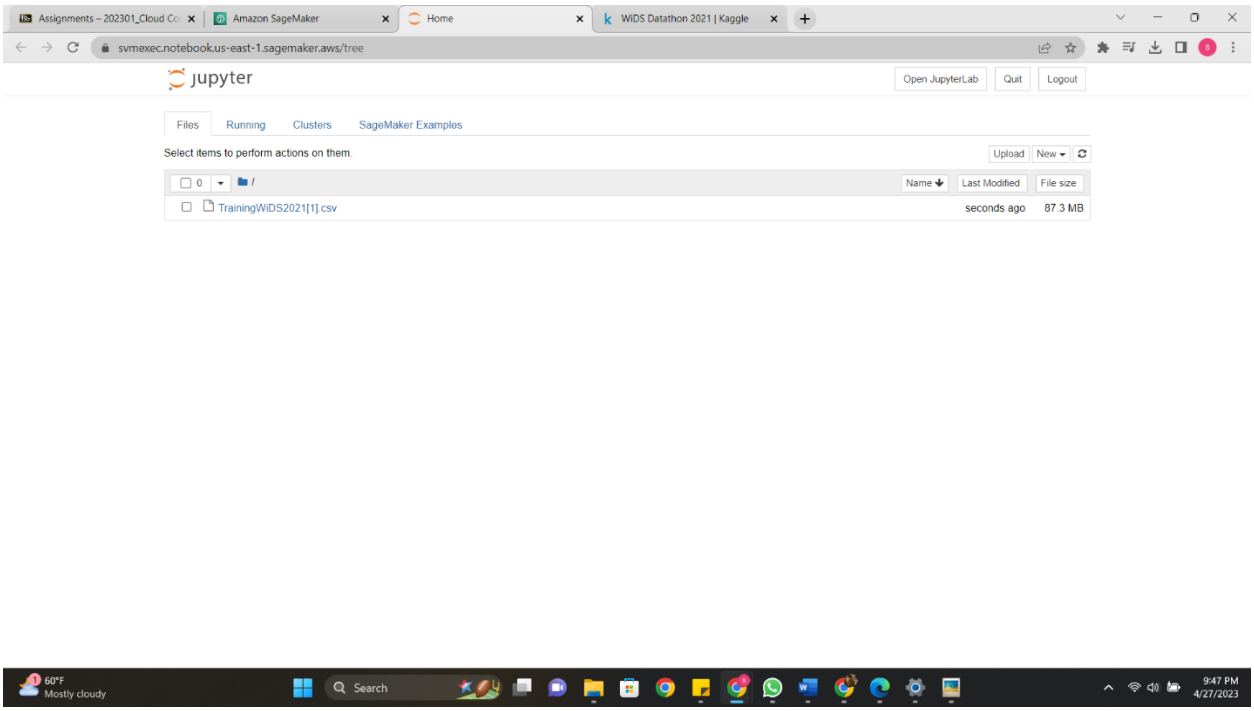
5. Create a New Notebook

- Click on the "+" icon in the left sidebar of JupyterLab.
- Choose the Python version and select the desired kernel.



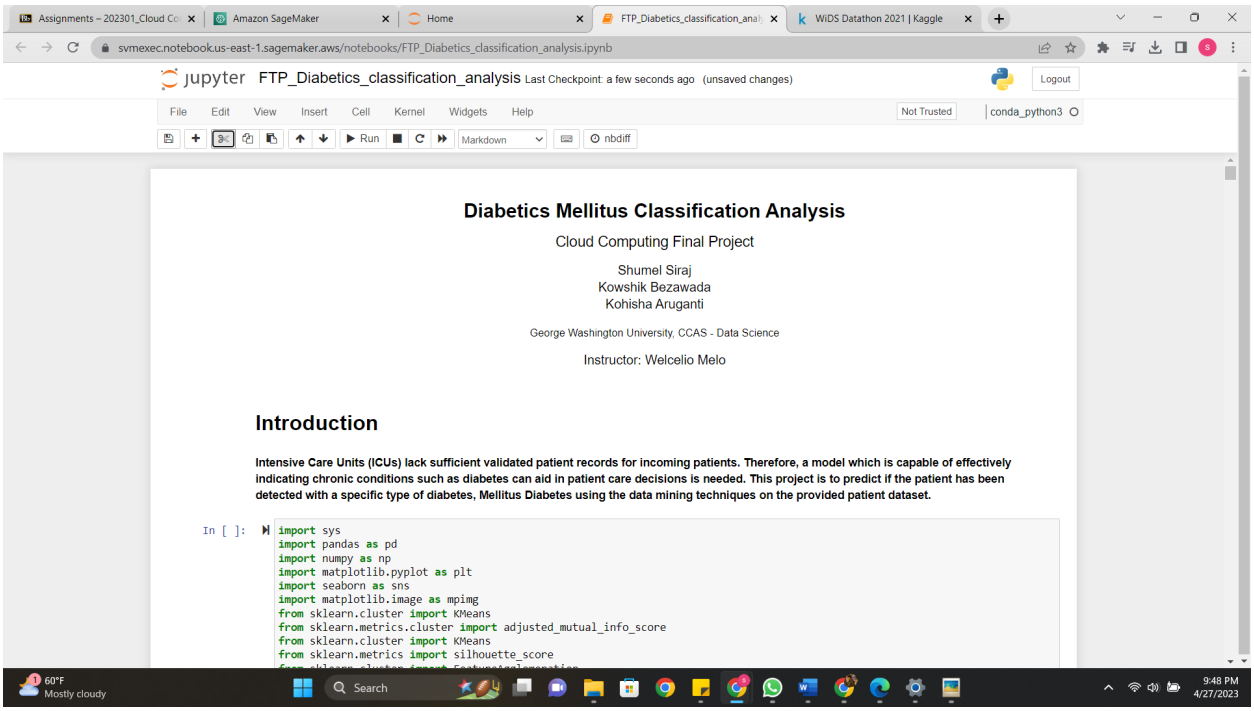
6. Load Data

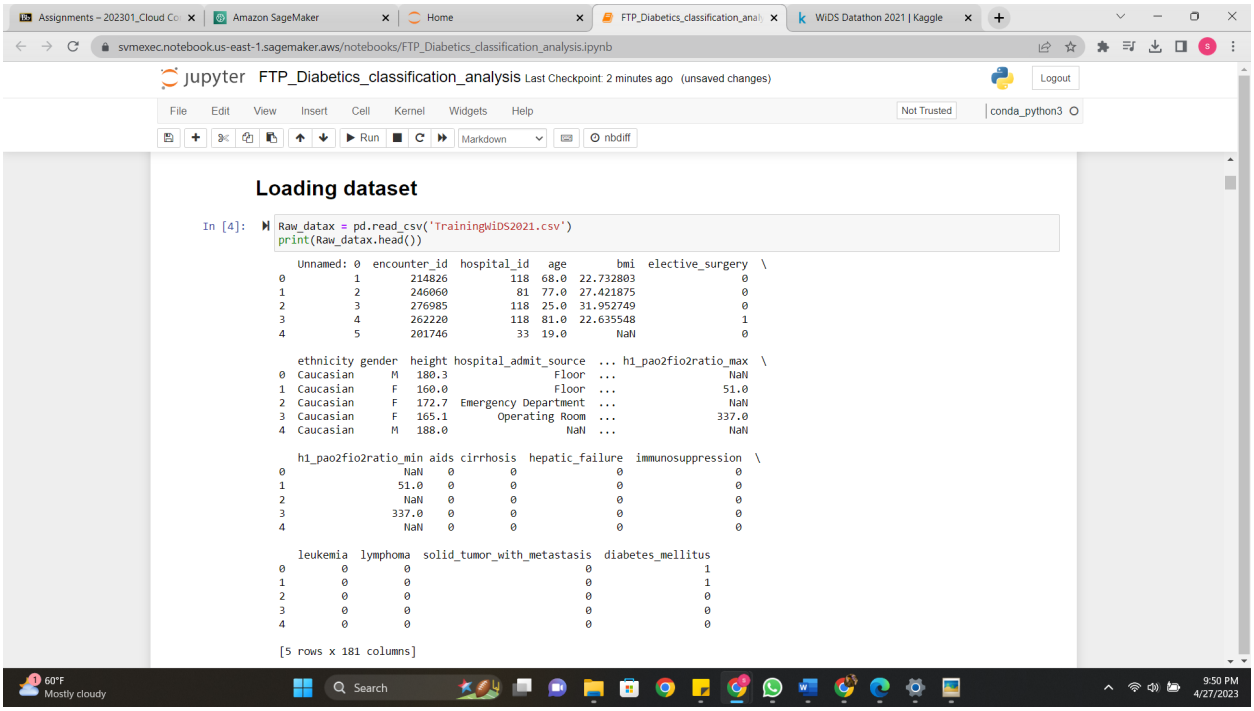
- Upload your Diabetes Mellitus dataset to the notebook instance using the "Upload files" button in the JupyterLab file explorer.

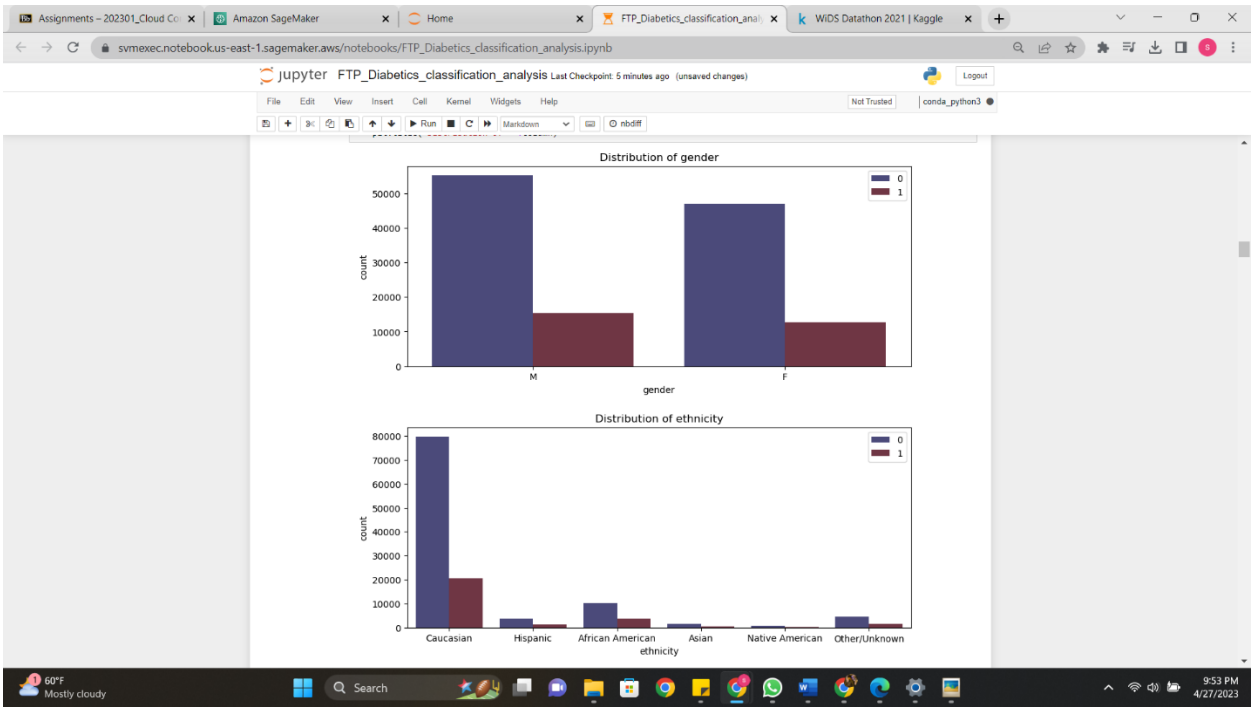


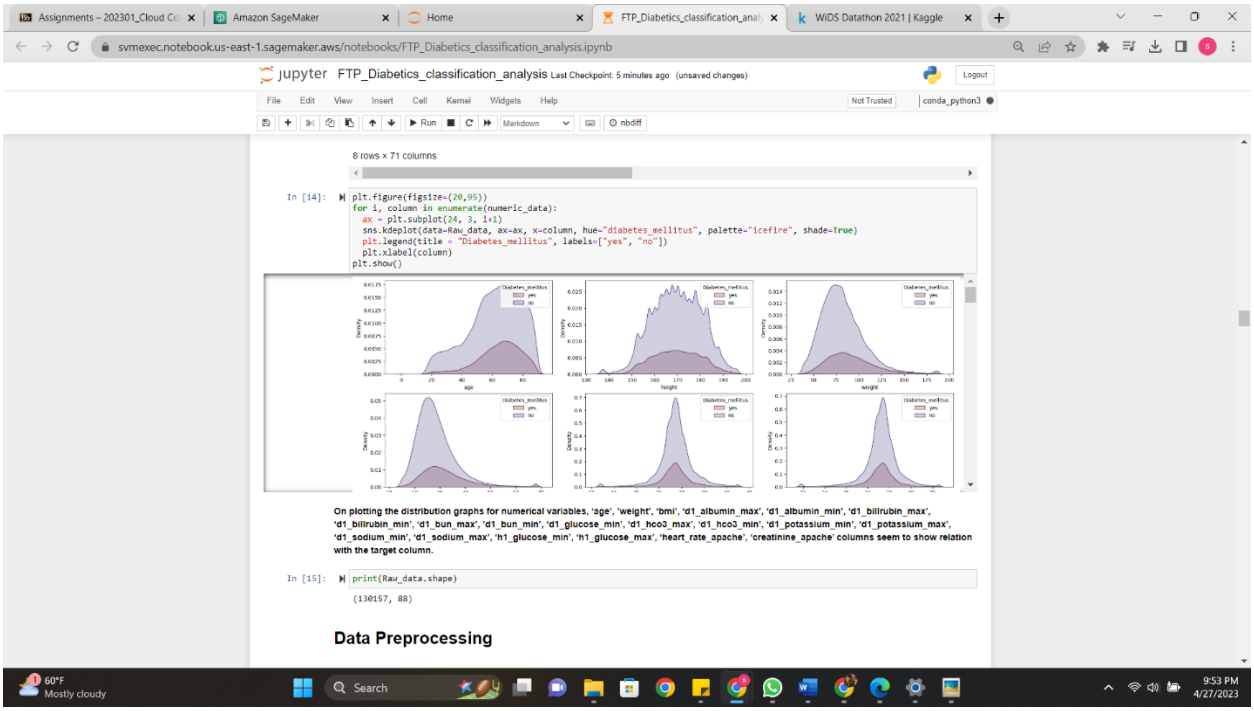
7. Prepare the Data

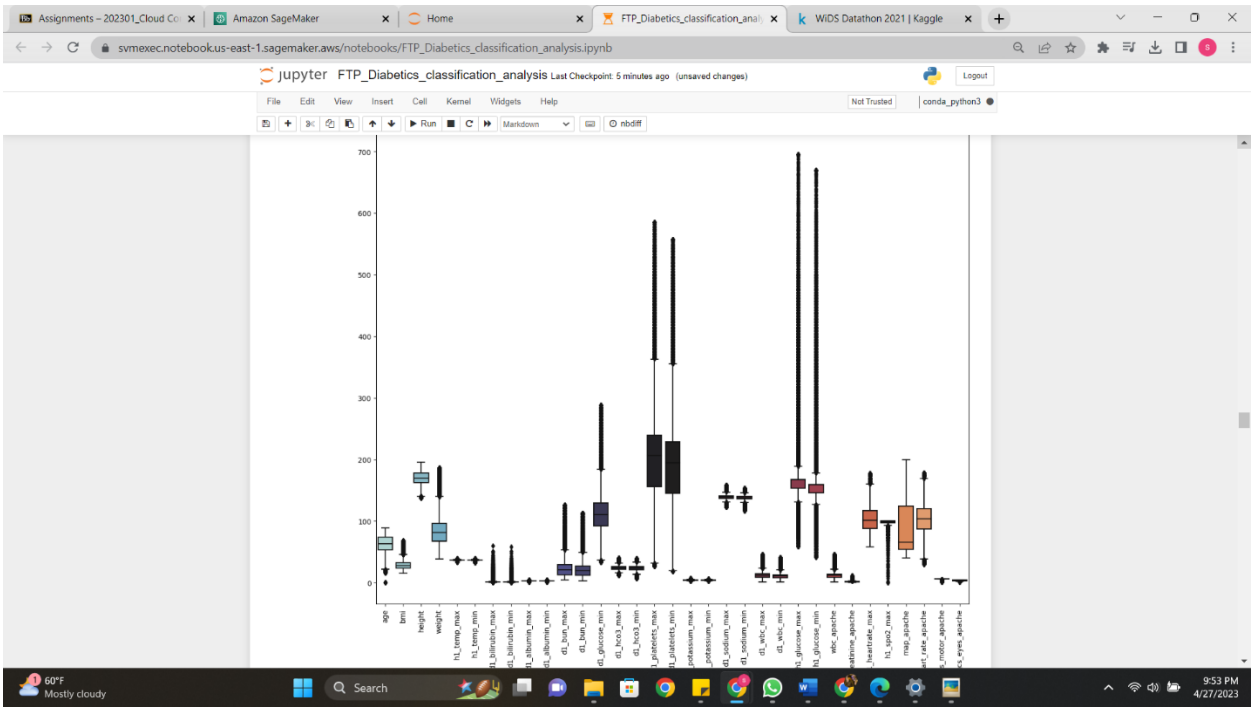
- Import necessary libraries for preprocessing, such as pandas and NumPy.
- Clean, transform, and normalize the Diabetes Mellitus dataset as required by the SVM model.


Assignments – 202301_Cloud Co
Amazon SageMaker
Home
FTP_Diabetics_classification_anal
WiDS Datathon 2021 | Kaggle
svmexec.notebook.us-east-1.sagemaker.aws/notebooks/FTP_Diabetics_classification_analysis.ipynb
jupyter FTP_Diabetics_classification_analysis Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
conda_python3
Run
Markdown
nbdiff
Diabetics Mellitus Classification Analysis
Cloud Computing Final Project
Shumel Siraj
Kowshik Bezawada
Kohisha Aruganti
George Washington University, CCAS - Data Science
Instructor: Welcelio Melo
Introduction
Intensive Care Units (ICUs) lack sufficient validated patient records for incoming patients. Therefore, a model which is capable of effectively indicating chronic conditions such as diabetes can aid in patient care decisions is needed. This project is to predict if the patient has been detected with a specific type of diabetes, Mellitus Diabetes using the data mining techniques on the provided patient dataset.
In [ ]:
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.image as mpimg
from sklearn.cluster import KMeans
from sklearn.metrics.cluster import adjusted_mutual_info_score
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
60°F
Mostly cloudy
Search
9:48 PM
4/27/2023


Assignments – 202301_Cloud Co
Amazon SageMaker
Home
FTP_Diabetics_classification_anal
WiDS Datathon 2021 | Kaggle
svmexec.notebook.us-east-1.sagemaker.aws/notebooks/FTP_Diabetics_classification_analysis.ipynb
jupyter FTP_Diabetics_classification_analysis Last Checkpoint: 2 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
conda_python3
Run
Markdown
nbdiff
Loading dataset
In [4]:
Raw_datax = pd.read_csv('TrainingWiDS2021.csv')
print(Raw_datax.head())
Unnamed: 0 encounter_id hospital_id age bmi elective_surgery \
0 1 214826 118 68.0 22.732803 0
1 2 246060 81 77.0 27.421875 0
2 3 276985 118 25.0 31.952749 0
3 4 262220 118 81.0 22.635548 1
4 5 201746 33 19.0 NaN 0
ethnicity gender height hospital_admit_source ... h1_pao2fio2ratio_max \
0 Caucasian M 180.3 Floor ... NaN
1 Caucasian F 160.0 Floor ... 51.0
2 Caucasian F 172.7 Emergency Department ... NaN
3 Caucasian F 165.1 Operating Room ... 337.0
4 Caucasian M 188.0 NaN ... NaN
h1_pao2fio2ratio_min aids cirrhosis hepatic_failure immunosuppression \
0 NaN 0 0 0 0
1 51.0 0 0 0 0
2 NaN 0 0 0 0
3 337.0 0 0 0 0
4 NaN 0 0 0 0
leukemia lymphoma solid_tumor_with_metastasis diabetes_mellitus
0 0 0 0 1
1 0 0 0 1
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
[5 rows x 181 columns]
60°F
Mostly cloudy
Search
9:50 PM
4/27/2023

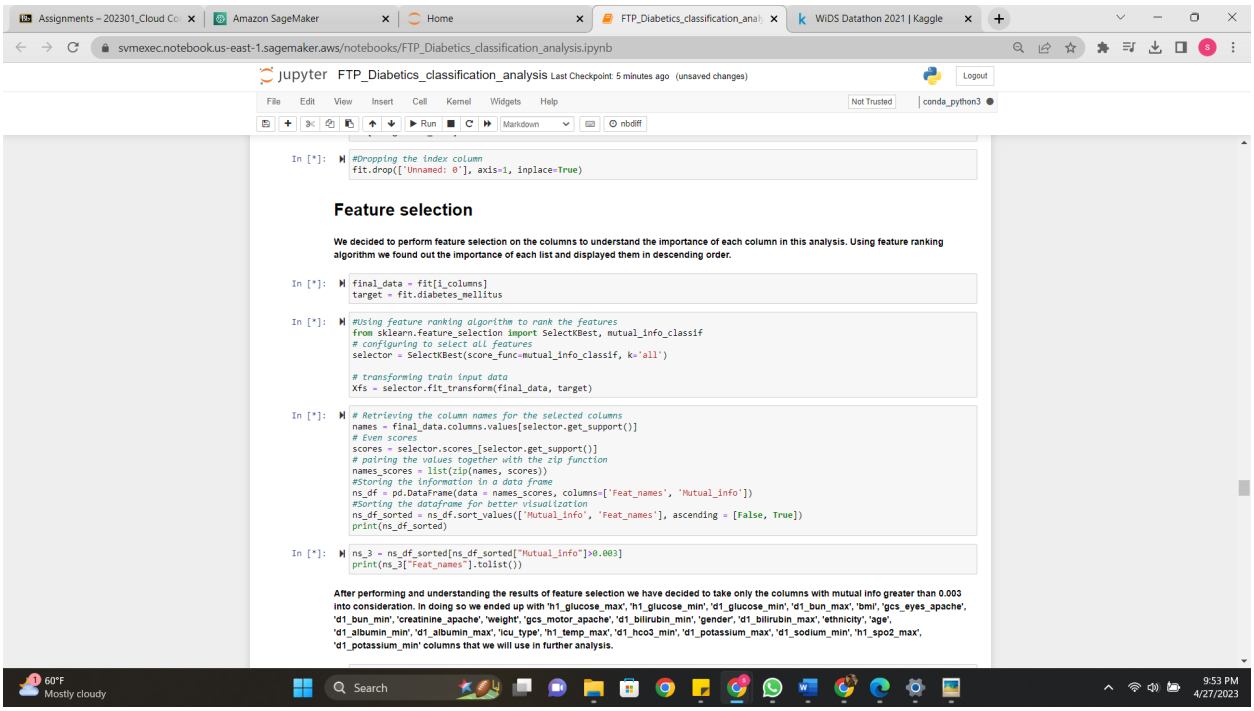

Distribution of gender
count
gender
M
F
0
1
Distribution of ethnicity
count
ethnicity
Caucasian
Hispanic
African American
Asian
Native American
Other/Unknown


8 rows × 71 columns
In [14]:
plt.figure(figsize=(20,95))
for i, column in enumerate(numeric_data):
ax = plt.subplot(24, 3, i+1)
sns.kdeplot(data=Raw_data, ax=ax, x=column, hue="diabetes_mellitus", palette="icefire", shade=True)
plt.legend(title = "Diabetes_mellitus", labels=["yes", "no"])
plt.xlabel(column)
plt.show()
On plotting the distribution graphs for numerical variables, 'age', 'weight', 'bmi', 'd1_albumin_max', 'd1_albumin_min', 'd1_bilirubin_max', 'd1_bilirubin_min', 'd1_bun_max', 'd1_bun_min', 'd1_glucose_min', 'd1_hco3_max', 'd1_hco3_min', 'd1_potassium_min', 'd1_potassium_max', 'd1_sodium_min', 'd1_sodium_max', 'h1_glucose_min', 'h1_glucose_max', 'heart_rate_apache', 'creatinine_apache' columns seem to show relation with the target column.
In [15]:
print(Raw_data.shape)
(130157, 88)
Data Preprocessing



8. Train the Diabetes Mellitus Classification Model

- Import necessary libraries for training, such as scikit-learn.
- Split the Diabetes Mellitus dataset into training and validation sets.
- Train the SVM model using the training data.
- Evaluate the model performance using the validation data.



9. Deploy the Model

- Save the trained Diabetes Mellitus classification model as a file or upload it to an Amazon S3 bucket.
- Deploy the model as a web service using Sage Maker's built-in deployment functionality.

10. Test the Model

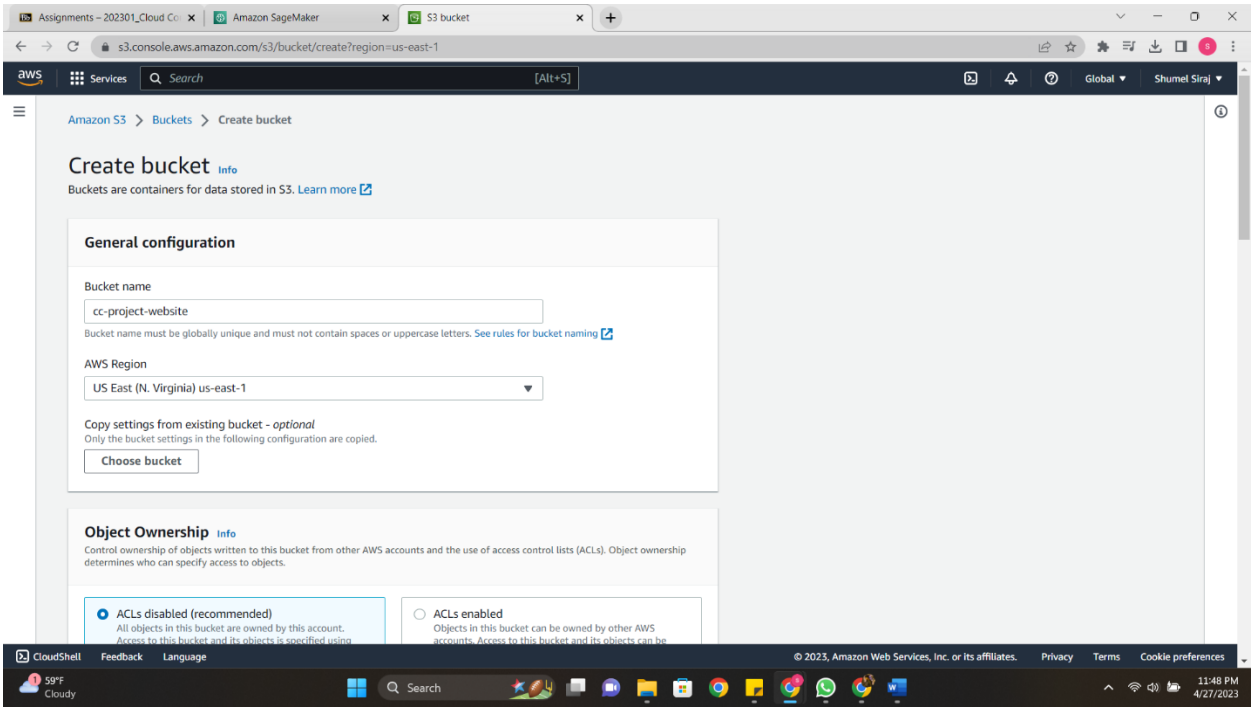
- Evaluate the deployed Diabetes Mellitus classification model using a test dataset to ensure it performs as expected.

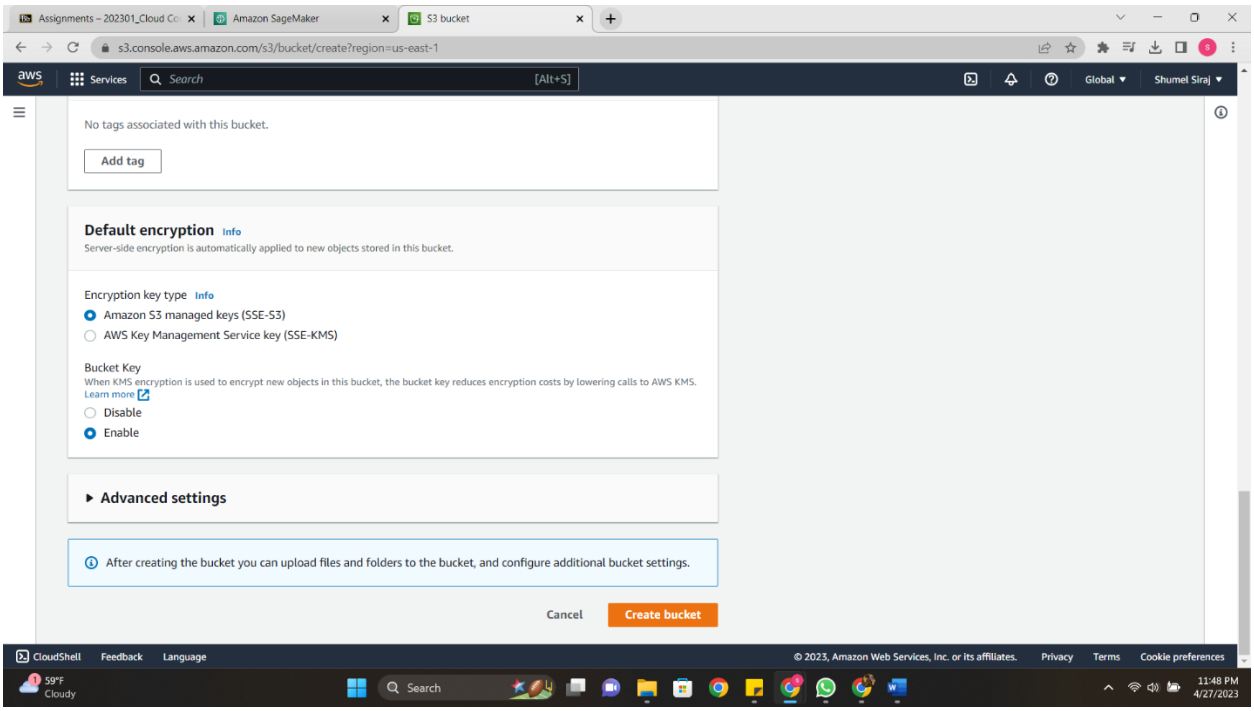
➢ **Steps to Host the Jupyter Notebook as a Static Webpage**

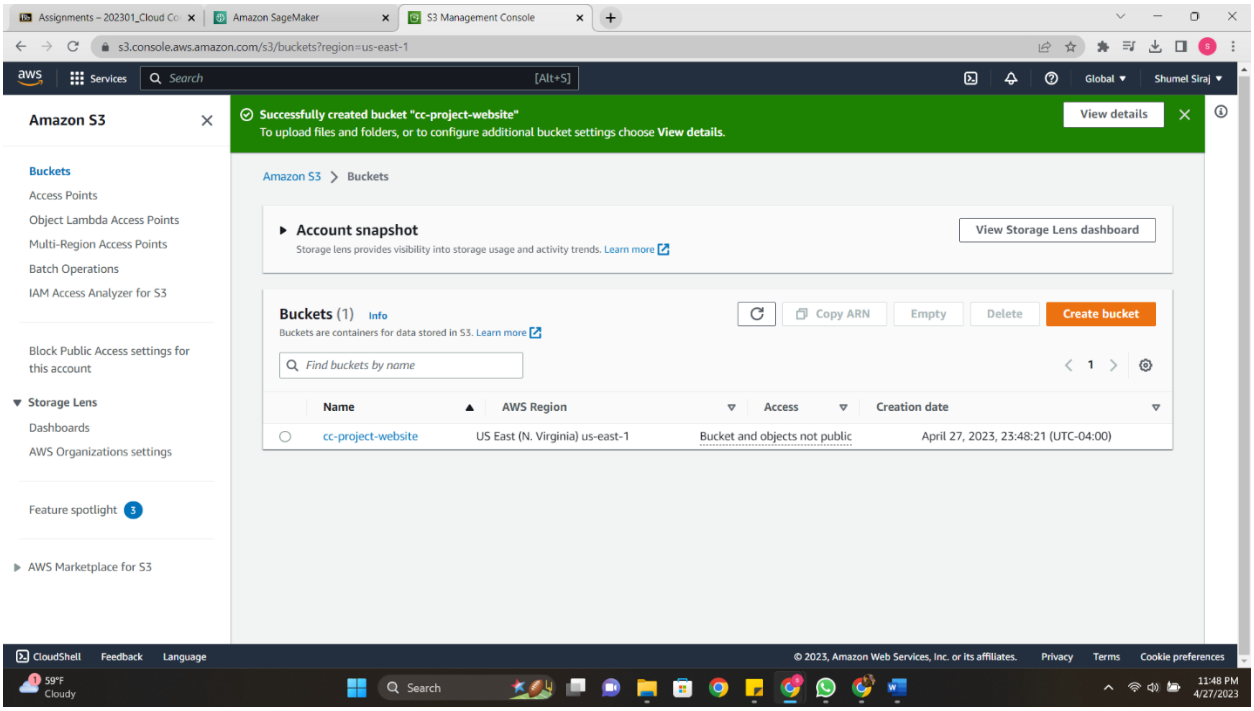
11. Create an S3 Bucket

- Log in to the AWS Management Console and navigate to the S3 service.
- Click the "Create bucket" button to create a new S3 bucket.

- Enter a unique name for your bucket, choose the region where you want to host your bucket, and click "Create".

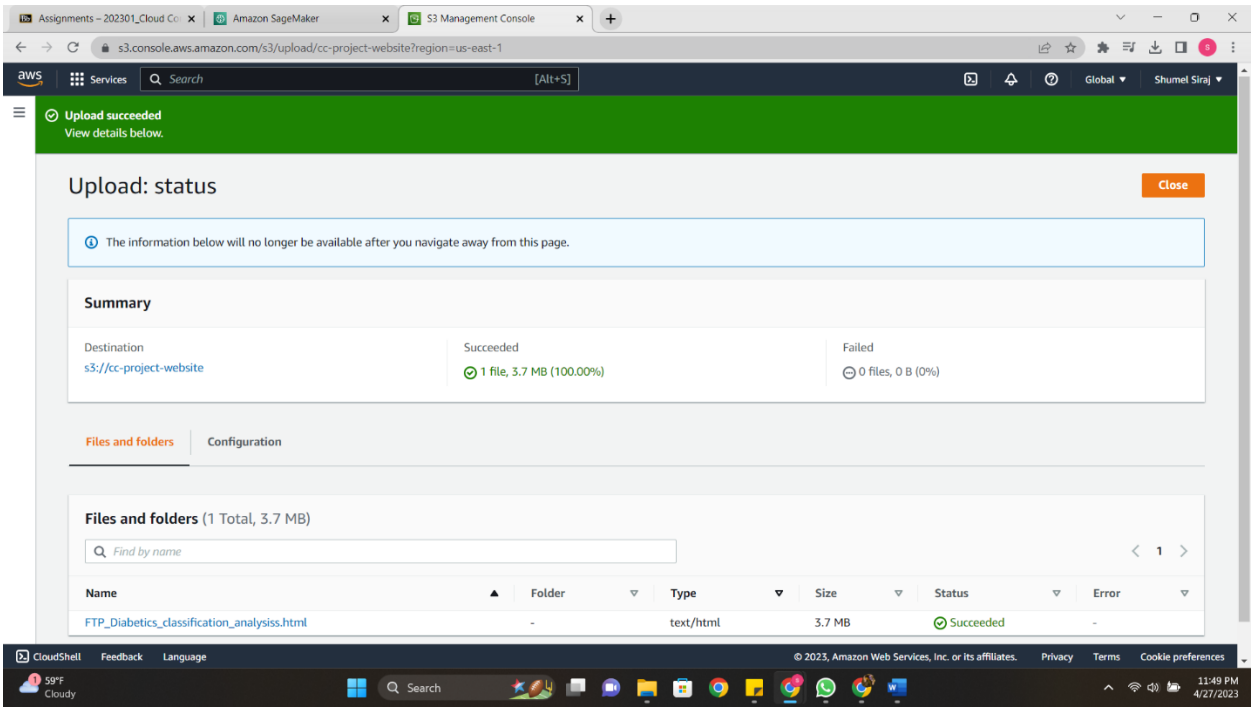






12. Upload Your Webpage

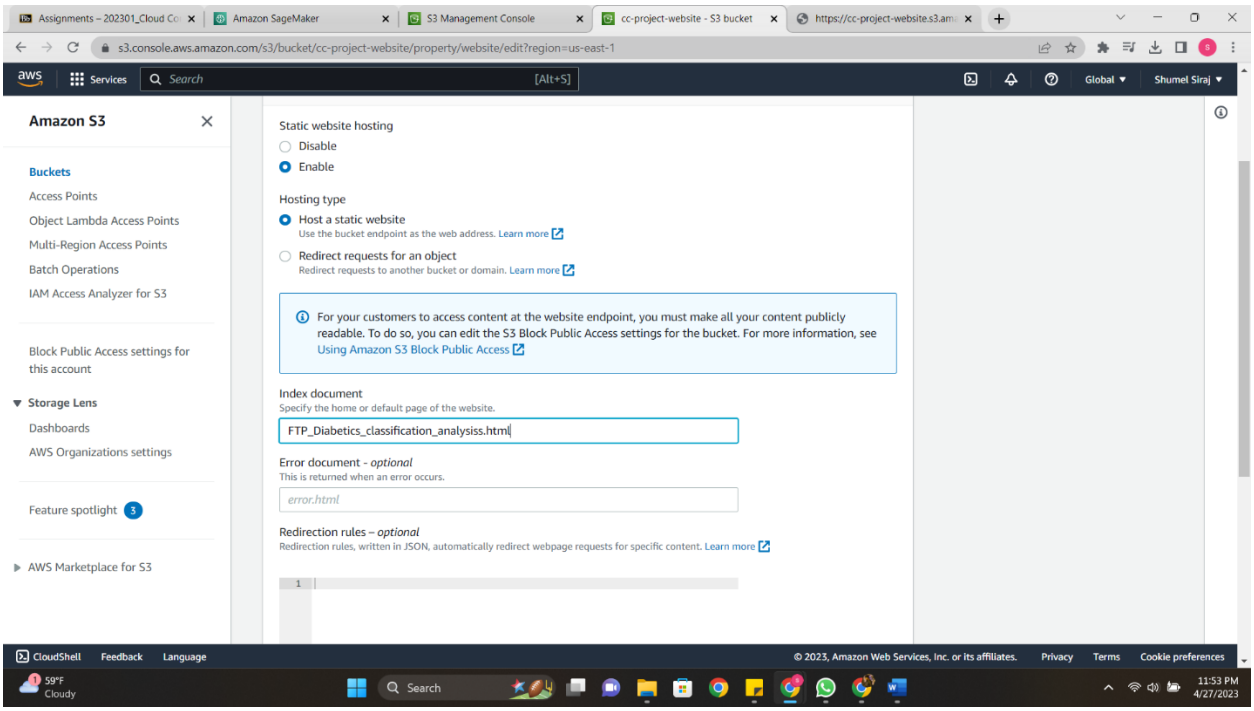
- Upload your HTML, CSS, and JavaScript files to the S3 bucket using the S3 console or command line interface.



Make sure your files are organized and linked correctly.

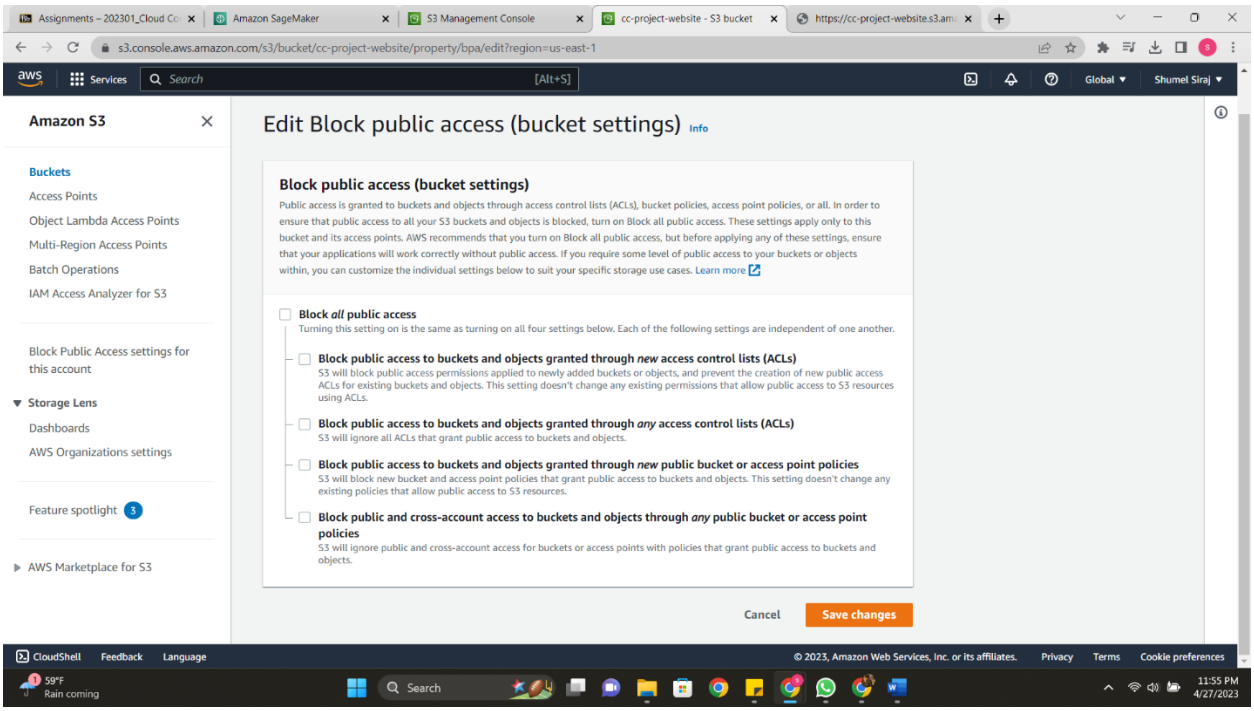
13. Configure Bucket Properties

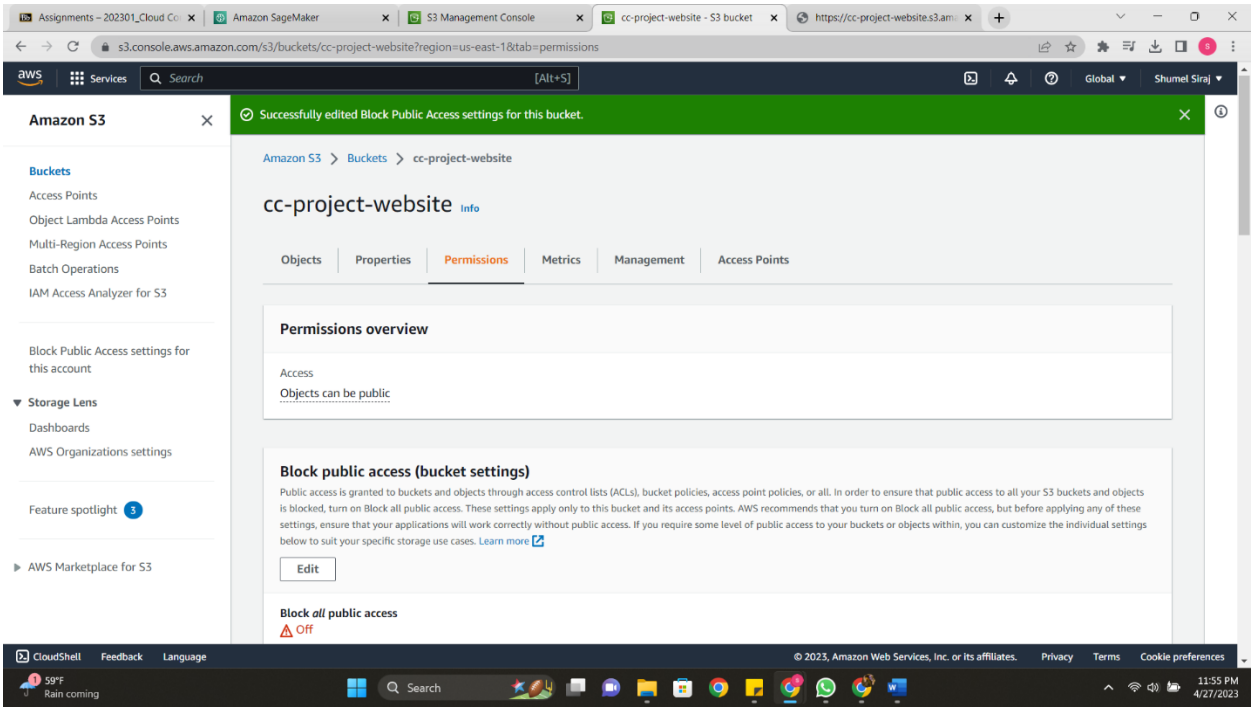
- In the S3 console, select your bucket and click the "Properties" tab.
- Click on "Static website hosting".
- Select "Use this bucket to host a website".
- Enter the name of your index document (usually "index.html").
- Click "Save".

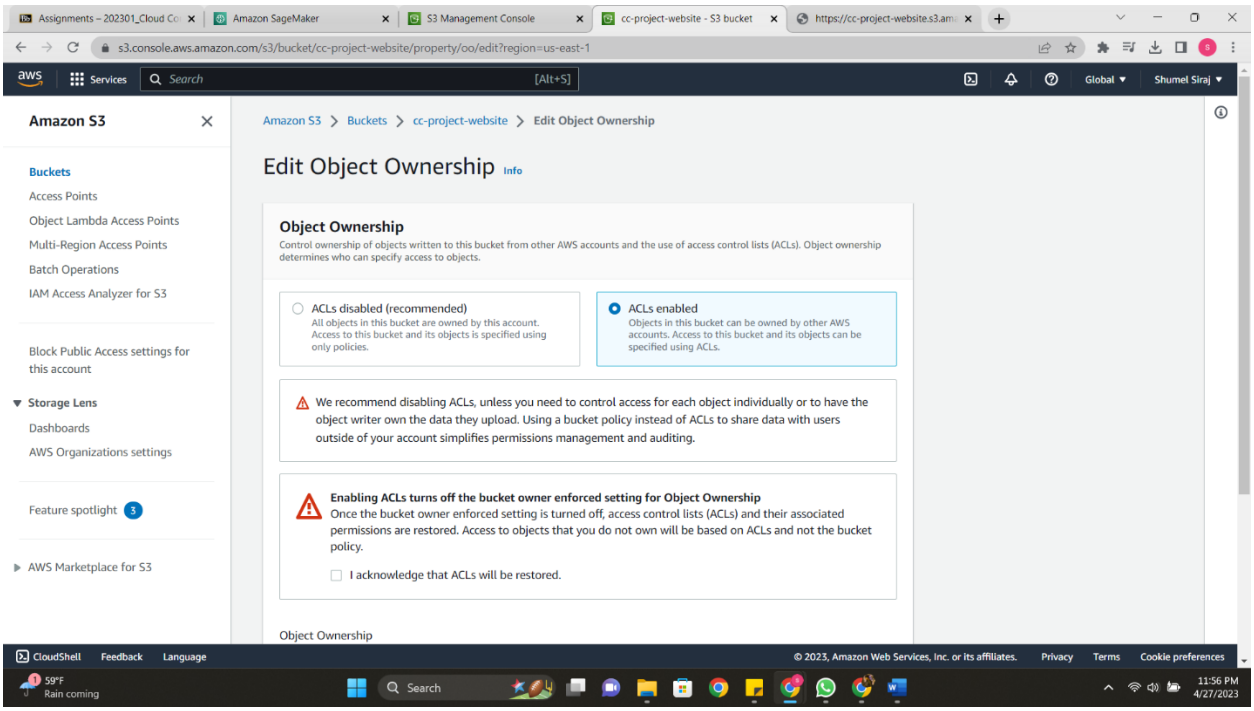


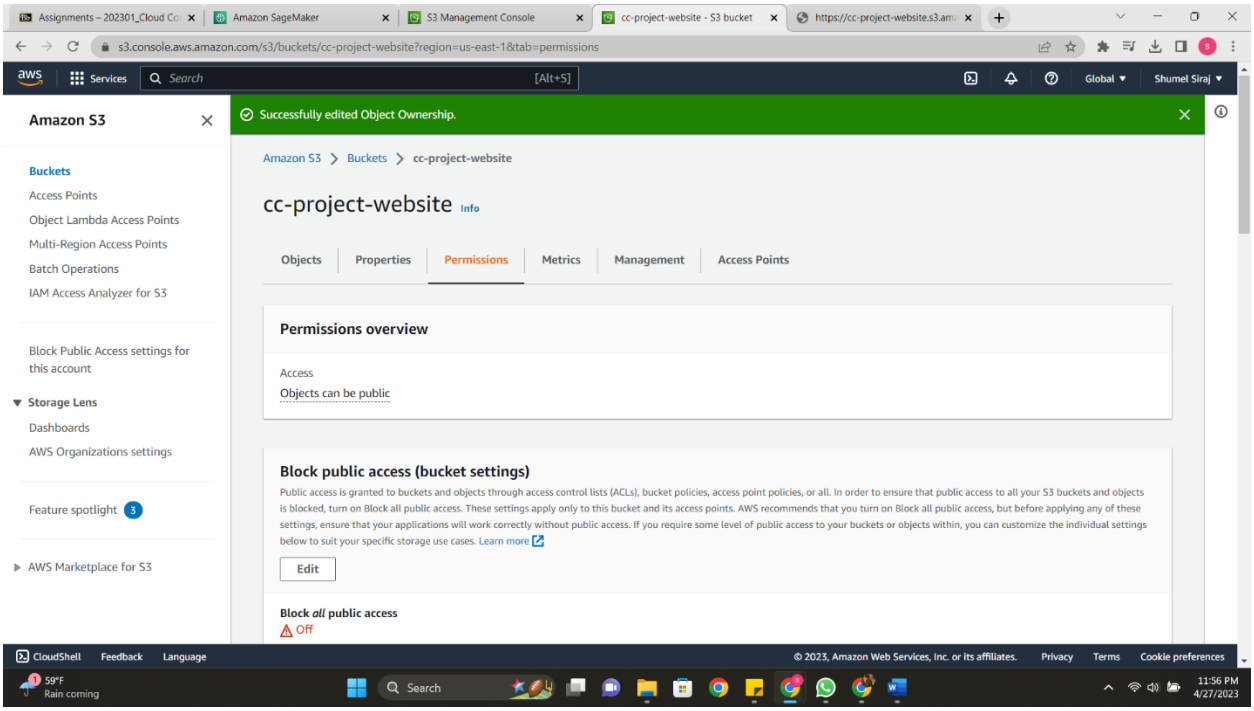
14. Set Permissions

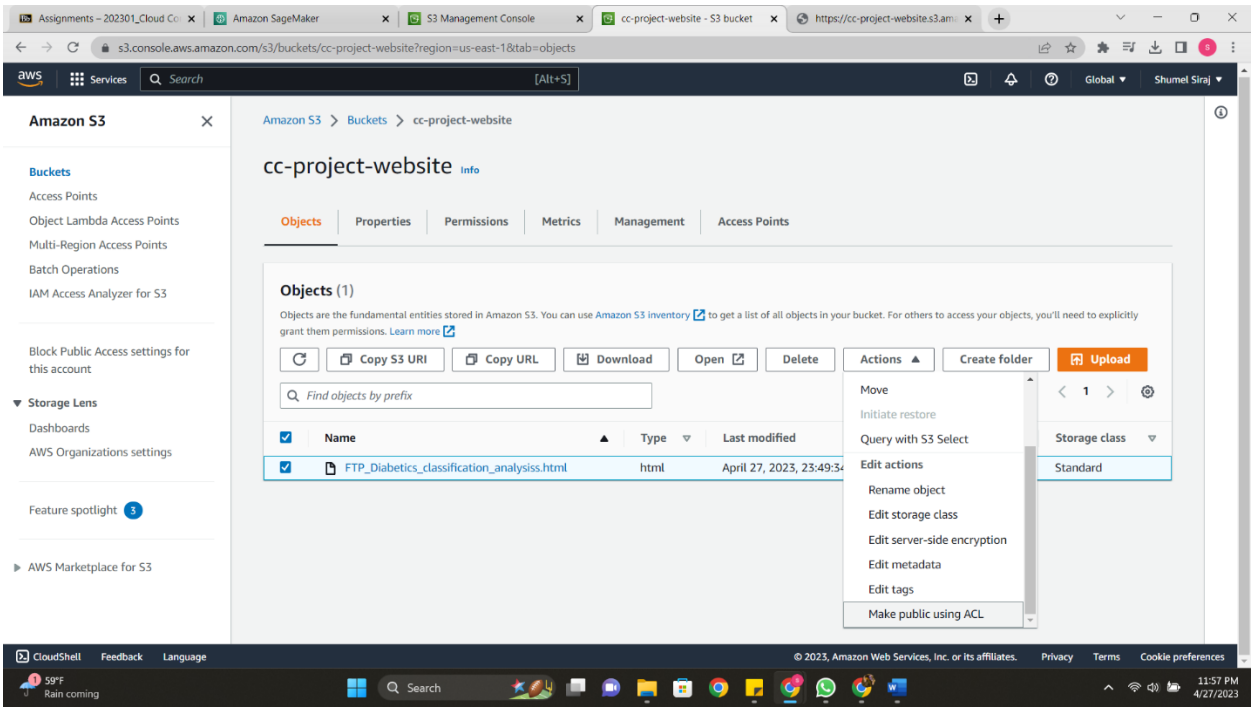
- In the S3 console, select your bucket and click the "Permissions" tab.
- Click on "Bucket Policy".
- Add a policy that allows public read access to your bucket.
- Save the policy.

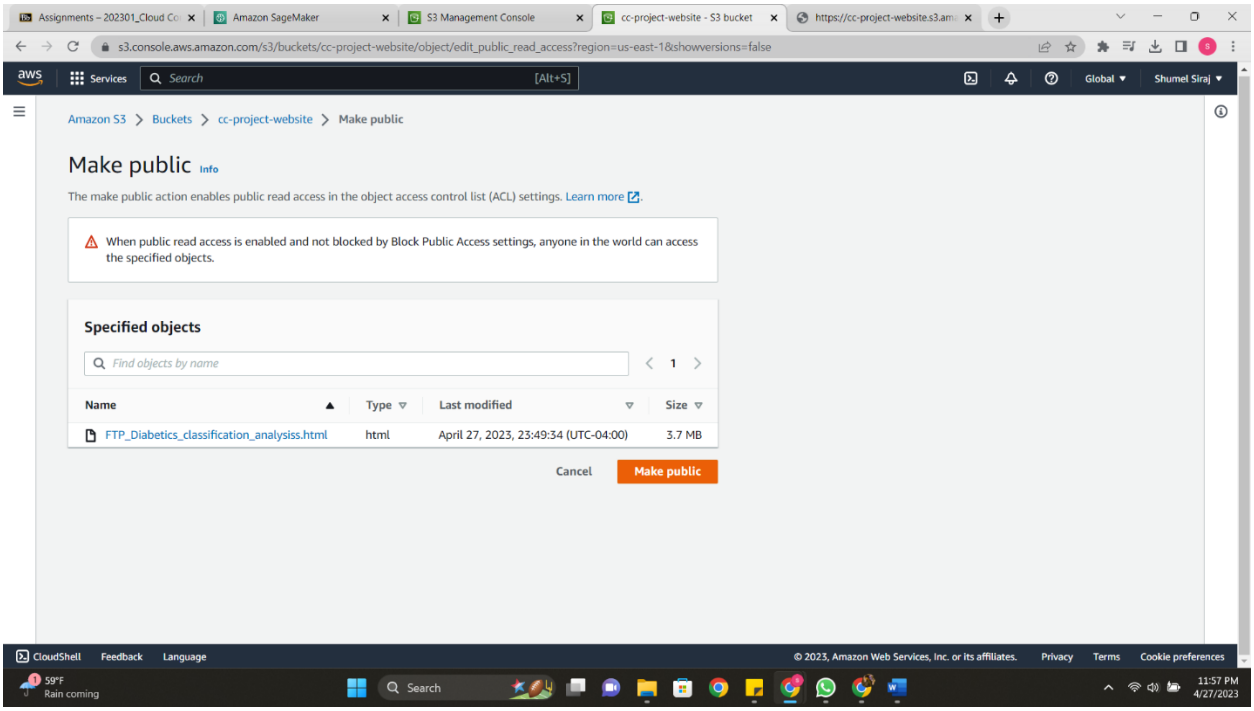

Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/bucket/cc-project-website/property/bpa/edit?region=us-east-1
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Edit Block public access (bucket settings) Info
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Cancel
Save changes
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:55 PM
4/27/2023

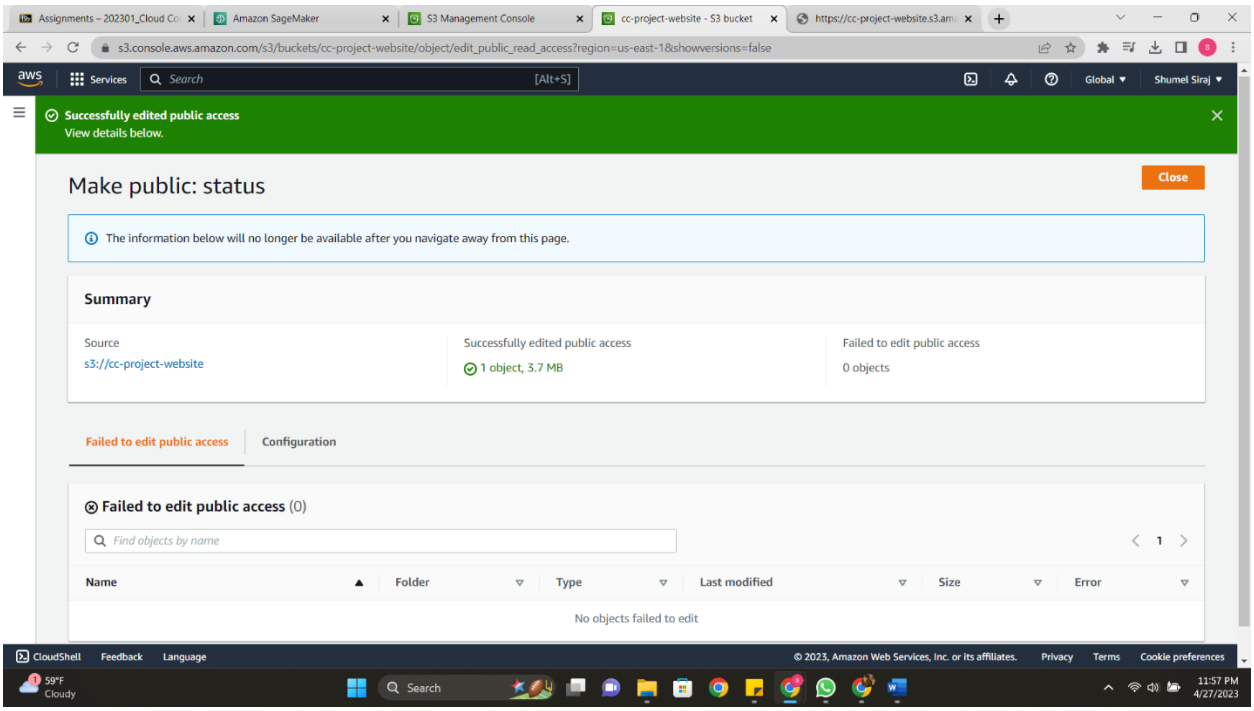

Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/buckets/cc-project-website?region=us-east-1&tab=permissions
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Successfully edited Block Public Access settings for this bucket.
Amazon S3 > Buckets > cc-project-website
cc-project-website Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access
Objects can be public
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Edit
Block all public access
Off
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:55 PM
4/27/2023


Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/bucket/cc-project-website/property/oo/edit?region=us-east-1
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > cc-project-website > Edit Object Ownership
Edit Object Ownership Info
Object Ownership
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.
Enabling ACLs turns off the bucket owner enforced setting for Object Ownership
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.
I acknowledge that ACLs will be restored.
Object Ownership
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:56 PM
4/27/2023


Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/buckets/cc-project-website?region=us-east-1&tab=permissions
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3
Successfully edited Object Ownership.
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > cc-project-website
cc-project-website Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access
Objects can be public
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Edit
Block all public access
Off
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:56 PM
4/27/2023

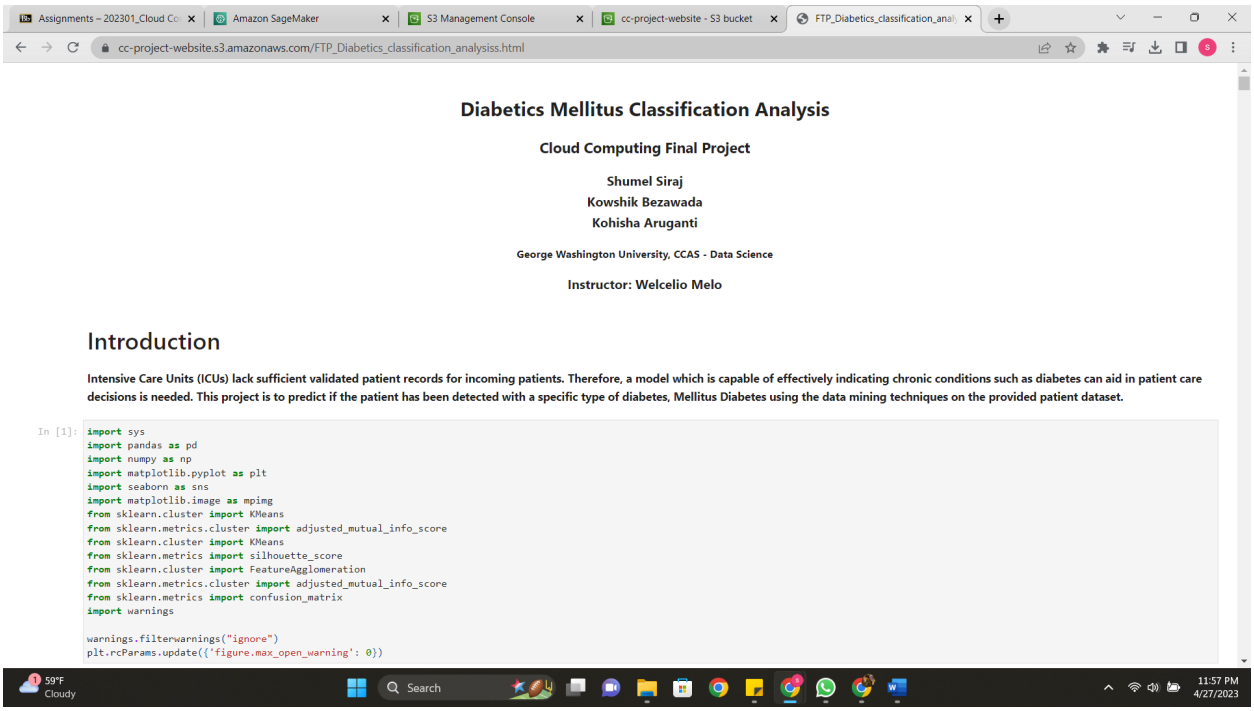

Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/buckets/cc-project-website?region=us-east-1&tab=objects
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > cc-project-website
cc-project-website Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Name
Type
Last modified
Storage class
FTP_Diabetics_classification_analysiss.html
html
April 27, 2023, 23:49:34
Standard
Move
Initiate restore
Query with S3 Select
Edit actions
Rename object
Edit storage class
Edit server-side encryption
Edit metadata
Edit tags
Make public using ACL
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:57 PM
4/27/2023


Assignments – 202301_Cloud Co
Amazon SageMaker
S3 Management Console
cc-project-website - S3 bucket
https://cc-project-website.s3.am
s3.console.aws.amazon.com/s3/buckets/cc-project-website/object/edit_public_read_access?region=us-east-1&showversions=false
Services
Search
[Alt+S]
Global
Shumel Siraj
Amazon S3 > Buckets > cc-project-website > Make public
Make public Info
The make public action enables public read access in the object access control list (ACL) settings. Learn more.
When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.
Specified objects
Find objects by name
Name
Type
Last modified
Size
FTP_Diabetics_classification_analysiss.html
html
April 27, 2023, 23:49:34 (UTC-04:00)
3.7 MB
Cancel
Make public
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
59°F
Rain coming
Search
11:57 PM
4/27/2023



15. Test Your Webpage

- Navigate to the URL provided in the "Static website hosting" section of your S3 bucket properties.
- Assess your webpage to make sure it's working correctly.
- Try entering different inputs to see if the SVM model is producing correct results.
- Make any necessary adjustments to the webpage or model based on your testing.



Assignments – 202301_Cloud Co | Amazon SageMaker | S3 Management Console | cc-project-website - S3 bucket | FTP_Diabetics_classification_anal

cc-project-website.s3.amazonaws.com/FTP_Diabetics_classification_analysiss.html

Diabetics Mellitus Classification Analysis

Cloud Computing Final Project

Shumel Siraj
Kowshik Bezawada
Kohisha Aruganti

George Washington University, CCAS - Data Science

Instructor: Welcelio Melo

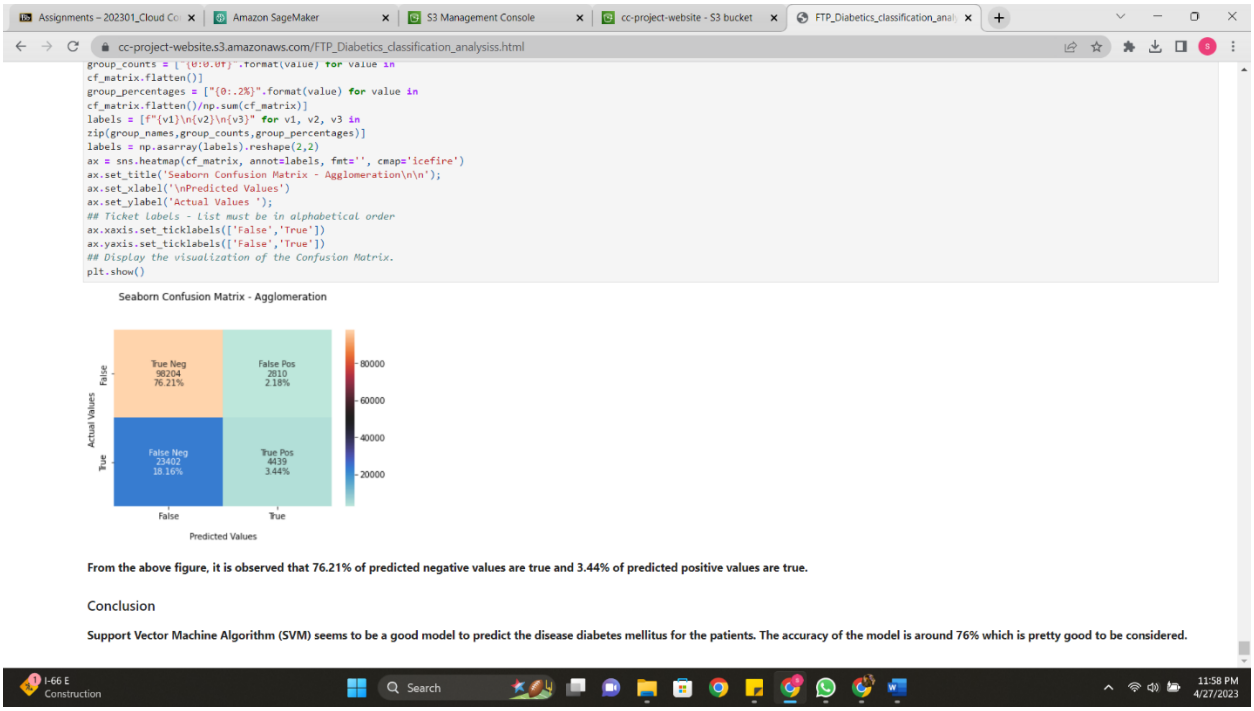
Introduction

Intensive Care Units (ICUs) lack sufficient validated patient records for incoming patients. Therefore, a model which is capable of effectively indicating chronic conditions such as diabetes can aid in patient care decisions is needed. This project is to predict if the patient has been detected with a specific type of diabetes, Mellitus Diabetes using the data mining techniques on the provided patient dataset.

In [1]:
```
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.image as mpimg
from sklearn.cluster import KMeans
from sklearn.metrics.cluster import adjusted_mutual_info_score
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.cluster import FeatureAgglomeration
from sklearn.metrics.cluster import adjusted_mutual_info_score
from sklearn.metrics import confusion_matrix
import warnings

warnings.filterwarnings("ignore")
plt.rcParams.update({'figure.max_open_warning': 0})
```

59°F Cloudy | Search | 11:57 PM 4/27/2023



Assignments – 202301_Cloud Co | Amazon SageMaker | S3 Management Console | cc-project-website - S3 bucket | FTP_Diabetics_classification_anal

cc-project-website.s3.amazonaws.com/FTP_Diabetics_classification_analysiss.html

```
group_counts = ["{0:0.0f}".format(value) for value in
cf_matrix.flatten()]
group_percentages = ["{0:.2%}".format(value) for value in
cf_matrix.flatten()/np.sum(cf_matrix)]
labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
zip(group_names,group_counts,group_percentages)]
labels = np.asarray(labels).reshape(2,2)
ax = sns.heatmap(cf_matrix, annot=labels, fmt='', cmap='icefire')
ax.set_title('Seaborn Confusion Matrix - Agglomeration\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');
## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['False','True'])
ax.yaxis.set_ticklabels(['False','True'])
## Display the visualization of the Confusion Matrix.
plt.show()
```

Seaborn Confusion Matrix - Agglomeration

From the above figure, it is observed that 76.21% of predicted negative values are true and 3.44% of predicted positive values are true.

Conclusion

Support Vector Machine Algorithm (SVM) seems to be a good model to predict the disease diabetes mellitus for the patients. The accuracy of the model is around 76% which is pretty good to be considered.

I-66 E Construction | Search | 11:58 PM 4/27/2023

Conclusion

You have successfully trained a Diabetes Mellitus classification model using an SVM on AWS SageMaker and hosted the Jupyter Notebook as a static webpage on AWS S3 by following

these steps. This powerful combination enables you to create, deploy, and share machine learning models and results quickly and efficiently. By leveraging AWS services, you have streamlined the process of developing and deploying a Diabetes Mellitus classification model, making it accessible to a wider audience through a static webpage.