CS-524 Intro to Cloud Computing

Creating a BitCoin Price Prediction Model Using AWS SageMaker

Final Report

Members:

David DeLaus, Abhishek Desai, Hardi Patel, Jayant Kumar, Jiahan Liu, Vishal Kuchadi, Malika Thakur

(TEAM 5)

CONTENTS

INTRODUCTION

Abstract

Bitcoin is the most well-known and longest-running cryptocurrency, having been published as an open source project in 2009 by the anonymous Satoshi Nakamoto. Bitcoin is a decentralized digital currency that allows transactions to be checked and documented in a public distributed ledger (the blockchain) without the need for a trustworthy record-keeping authority or a central intermediary. A transaction block is made up of a collection of transactions.

Content

bitstampUSD 1-min data 2012-01-01 to 2021-03-31.csv is the dataset used for select bitcoin exchanges, with minute-by-minute updates of OHLC (Open, High, Low, Close), for the period January 2012 to December March 2021. Unix time is used for timestamps. The data fields of timestamps with no trades or operation are loaded with NaNs.

How Bitcoin Works:

Bitcoin is the most popular and valuable cryptocurrency that is currently being traded and mined. It was launched in 2009 and ever since then the price has continued to rise with the ever increasing demand. The demand for Bitcoin is due to its nature of being a digital currency and the scarcity that is caused by its ever increasing difficulty in mining. Mining is the process of creating new Bitcoin. It works by having computers solve very difficult math problems that help update the blockchain ledger for the cryptocurrency and create new coins.

Technology Used

AWS S3 Bucket

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers elite levels of performance, versatility, and accessibility. It has very easy to use and understand management tools allowing the user to customize their buckets to fit their needs.

You can simply upload the resources you need to the S3 Bucket and then access it anytime from a website or another AWS service. There is no charge for an S3 Bucket as it is in the free tier.

Benefits:

1. Fast and simple to access
2. Available anytime
3. High read speeds

AWS SageMaker

AWS SageMaker is Amazon's most comprehensive machine learning service. It is the first IDE for machine learning and it was designed from the ground up to be as functional as possible. SageMaker comes with the SageMaker studio which simplifies the process for the development of machine learning models.

All you need to do is upload a notebook file to the service and it will run it for you at a fraction of the time it would take a personal machine to run the algorithms.

Benefits:

1. Much faster than using a personal computer to train models
2. Very easy to work with
3. Can be accessed by multiple people

AWS Sagemaker JupyterLab

JupyterLab is a web-based development platform that allows you to work with Jupyter notebooks, code, and data. You can write and test code in Jupyter notebooks without having to recompile entire files because you can compile line by line. Notebooks often make it simple to share your code by allowing you to share it as a notebook, pdf, or html file.

Benefits:

1. Easily share code to speed development process
2. Lightweight to run on any machine
3. Accepted by AWS SageMaker

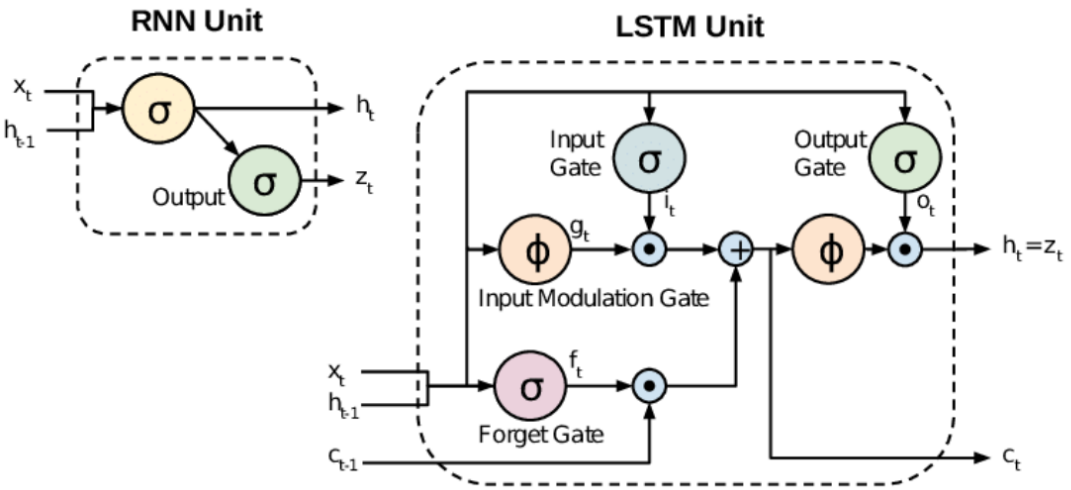
Prediction Models

Simple RNN

RNN or recurrent neural network is a class of artificial neural networks. It works by using connections between nodes to form directed graphs that can be used to then predict outcomes of events.

LSTM

LSTM which stands for Long short-term memory. It is an artificial RNN that is used in the deep learning field of study. It is commonly used for speech recognition, machine translation, and financial modeling. It works by using feedback connections as opposed feedforward.



Deployment Guide

The project deployment works in 4 major stages:

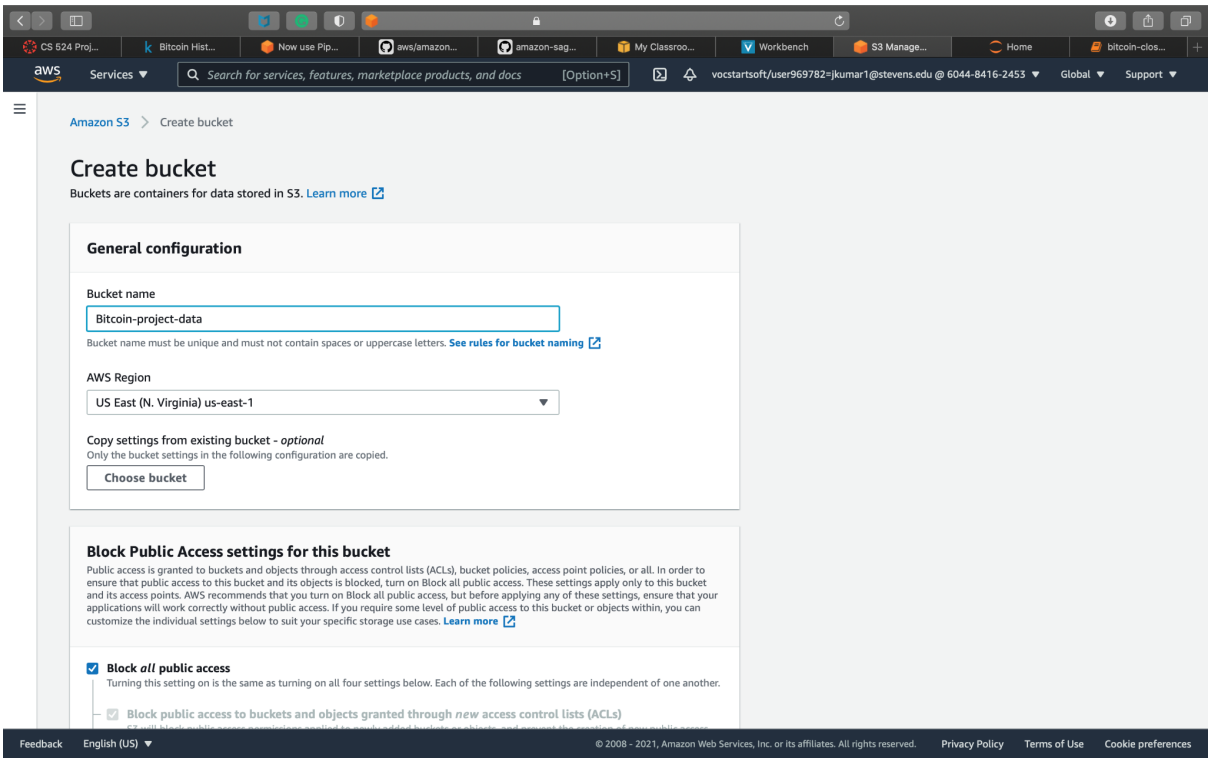
1. Setting up the A**WS service i.e S3 bucket, and Creating AWS Sagemaker notebook instance with ml.t2.medium and AWS IAM instance.**
2. Writing code for Data analysis and cleaning, and
3. Writing the machine learning code in **Python** and uploading it to the Sage Maker instance and deploying it on AWS.
4. Deleting the environment

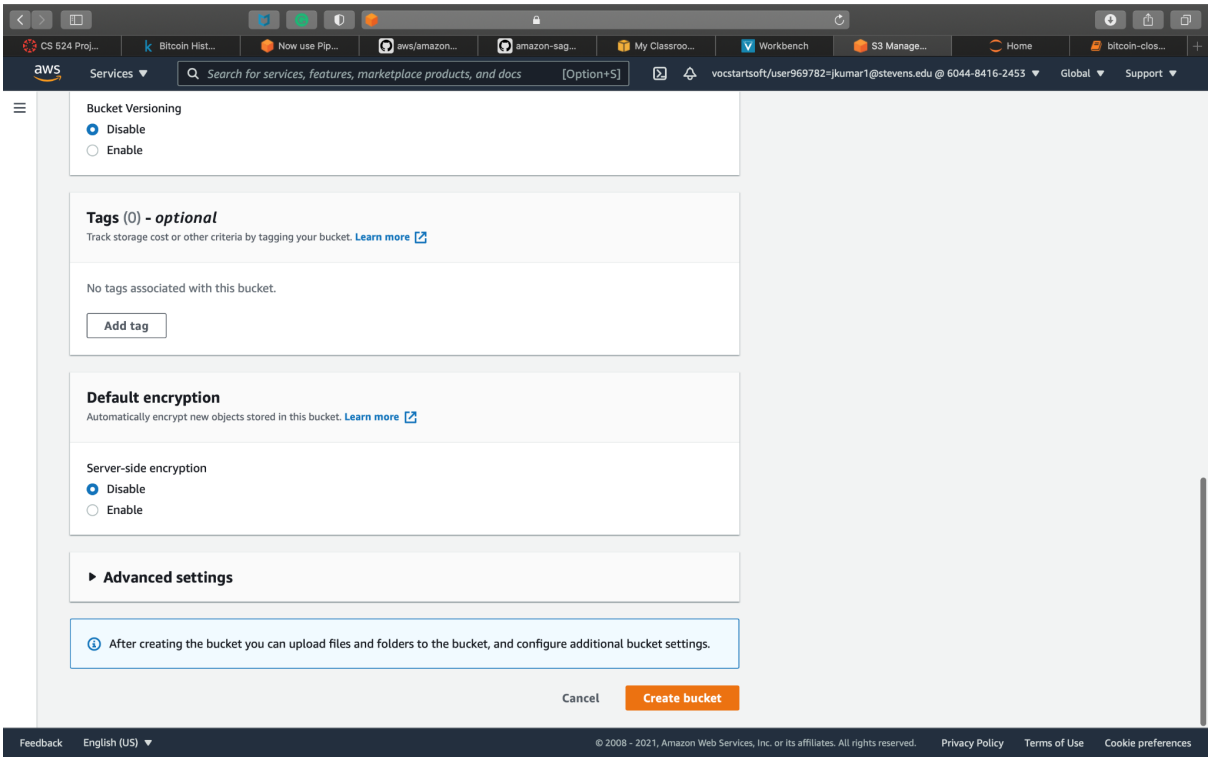
Setting Up AWS Services

We decided to use SageMaker as it was a strong machine learning service that AWS has and seemed to fit our needs the best. We also had to utilize an S3 bucket in order to have SageMaker access the bitcoin sale data.

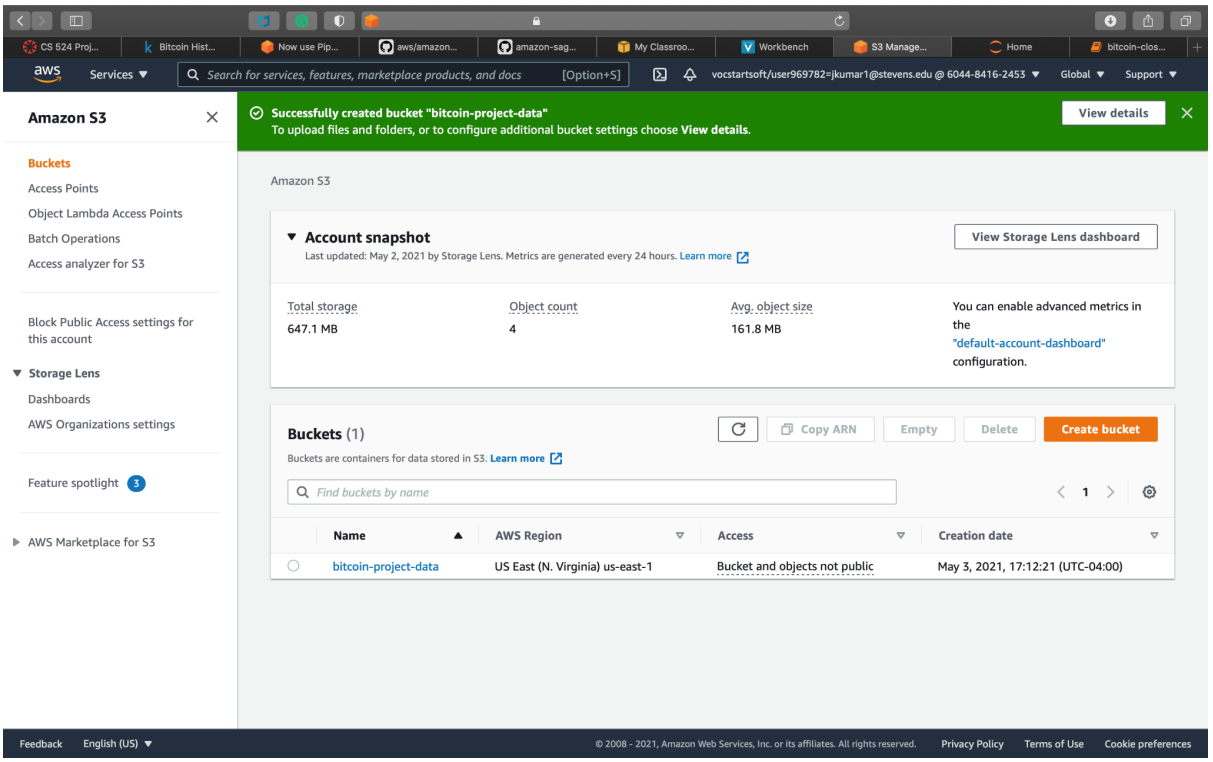
Step 1: Creating a S3 bucket for our data.

Add bucket name, Select AWS region, Block all Public access for not giving public access of the URL.
Select the versioning, encryption options and click on CREATE BUCKET.

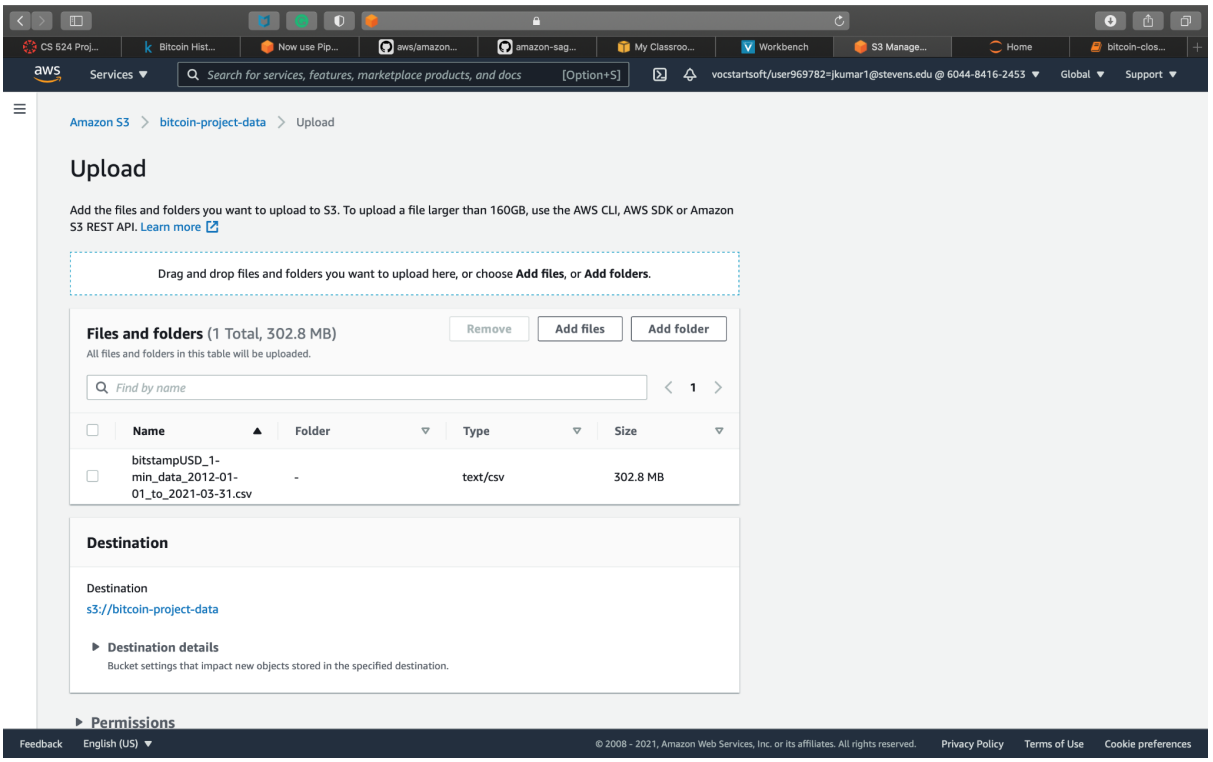


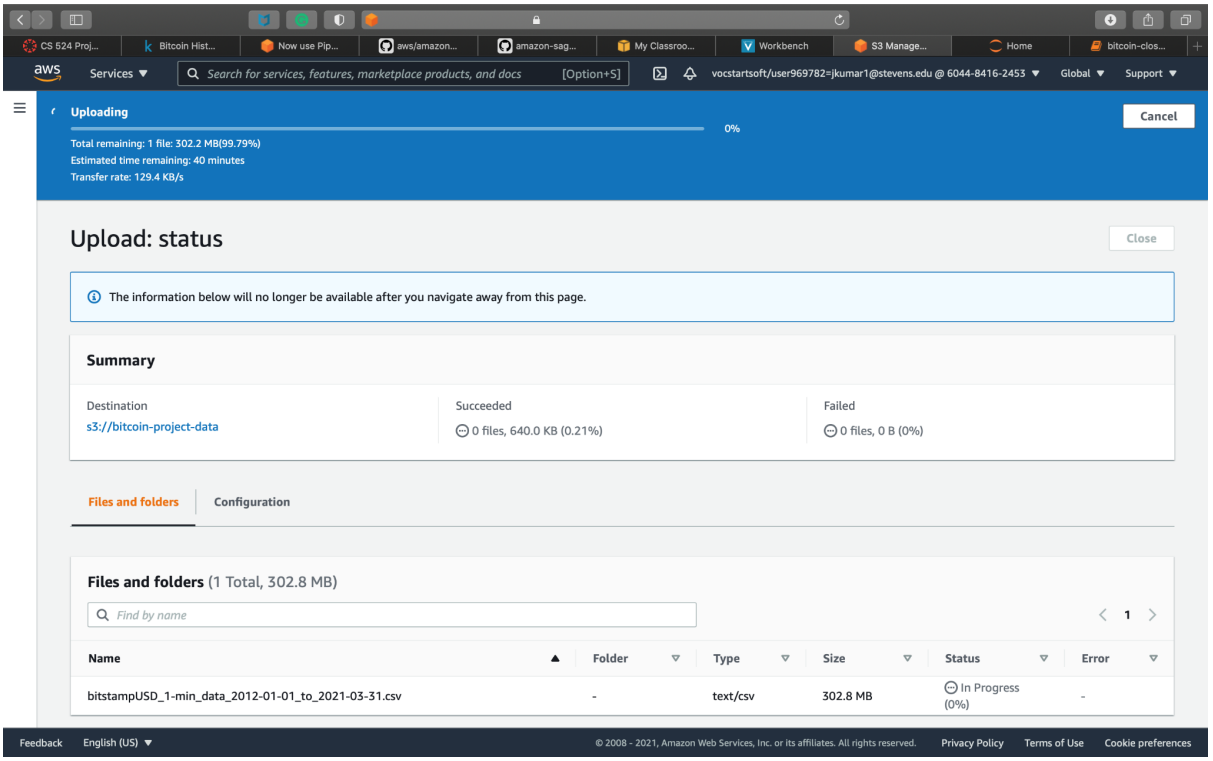


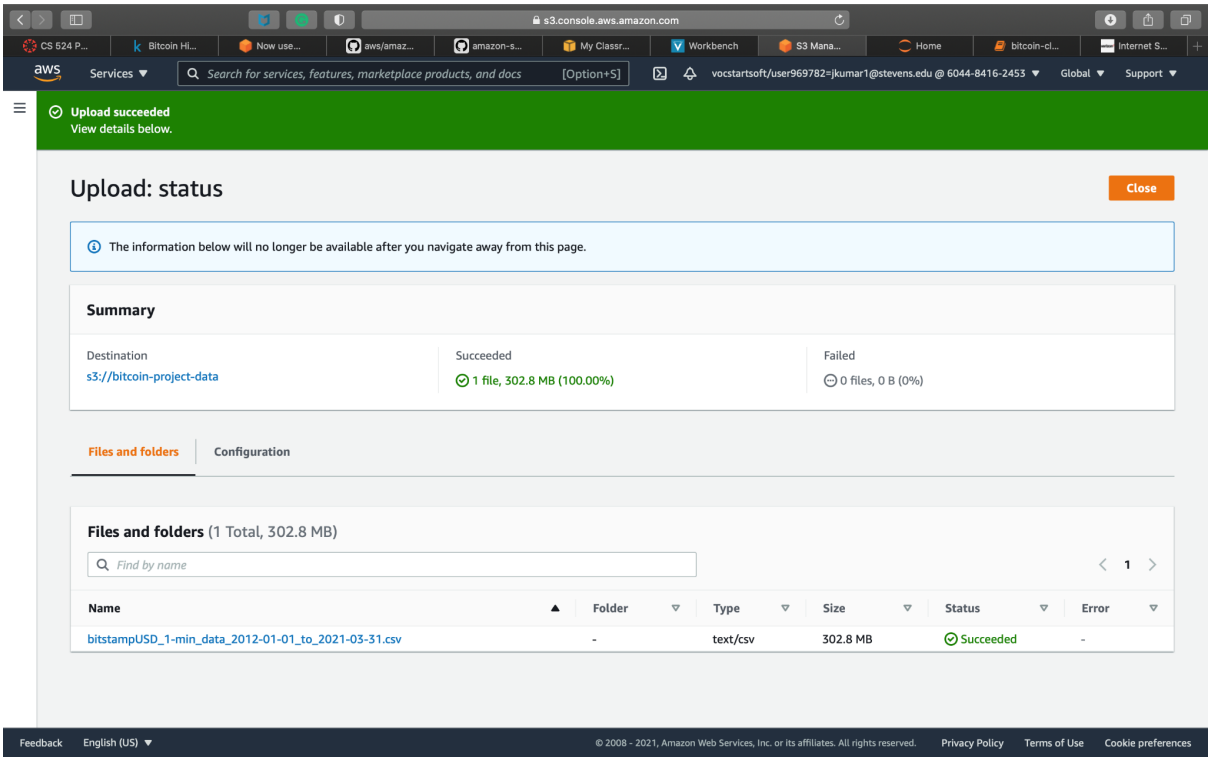
S3 bucket has been successfully created.



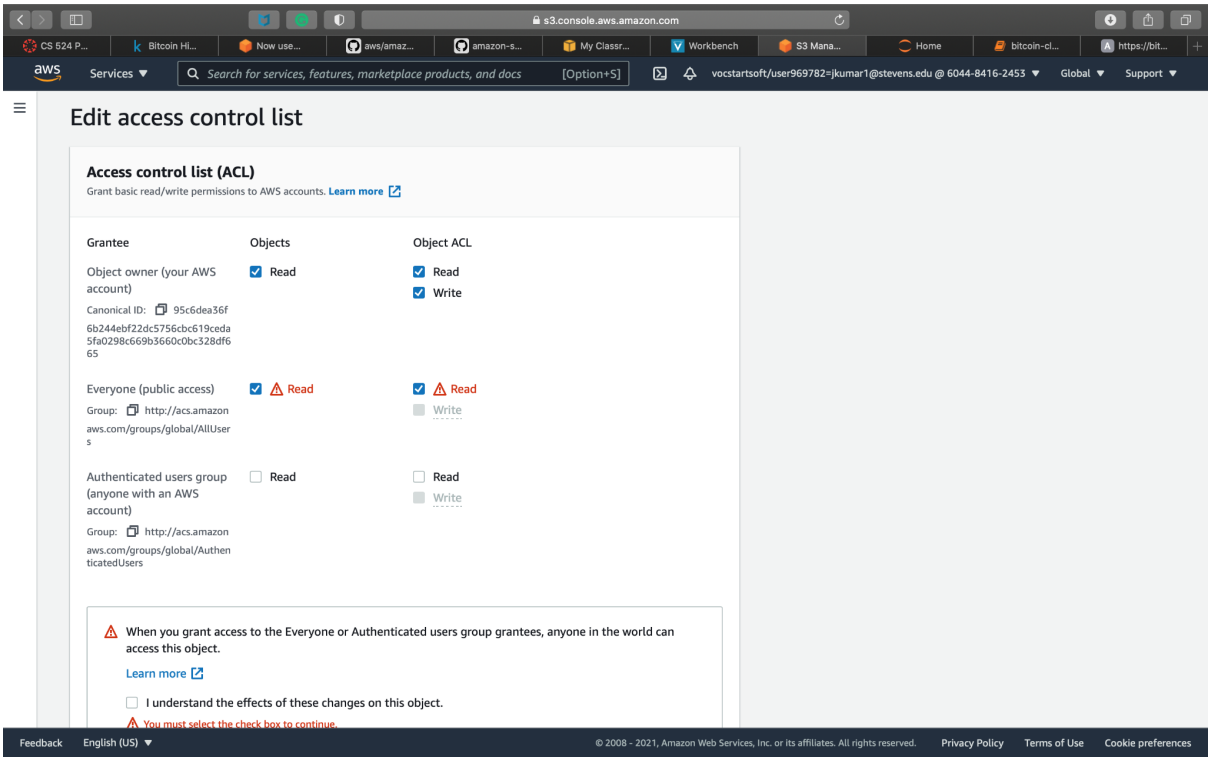
Step 2: Uploading our data to our bucket.

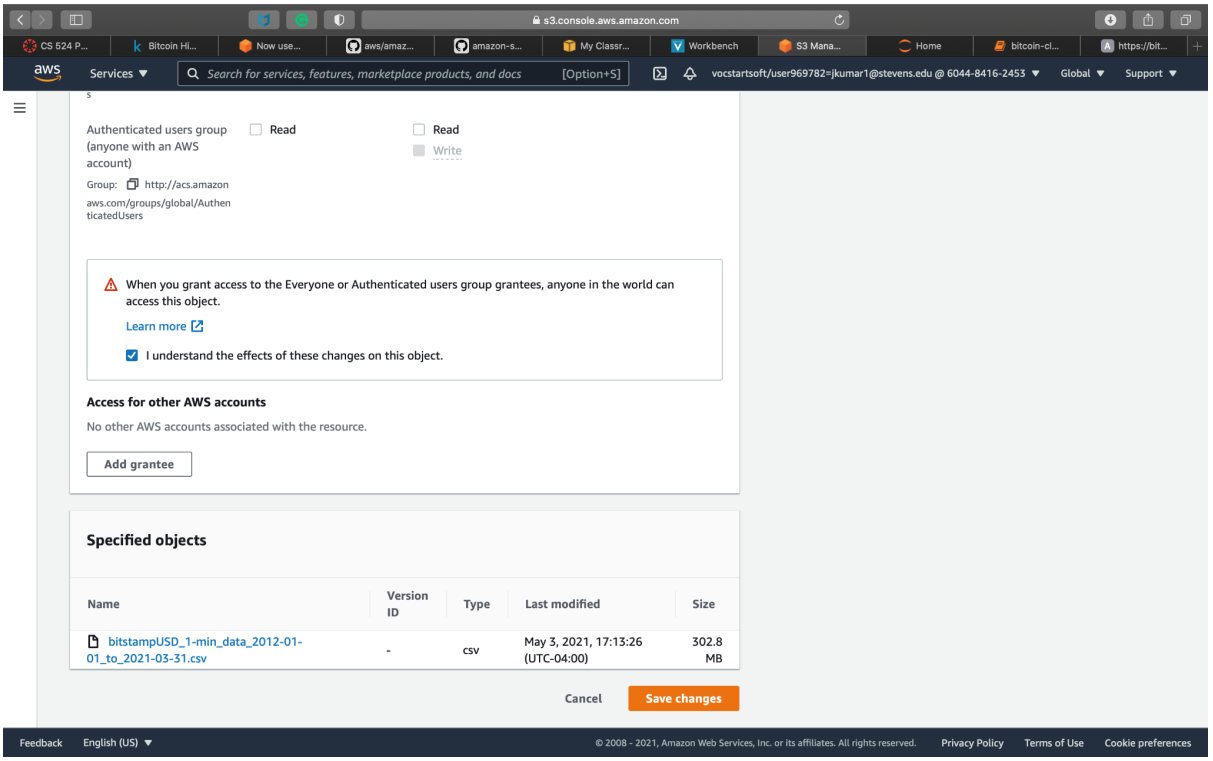




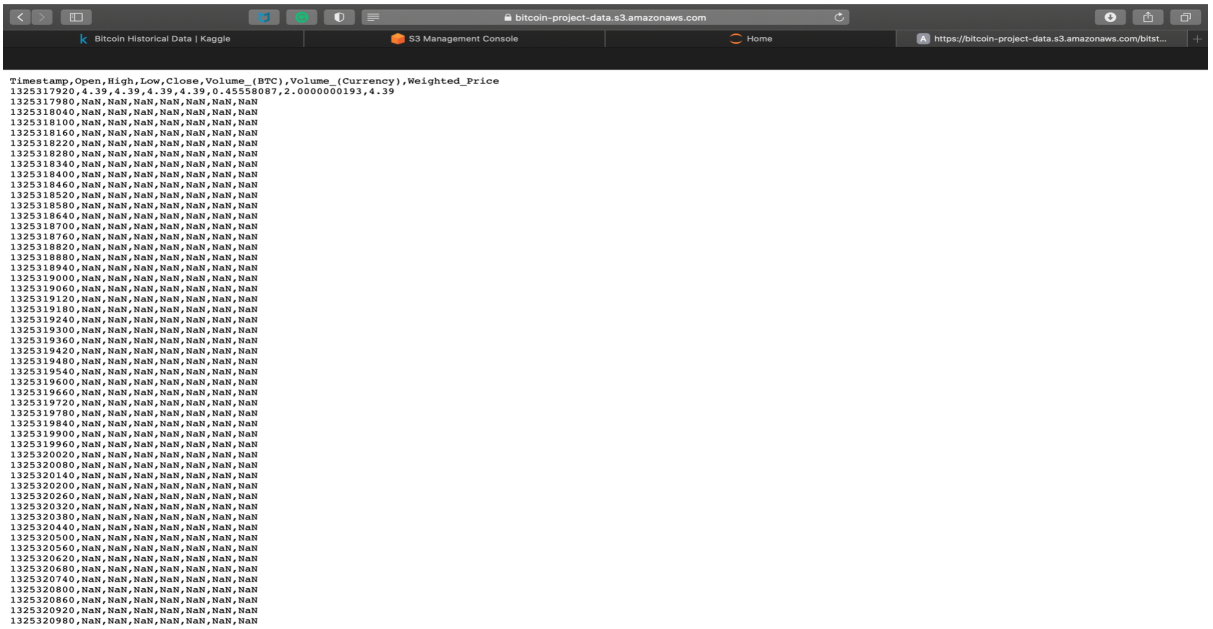


Step 3: Configuring our access list to make sure the SageMaker instance could have access to it.



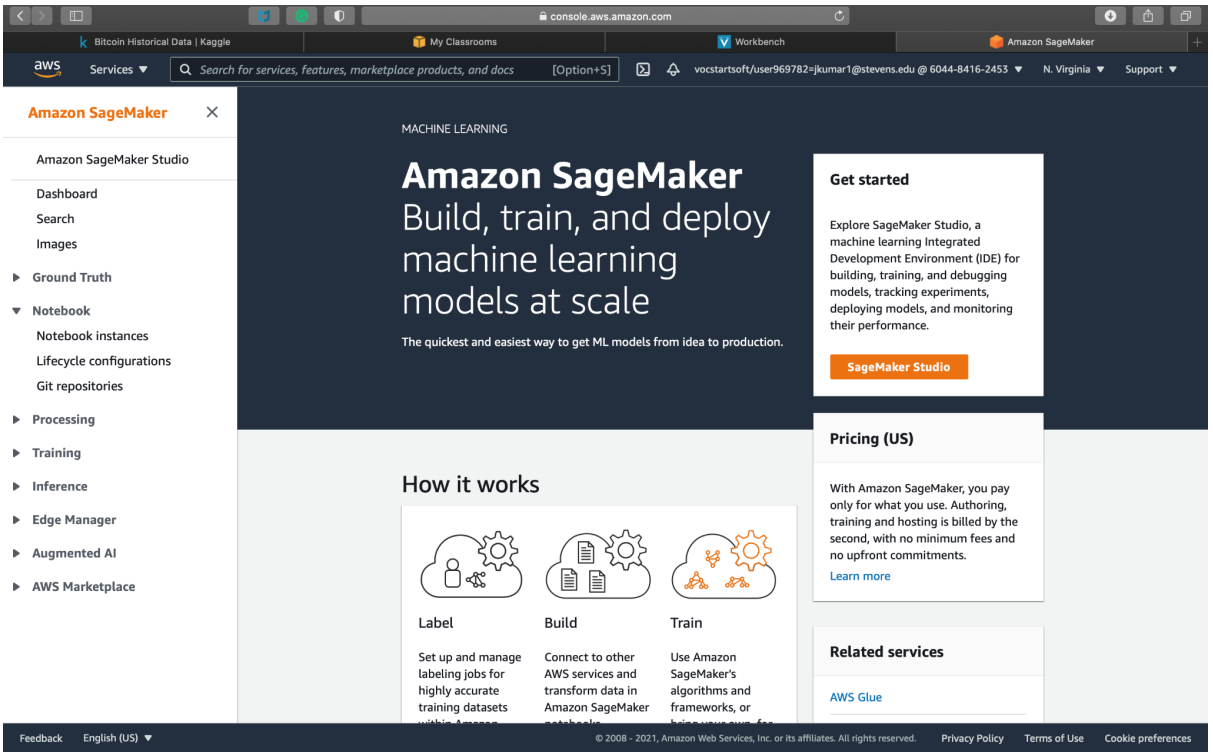


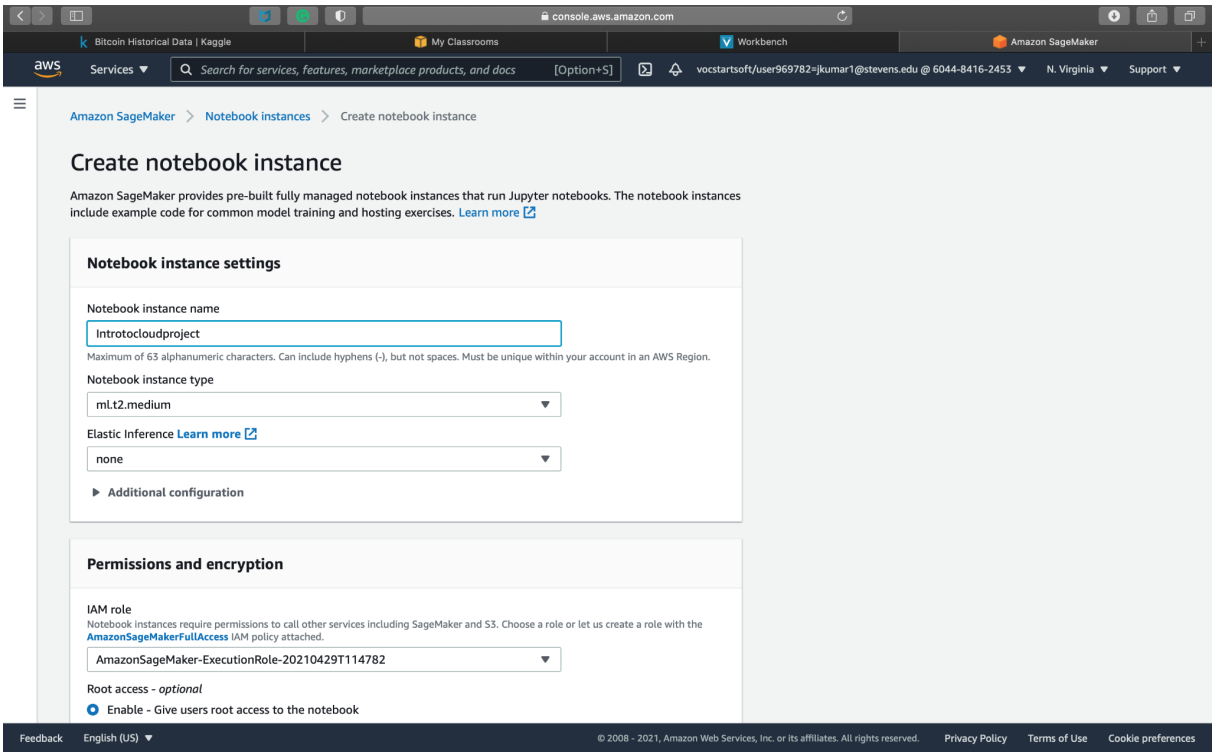
URL of our dataset extracted from Kaggle and uploaded in our S3 bucket: https://bitcoin-project-data.s3.amazonaws.com/bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv



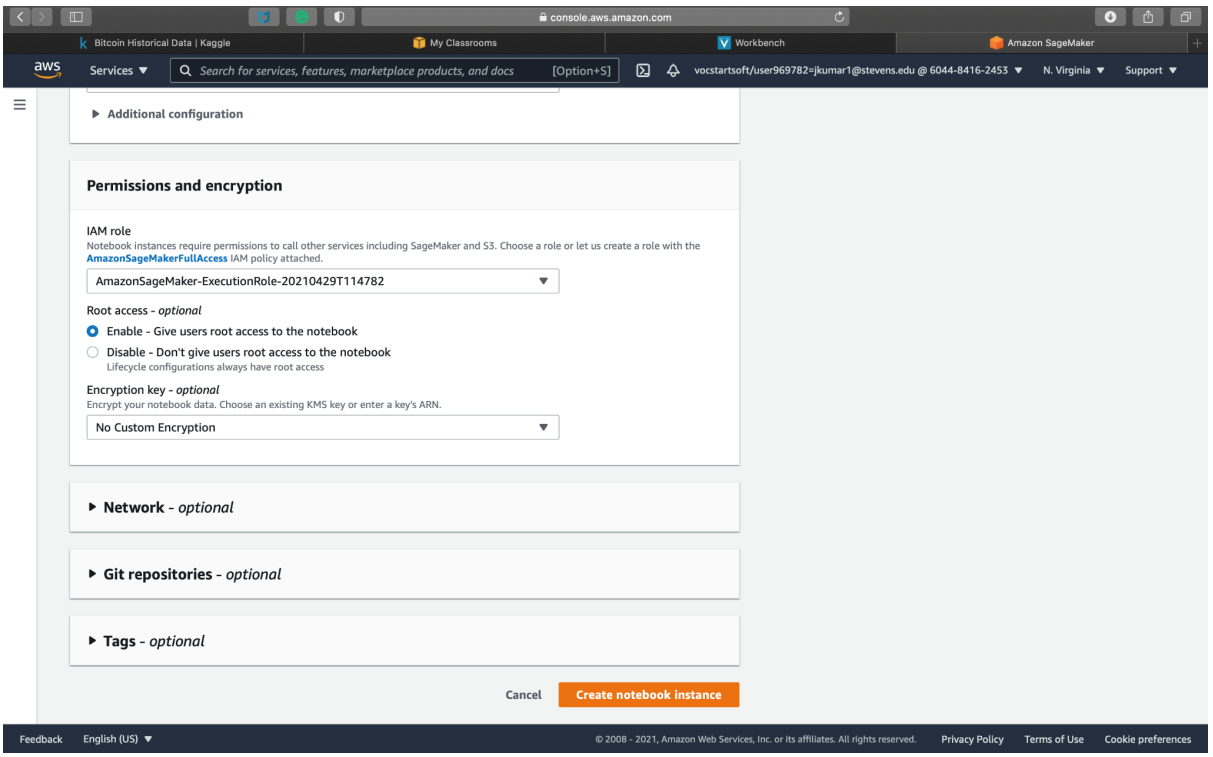
Step 4: Creating our SageMaker instance

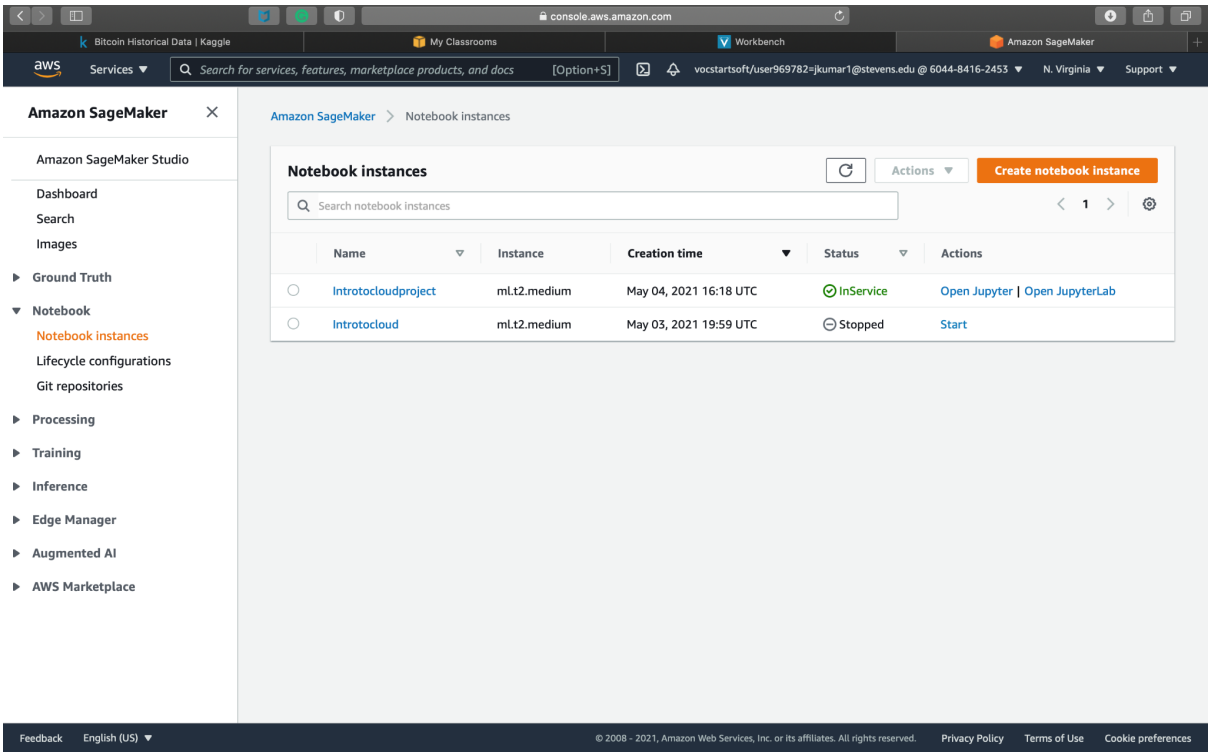
Give the “notebook instance name”, “notebook instance type” , enable “IAM role” and keep the root access “enable”.

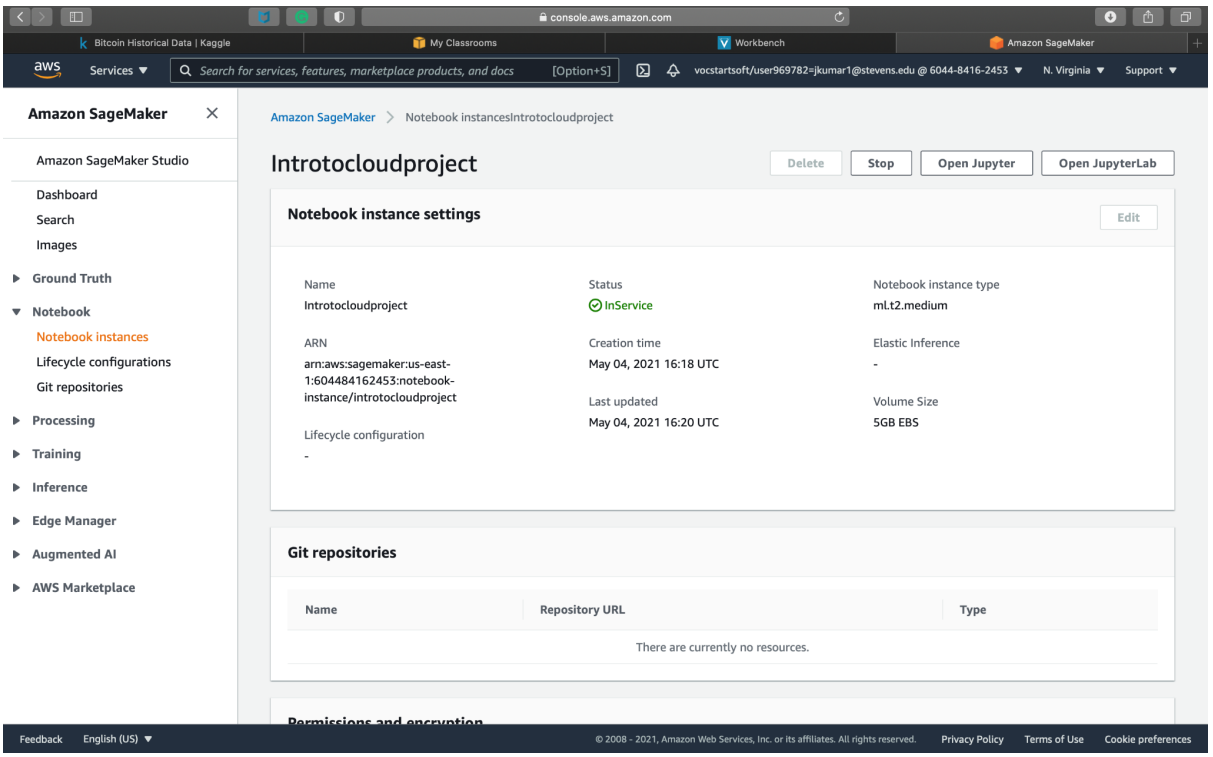




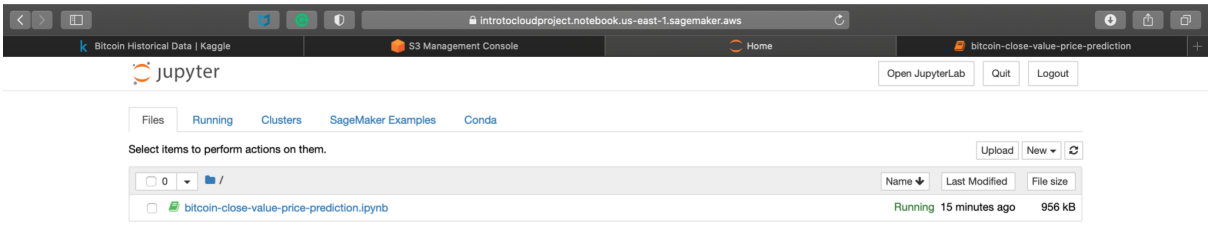
- Click on “Create Notebook Instance” and check the status - “**InService**”







- Open the previously created **Sagemaker Jupyter Notebook.**
  Url for jupyter notebook:
  https://introtocloudproject.notebook.us-east-1.sagemaker.aws/tree



Data Analysis and Cleaning:

Step 1: Exploratory Data Analysis

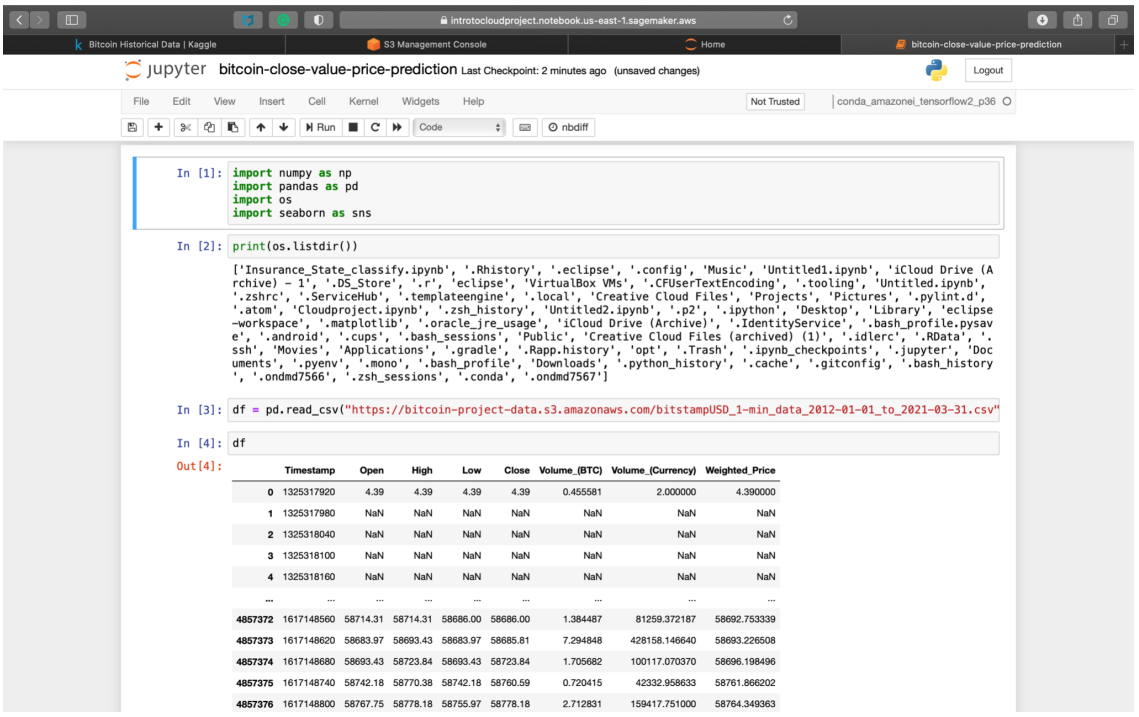
This is the standard first step for any machine learning project. There were large amounts of junk data found in the csv file given by kaggle. The first step is to examine all the data through a pandas dataframe. While using Trend Analysis we figure it out our data analysis by importing matplotlib, scipy, pandas, numpy, seaborn, ARIMA model frameworks.

Data features to be considered for prediction:

The opening and closing price on a particular day is given by the **Open and Close** columns.

The highest and the lowest price on a particular day is given by **High and Low** columns.

The total volume of traded on a particular day is defined by **Volume** column and The **Weighted price** is a trading benchmark which is mostly used by traders. Both the trend and value of a security insight info is given by the weighted price to traders.We are predicting here the close value of Bitcoin price using deep learning models.



introtocloudproject.notebook.us-east-1.sagemaker.aws

Bitcoin Historical Data | Kaggle | S3 Management Console | Home | bitcoin-close-value-price-prediction

jupyter bitcoin-close-value-price-prediction Last Checkpoint: 2 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted conda_amazonei_tensorflow2_p36

Run Code nbdiff

In [3]:
```
df = pd.read_csv("https://bitcoin-project-data.s3.amazonaws.com/bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv"
```

In [4]:
```
df
```

Out[4]:

	Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price
0	1325317920	4.39	4.39	4.39	4.39	0.455581	2.000000	4.390000
1	1325317980	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1325318040	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	1325318100	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	1325318160	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
4857372	1617148560	58714.31	58714.31	58686.00	58686.00	1.384487	81259.372187	58692.753339
4857373	1617148620	58683.97	58693.43	58683.97	58685.81	7.294848	428158.146640	58693.226508
4857374	1617148680	58693.43	58723.84	58693.43	58723.84	1.705682	100117.070370	58696.198496
4857375	1617148740	58742.18	58770.38	58742.18	58760.59	0.720415	42332.958633	58761.866202
4857376	1617148800	58767.75	58778.18	58755.97	58778.18	2.712831	159417.751000	58764.349363

4857377 rows × 8 columns

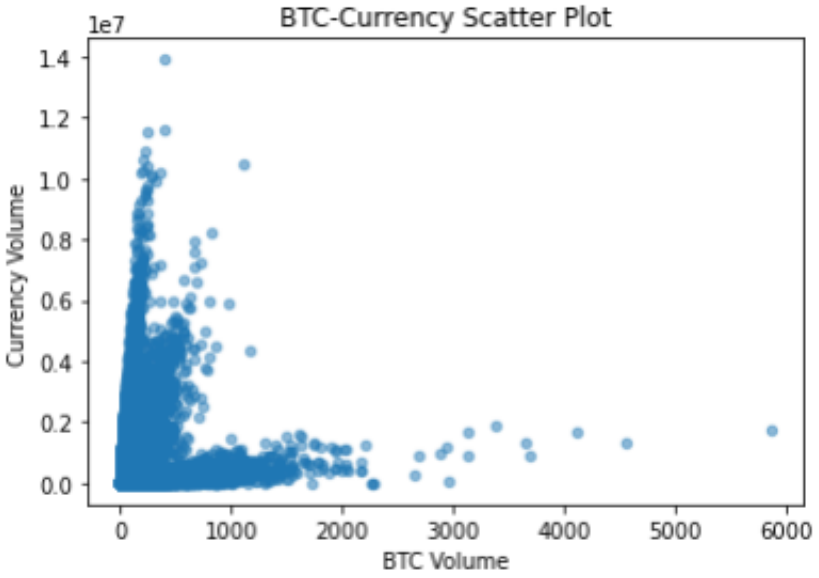
In [5]:
```
df.isnull().sum()
```

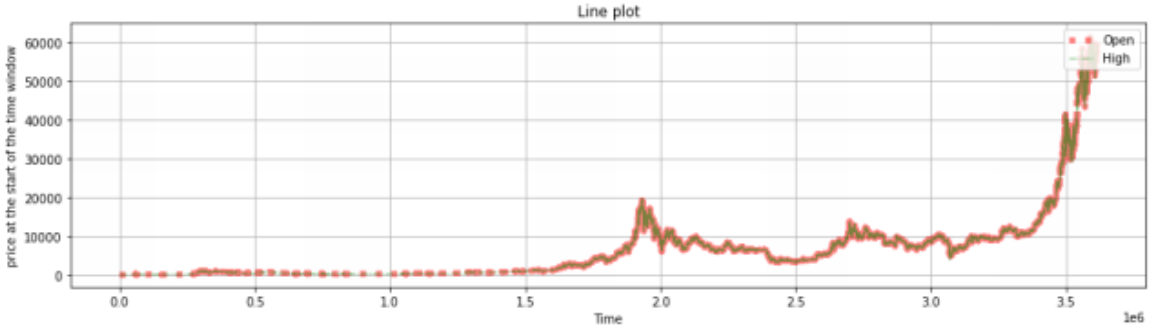
Out[5]:
```
Timestamp              0
Open             1243608
High             1243608
Low              1243608
Close            1243608
Volume_(BTC)     1243608
Volume_(Currency)  1243608
Weighted_Price   1243608
dtype: int64
```

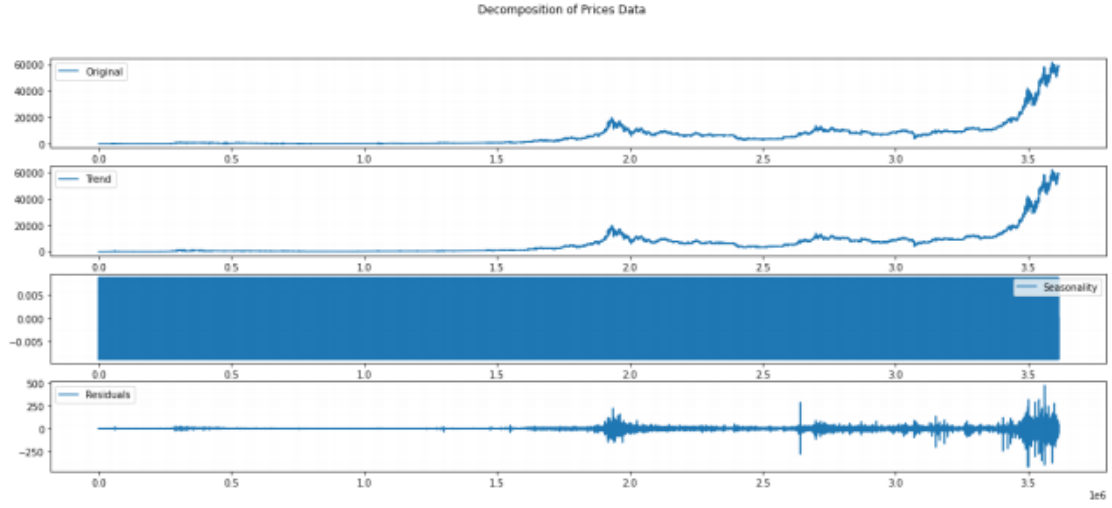
In [6]:
```
df.dropna(inplace= True)
```

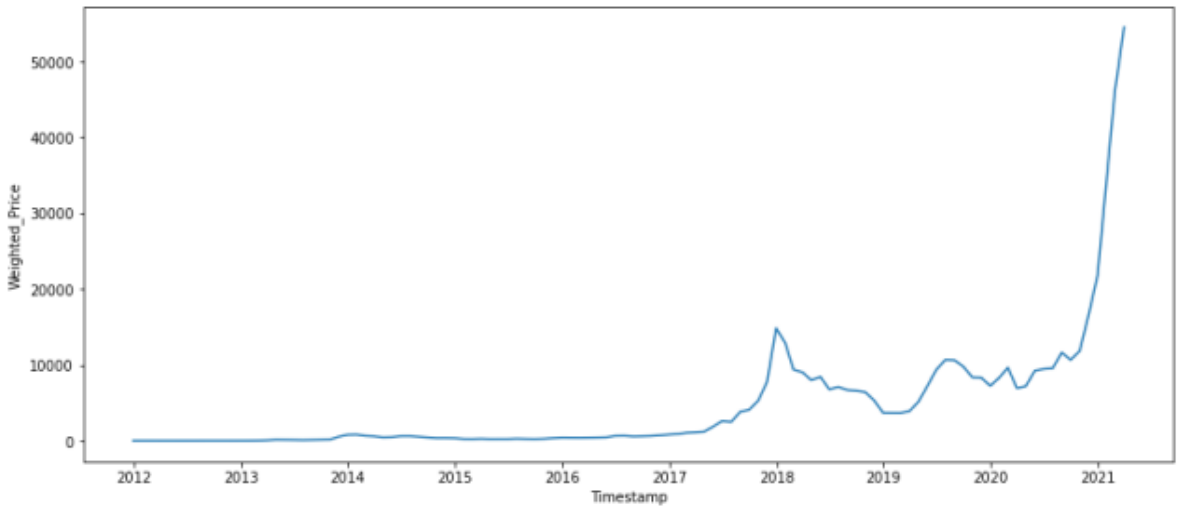
Data Visualization

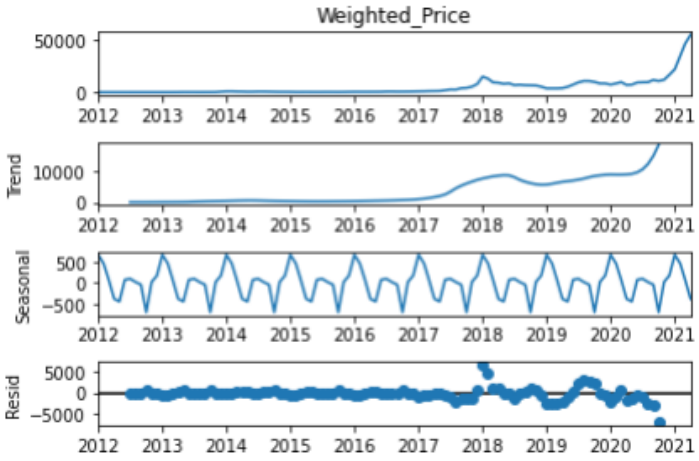
Following these first steps of examining the data we then began to **visualize** different aspects of the data in order to have a better understanding of the **various correlations that existed within the data that can impact the predicted price.**

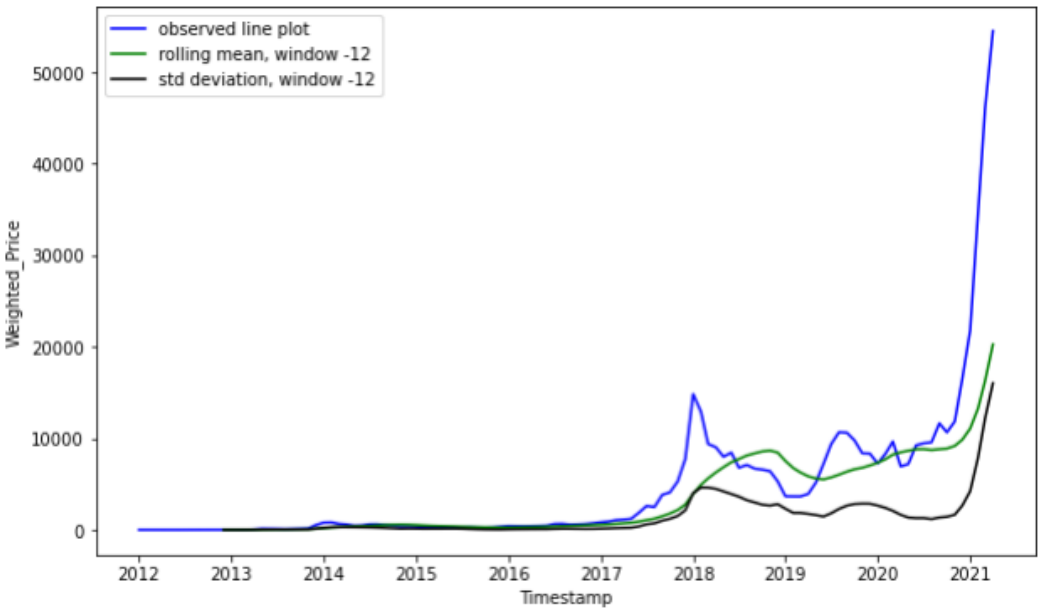

BTC-Currency Scatter Plot
1e7
Currency Volume
BTC Volume
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
1000
2000
3000
4000
5000
6000

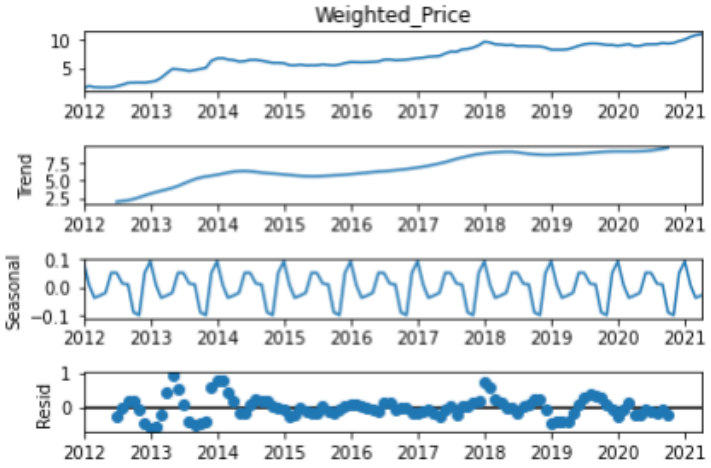

Line plot
price at the start of the time window
Time
1e6
Open
High
0
10000
20000
30000
40000
50000
60000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

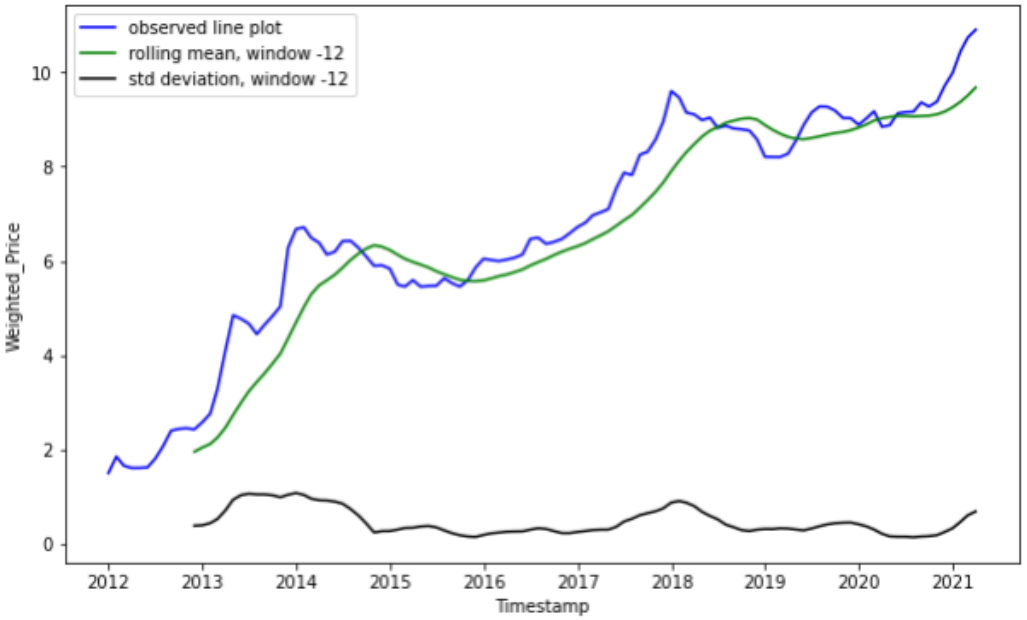

Decomposition of Prices Data
Original
Trend
Seasonality
Residuals
1e6

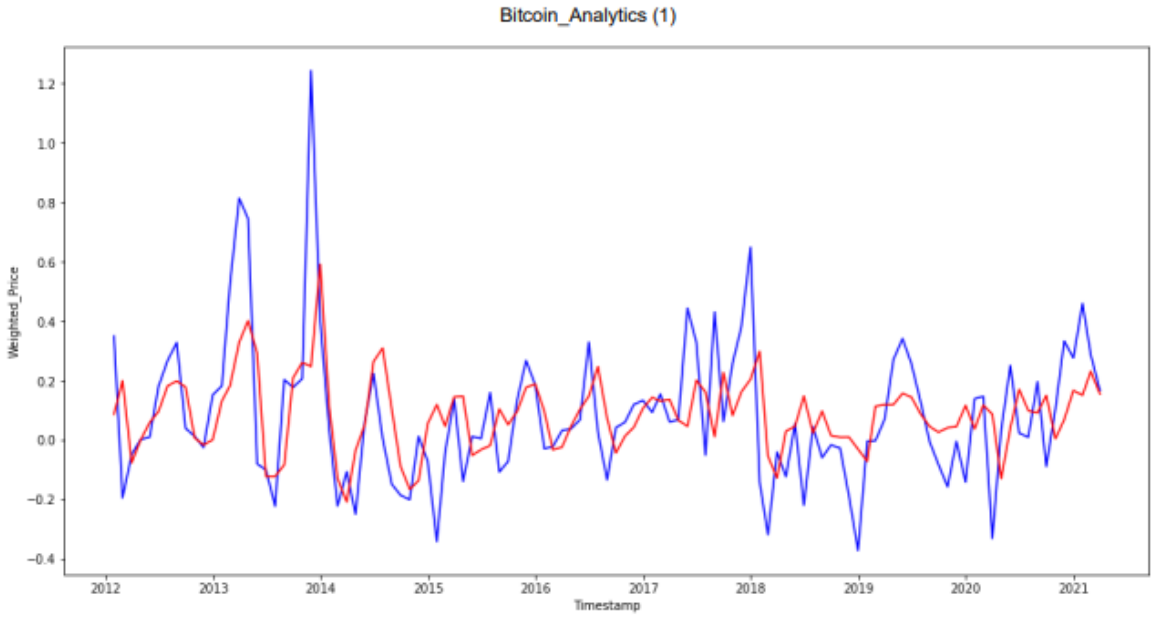

Weighted_Price
50000
40000
30000
20000
10000
0
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
Timestamp

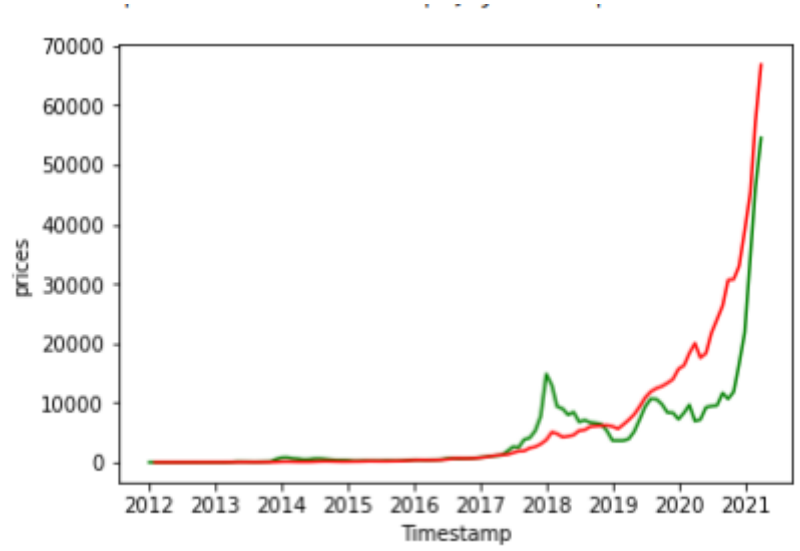

Weighted_Price
Trend
Seasonal
Resid


observed line plot
rolling mean, window -12
std deviation, window -12
Weighted_Price
Timestamp


Weighted_Price
Trend
Seasonal
Resid
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


observed line plot
rolling mean, window -12
std deviation, window -12
Weighted_Price
Timestamp


Bitcoin_Analytics (1)
Weighted_Price
Timestamp



After doing our data analysis it was determined that we should use all the data types for our model as they all provided valuable information for training our model. We broke it down into x_train, x_test, y_train, y_test. The x values were all the data points besides the closing price and the y values were the closing price.

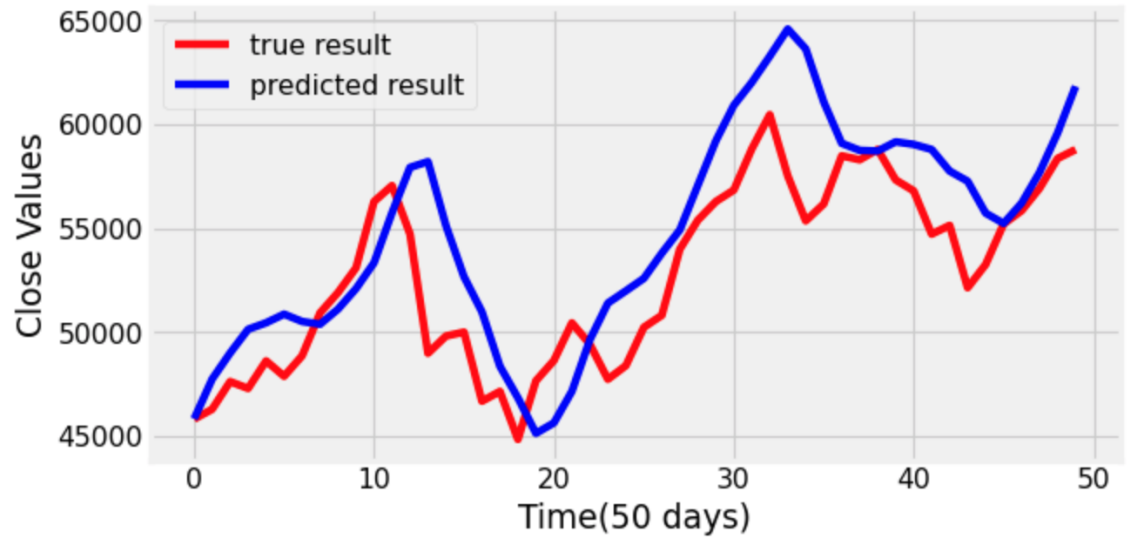
Creating the Models

LSTM Model

The first model we used was LSTM, this model proved very effective at predicting the price of bitcoin.

Using LSTM

```
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout,Flatten

Lstm_m=Sequential()

Lstm_m.add(LSTM(10,input_shape=(None,1)))

Lstm_m.add(Activation('relu'))

Lstm_m.add(Dense(1))

Lstm_m.compile(loss="mean_squared_error",optimizer="adam")

Lstm_m.fit(x_train,y_train,epochs=50,batch_size=32)
```

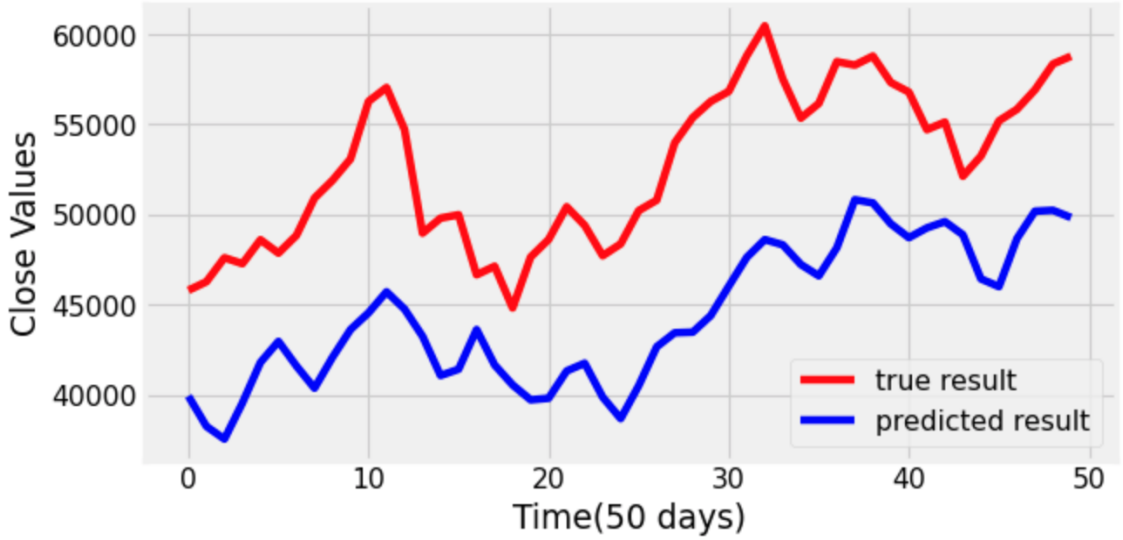


As it can be seen by the above graph our model was very close to the correct value over the course of the testing data.

Simple RNN model

RNN model

```
model=Sequential()

model.add(SimpleRNN(128,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(SimpleRNN(256,return_sequences=True,activation='relu'))
model.add(Dropout(0.20))

model.add(SimpleRNN(256,return_sequences=True,activation='relu'))
model.add(Dropout(0.30))

model.add(SimpleRNN(512,return_sequences=True,activation='relu'))
model.add(Dropout(0.45))

model.add(SimpleRNN(256,return_sequences=True,activation='relu'))
model.add(Dropout(0.20))

model.add(SimpleRNN(128,return_sequences=True,activation='relu'))
model.add(Dropout(0.20))

model.add(Flatten())

model.add(Dense(1))


model.compile(optimizer='adam',loss='mean_squared_error')
model.fit(x_train,y_train,epochs=50,batch_size=64)

```

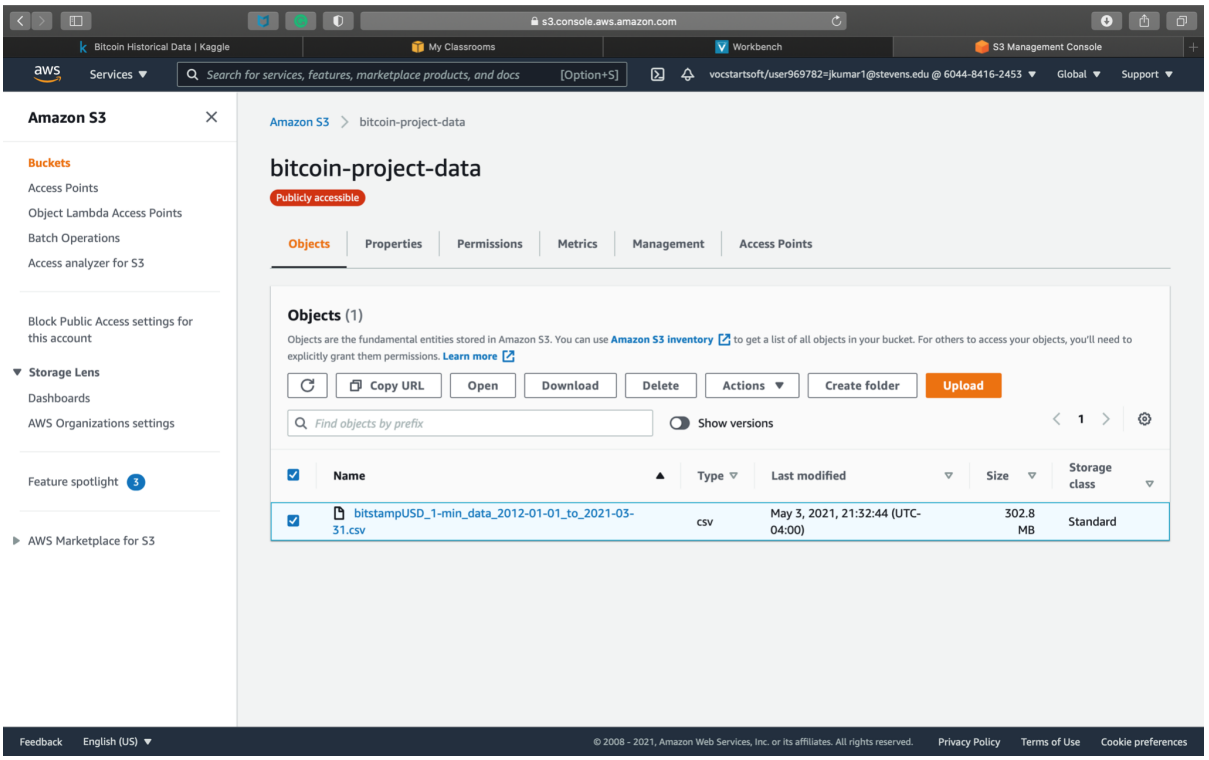


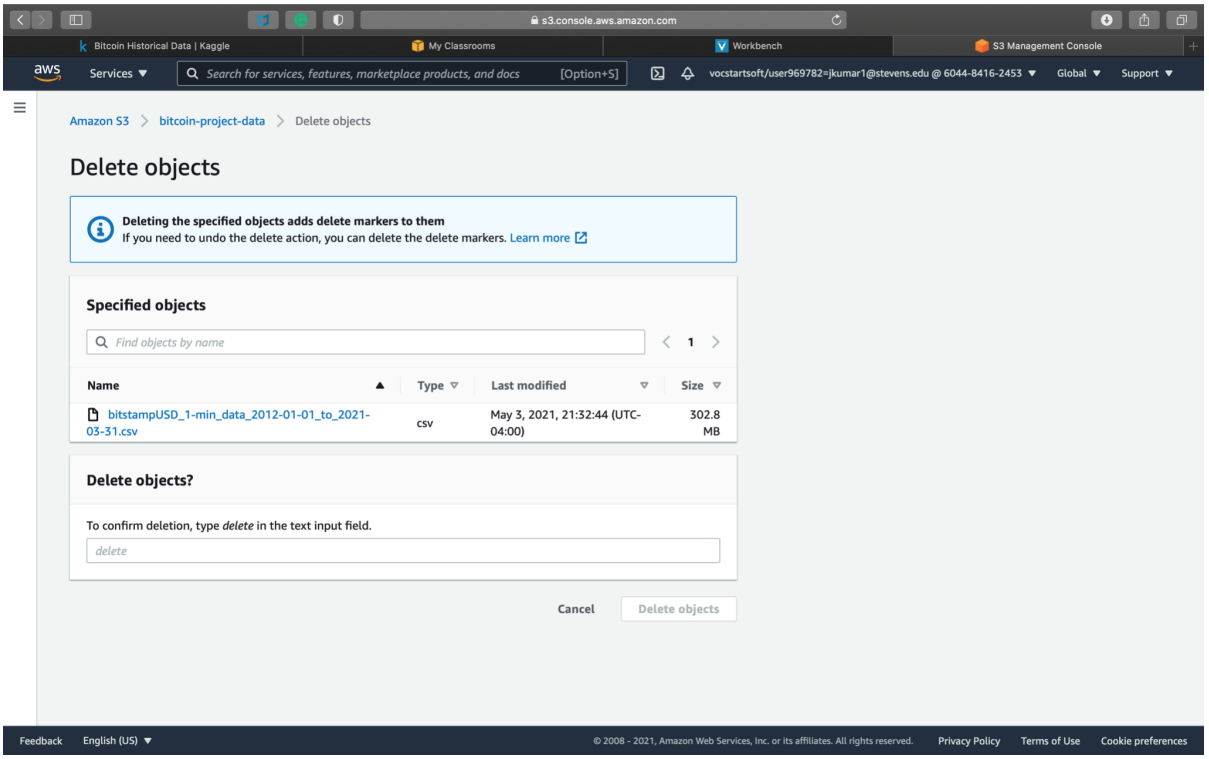
As it can be seen above the RNN model was not nearly as effective as the LSTM model. This was for a simple reason. As RNN trains further and further it loses memory of early training rounds. This means that it loses context for data and causes inaccurate predictions

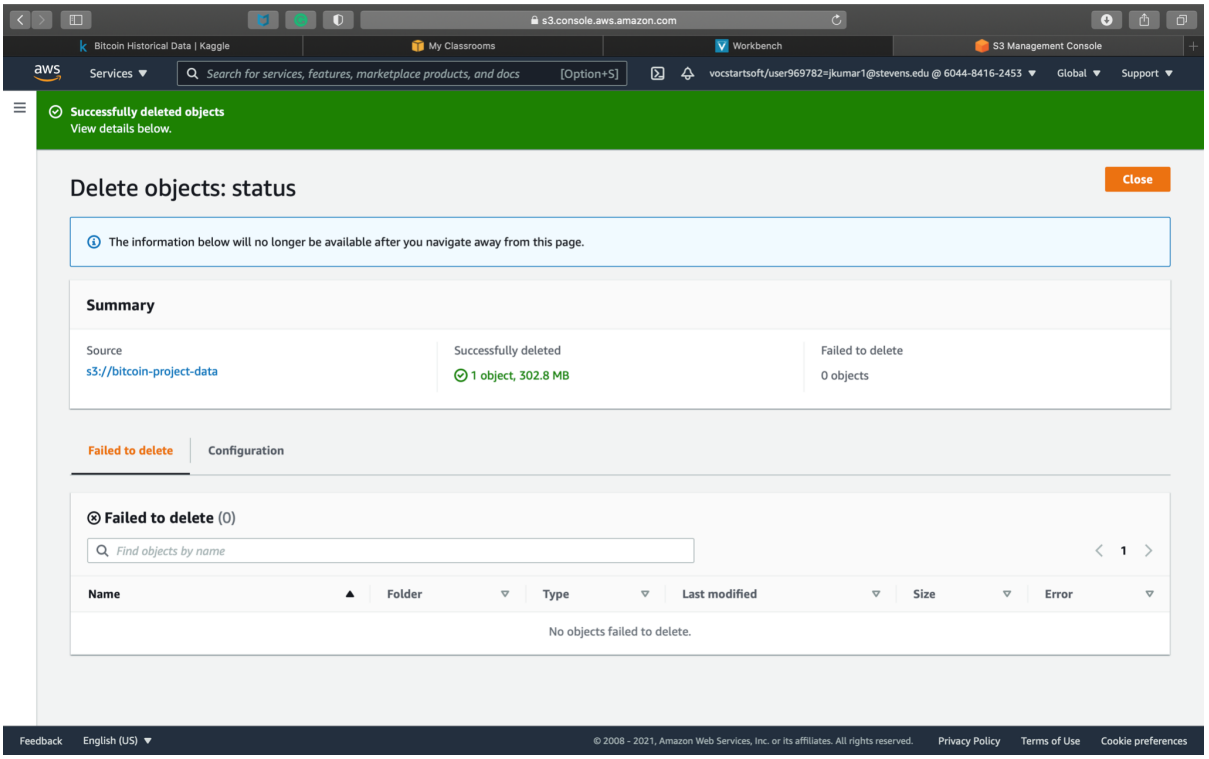
Deletion of Environment:

Deleting S3 bucket and AWS Sagemaker:

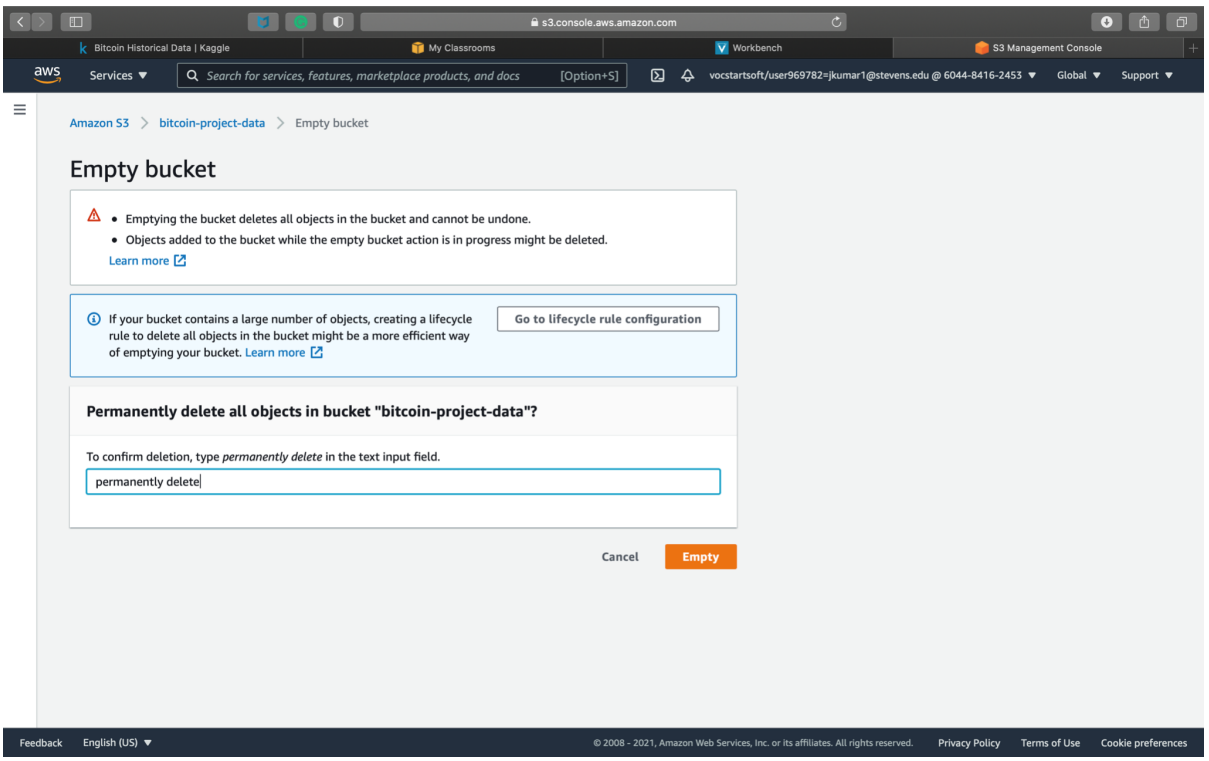
Deleting the dataset and emptying bucket for deletion:

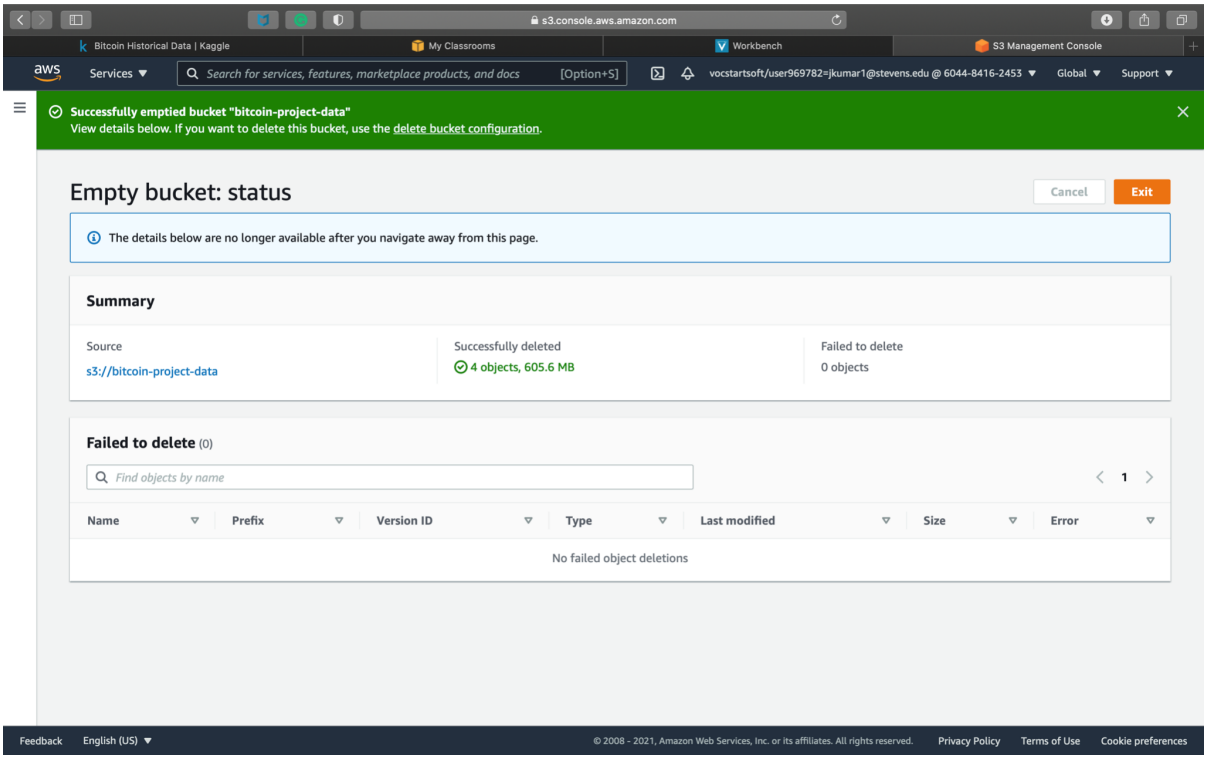


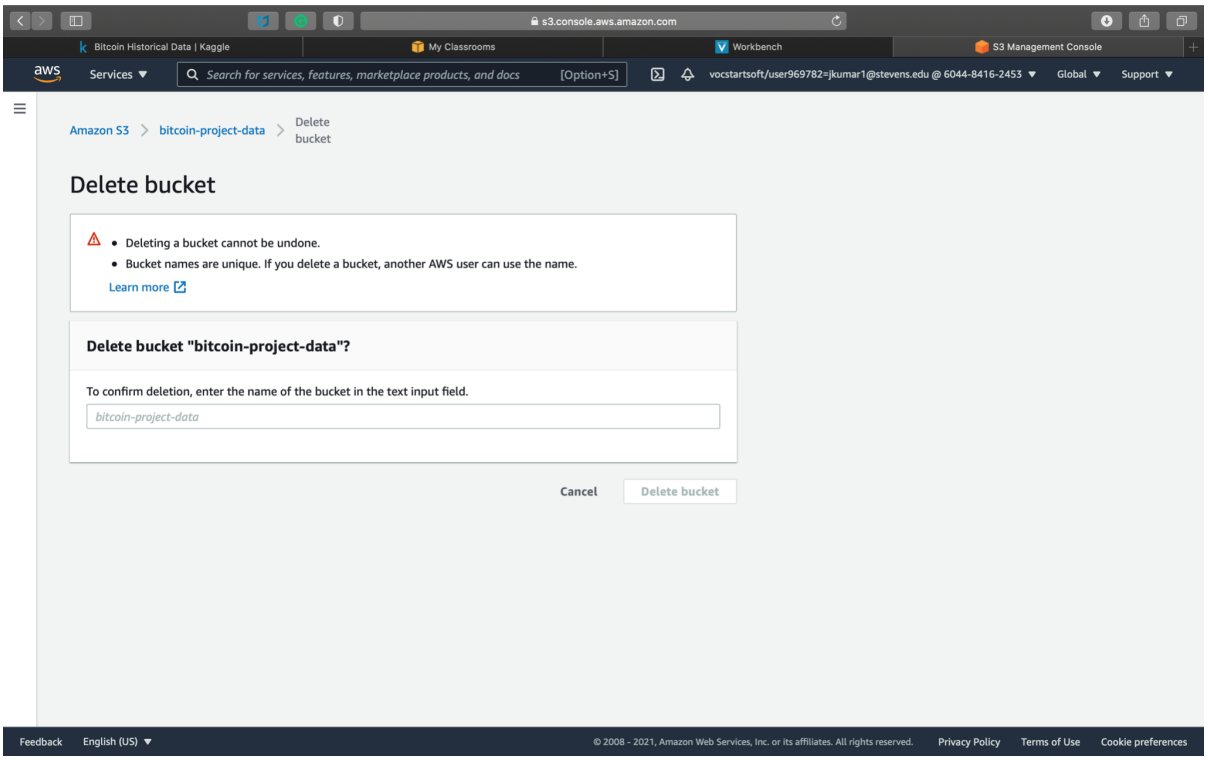


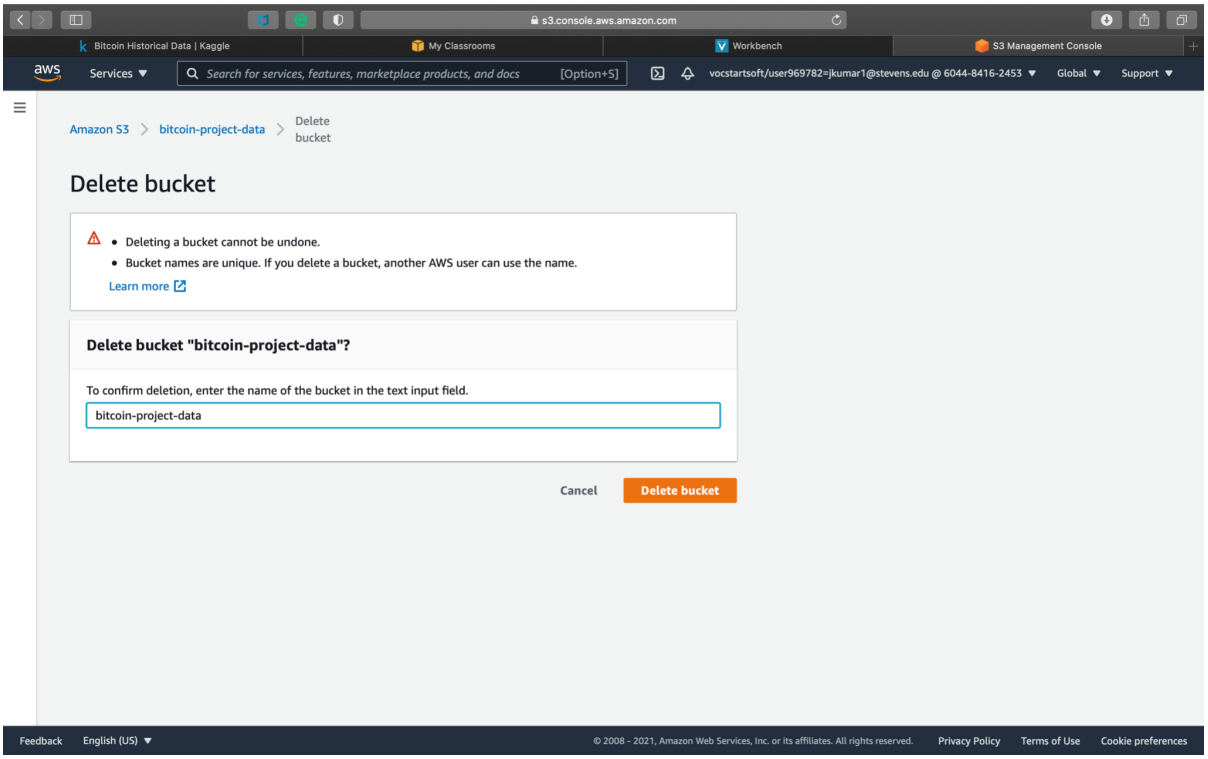


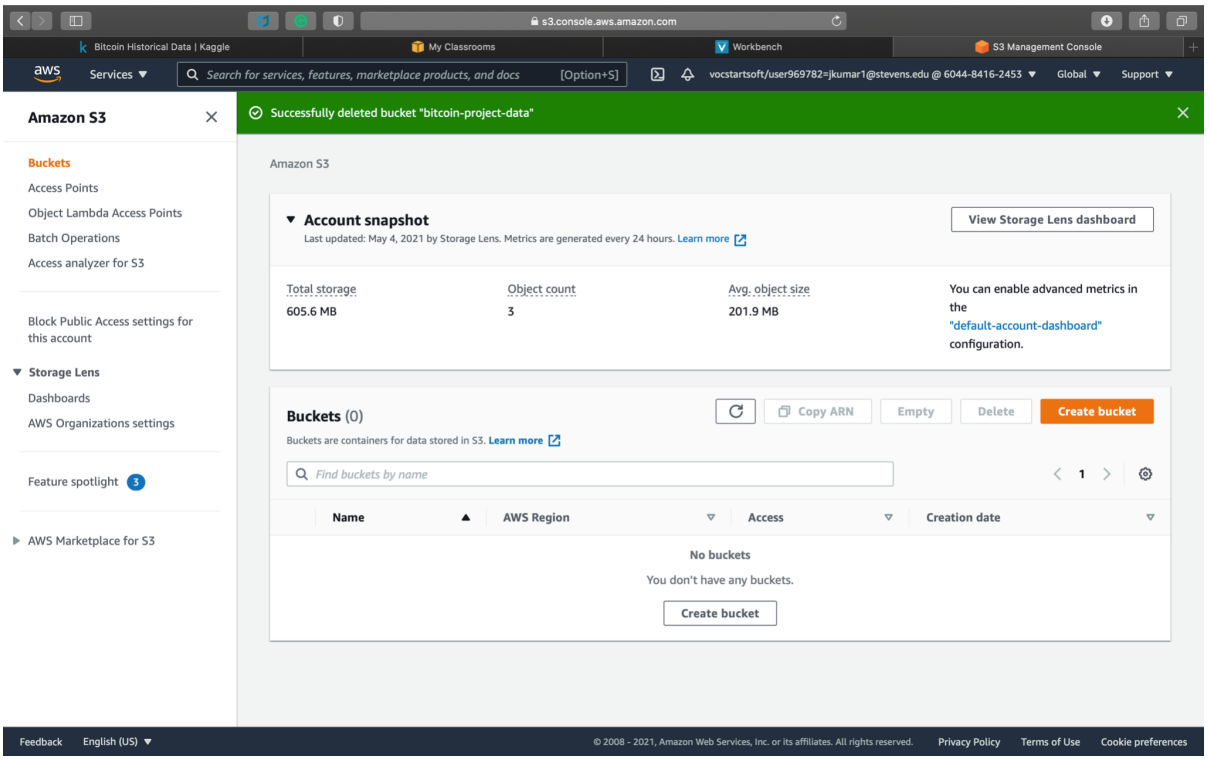
Emptying bucket:



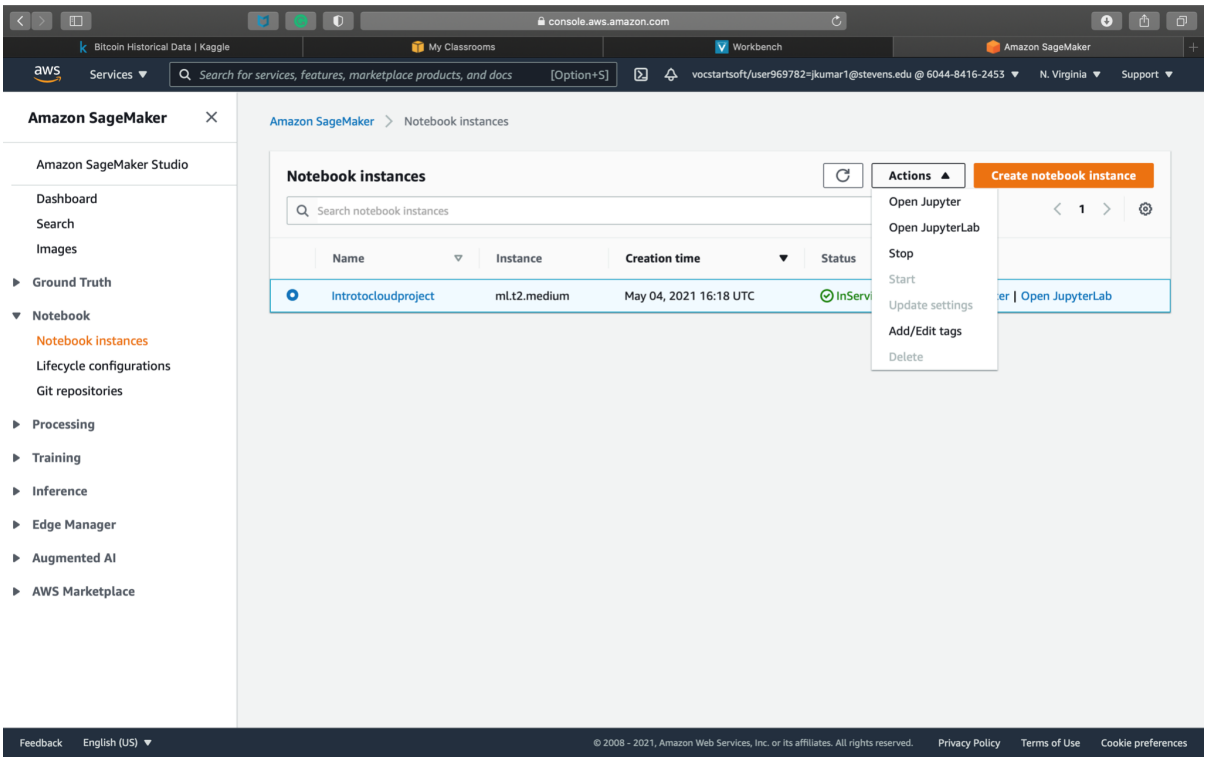

s3.console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
S3 Management Console
Services
Search for services, features, marketplace products, and docs
[Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
Global
Support
Successfully emptied bucket "bitcoin-project-data"
View details below. If you want to delete this bucket, use the delete bucket configuration.
Empty bucket: status
Cancel
Exit
The details below are no longer available after you navigate away from this page.
Summary
Source
s3://bitcoin-project-data
Successfully deleted
4 objects, 605.6 MB
Failed to delete
0 objects
Failed to delete (0)
Find objects by name
1
Name
Prefix
Version ID
Type
Last modified
Size
Error
No failed object deletions
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

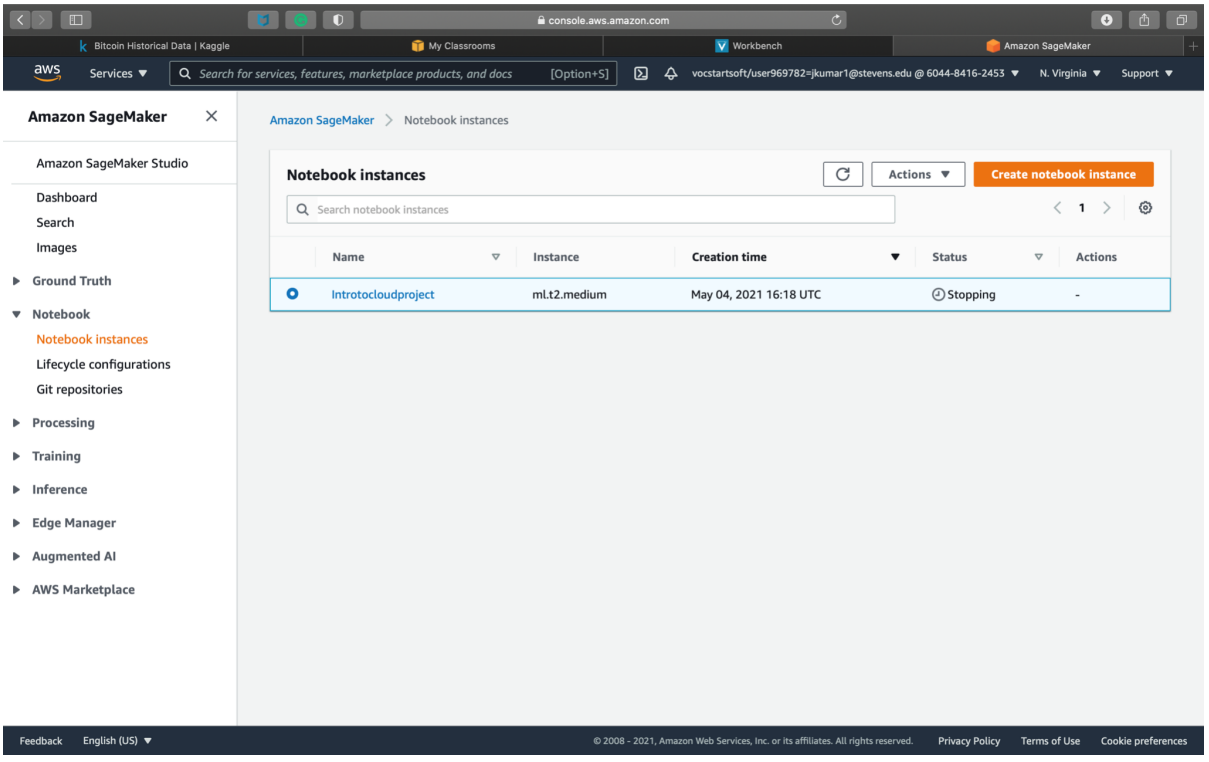

s3.console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
S3 Management Console
Services
Search for services, features, marketplace products, and docs
[Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
Global
Support
Amazon S3
bitcoin-project-data
Delete bucket
Delete bucket
Deleting a bucket cannot be undone.
Bucket names are unique. If you delete a bucket, another AWS user can use the name.
Learn more
Delete bucket "bitcoin-project-data"?
To confirm deletion, enter the name of the bucket in the text input field.
bitcoin-project-data
Cancel
Delete bucket
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


s3.console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
S3 Management Console
Services
Search for services, features, marketplace products, and docs
[Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
Global
Support
Amazon S3 > bitcoin-project-data > Delete bucket
Delete bucket
Deleting a bucket cannot be undone.
Bucket names are unique. If you delete a bucket, another AWS user can use the name.
Learn more
Delete bucket "bitcoin-project-data"?
To confirm deletion, enter the name of the bucket in the text input field.
bitcoin-project-data
Cancel
Delete bucket
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

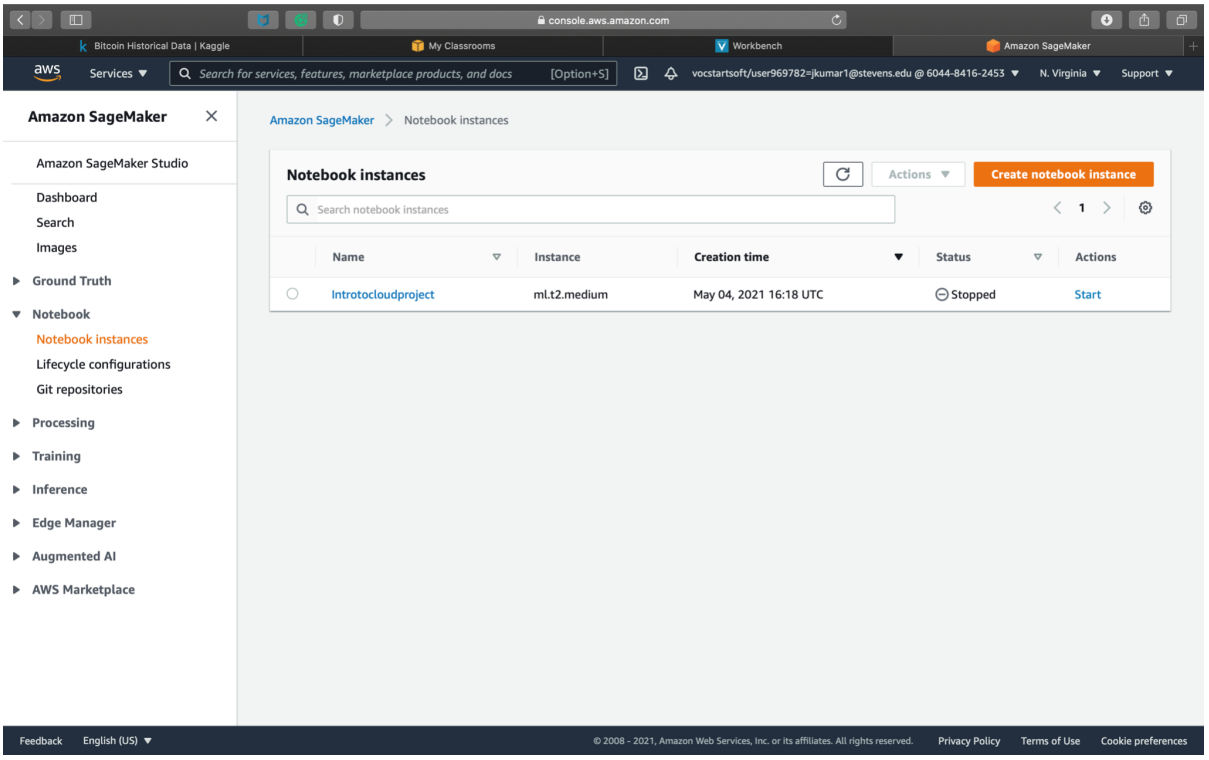

s3.console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
S3 Management Console
Services
Search for services, features, marketplace products, and docs
[Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
Global
Support
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Batch Operations
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 deleted bucket "bitcoin-project-data"
Amazon S3
Account snapshot
Last updated: May 4, 2021 by Storage Lens. Metrics are generated every 24 hours. Learn more
View Storage Lens dashboard
Total storage
605.6 MB
Object count
3
Avg. object size
201.9 MB
You can enable advanced metrics in the "default-account-dashboard" configuration.
Buckets (0)
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
No buckets
You don't have any buckets.
Create bucket
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

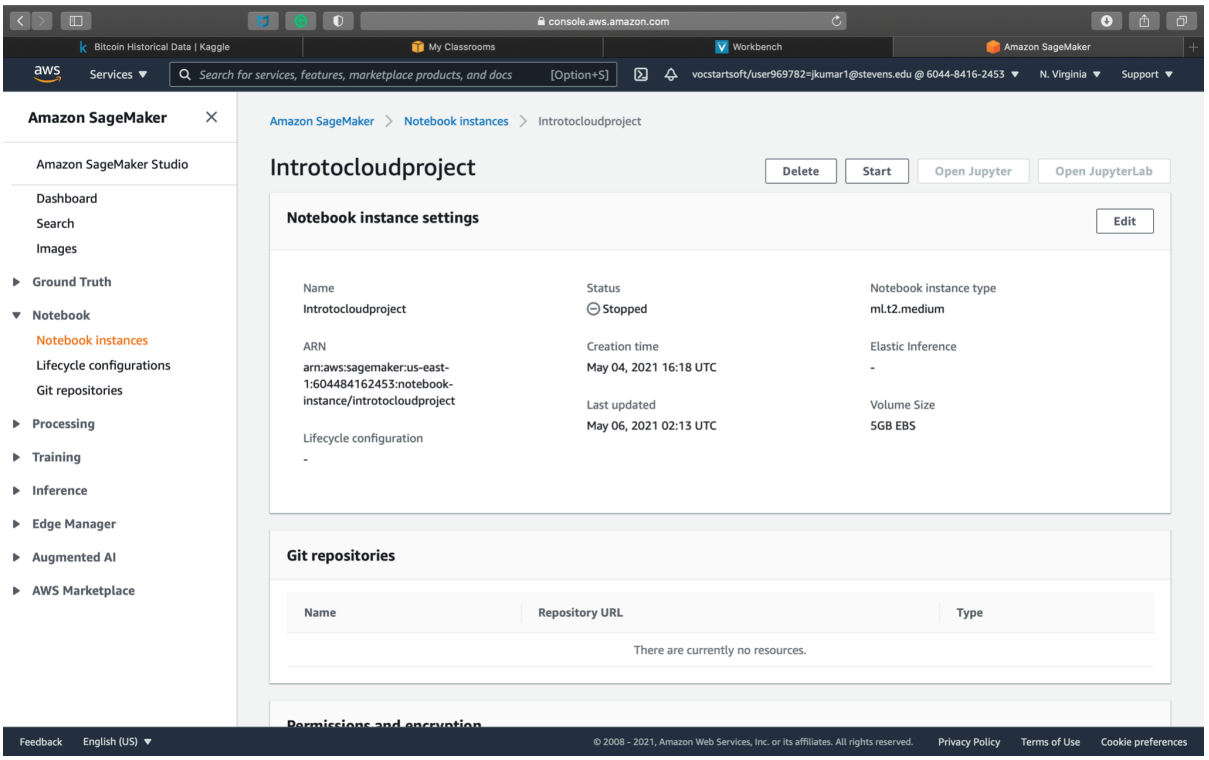
Deleting the AWS sagemaker notebook instance:

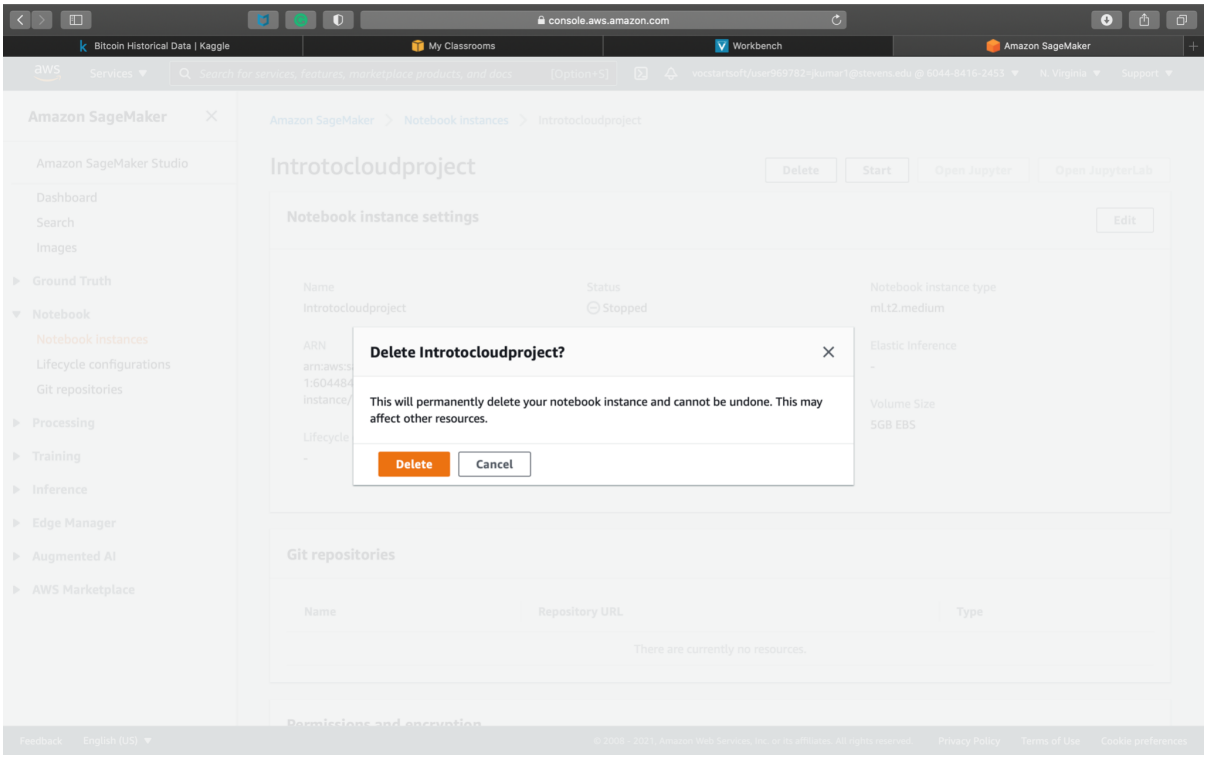


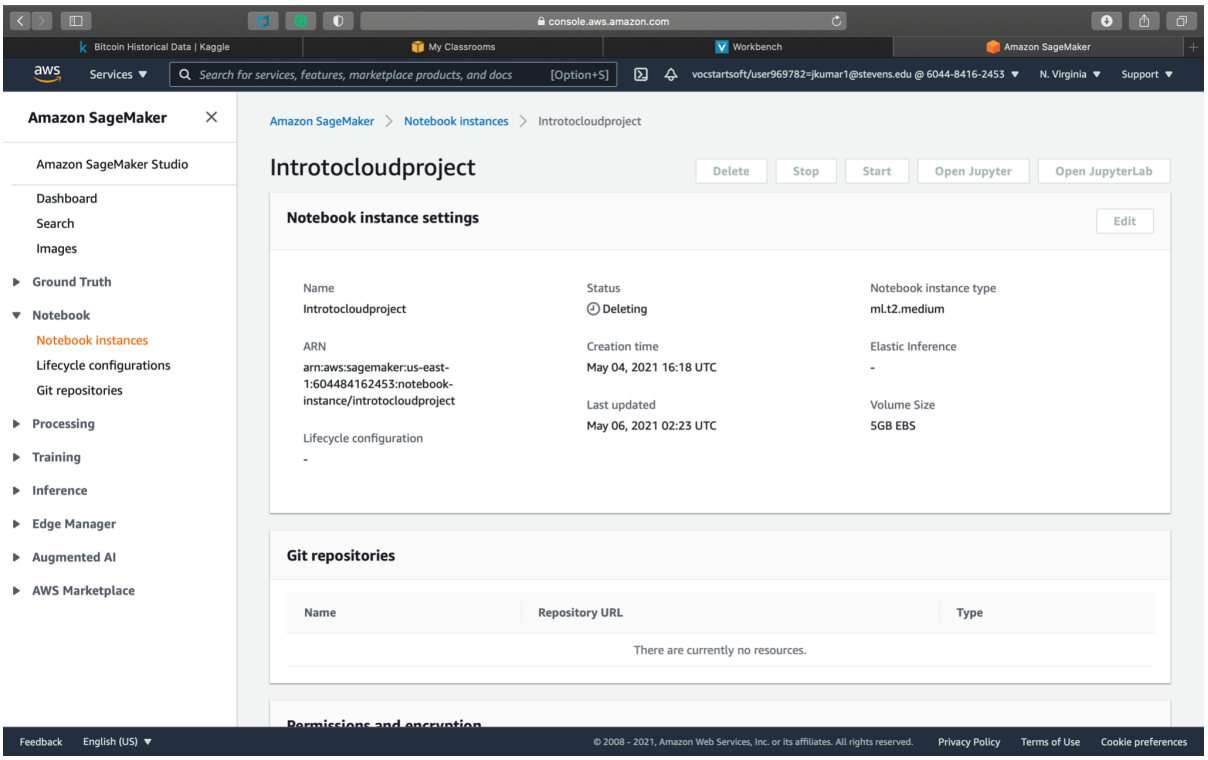


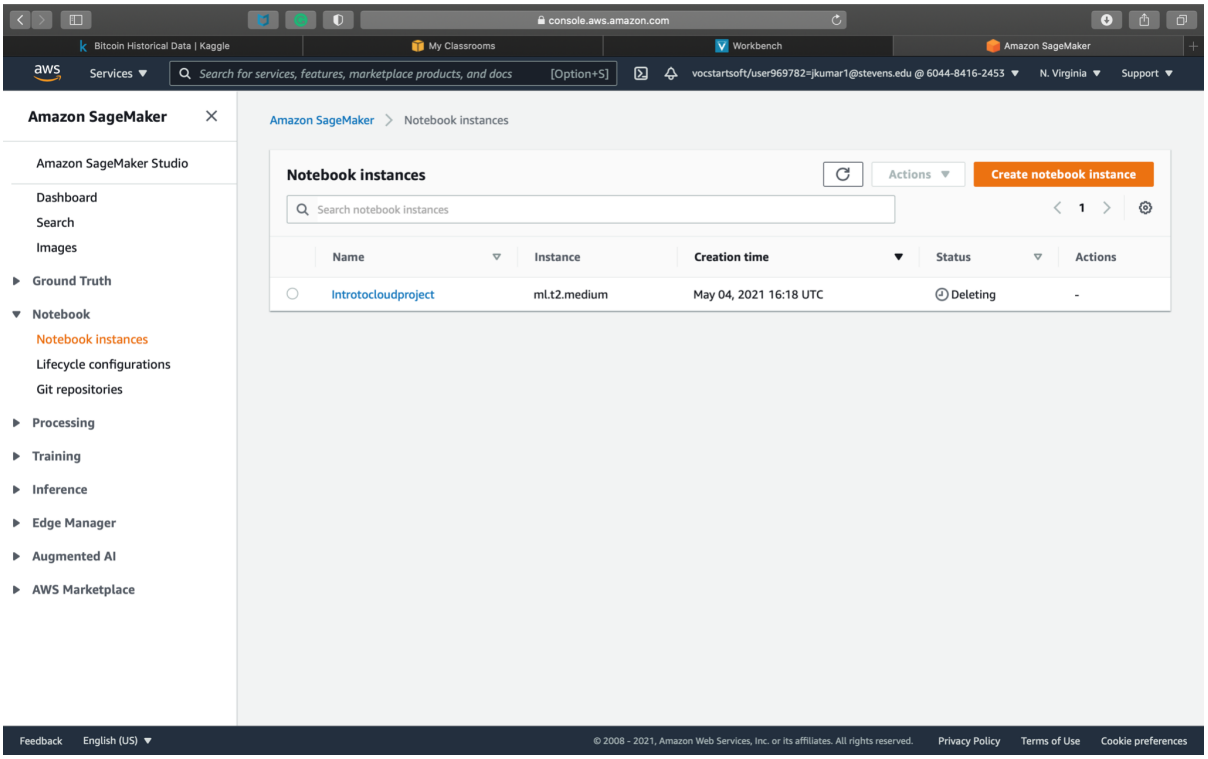
- Waiting for instance to stop and then we can delete the instance:

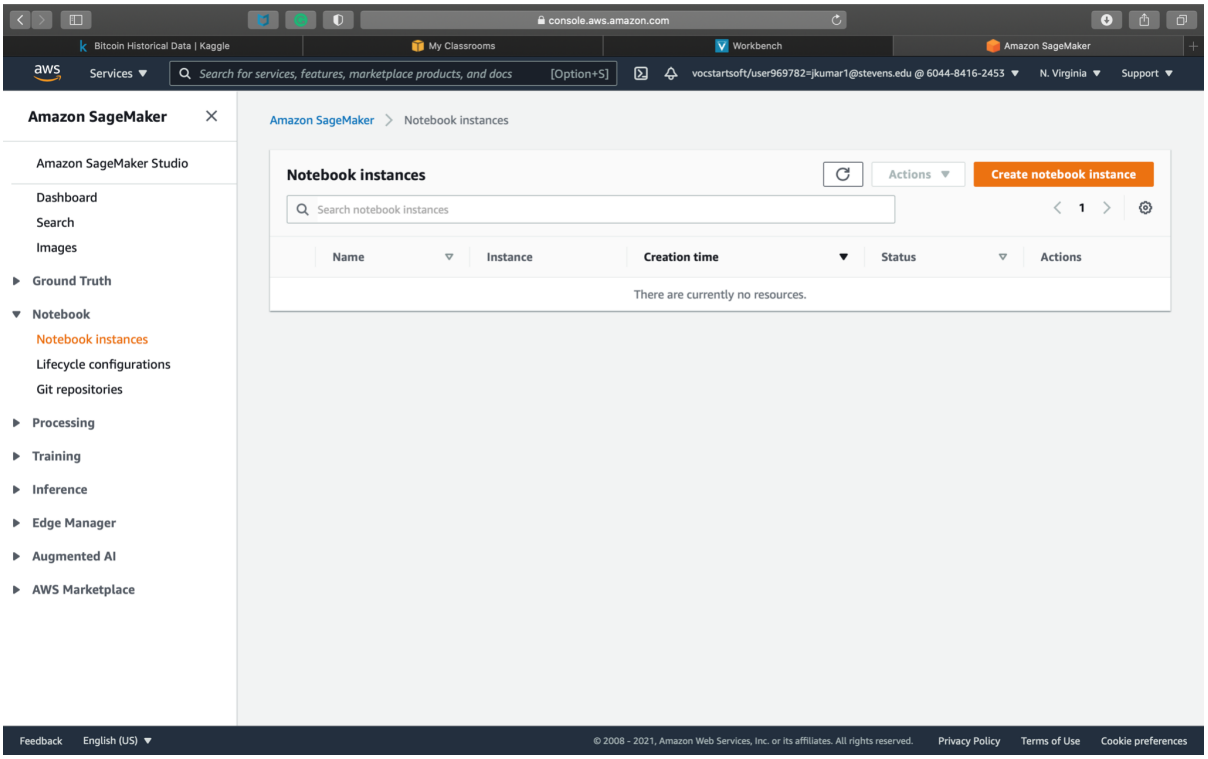




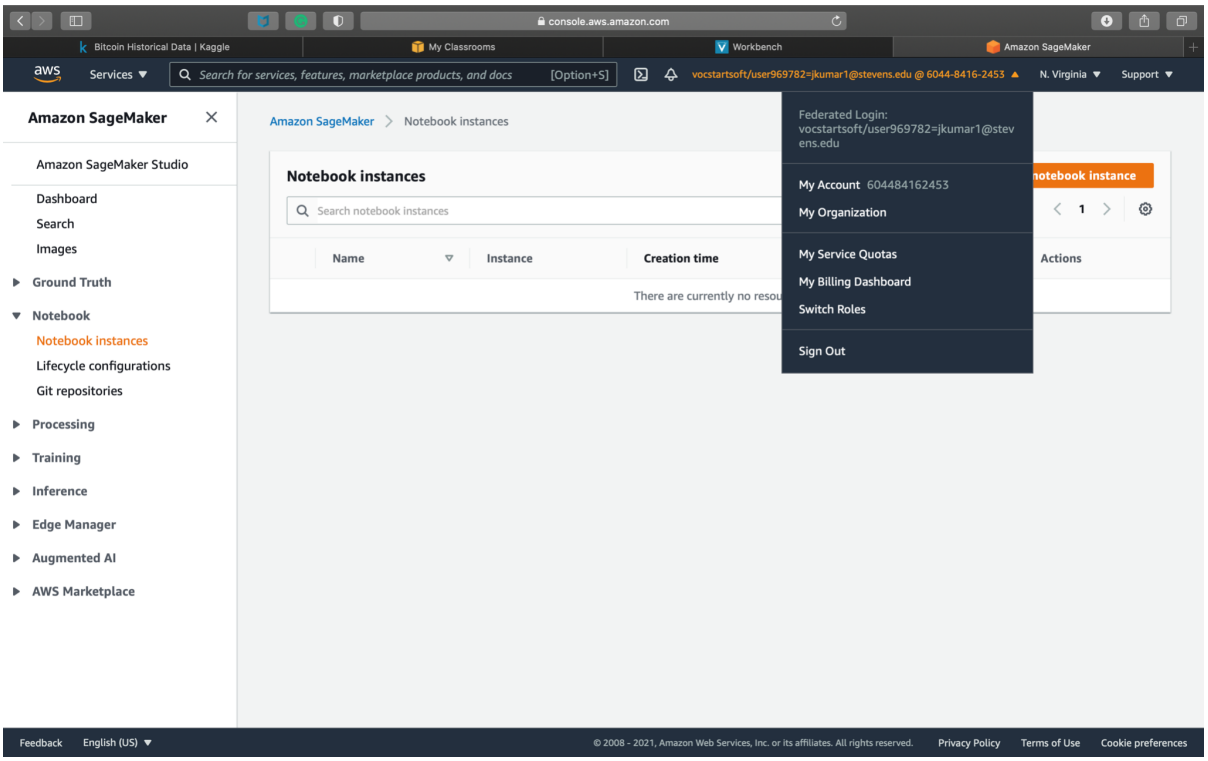

console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
Amazon SageMaker
Amazon SageMaker
Introtocloudproject
Delete
Start
Open Jupyter
Open JupyterLab
Notebook instance settings
Edit
Delete Introtocloudproject?
This will permanently delete your notebook instance and cannot be undone. This may affect other resources.
Delete
Cancel
Git repositories
There are currently no resources.


console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
Amazon SageMaker
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
N. Virginia
Support
Amazon SageMaker
Amazon SageMaker Studio
Dashboard
Search
Images
Ground Truth
Notebook
Notebook instances
Lifecycle configurations
Git repositories
Processing
Training
Inference
Edge Manager
Augmented AI
AWS Marketplace
Amazon SageMaker > Notebook instances > Introtocloudproject
Introtocloudproject
Delete
Stop
Start
Open Jupyter
Open JupyterLab
Notebook instance settings
Edit
Name
Introtocloudproject
Status
Deleting
Notebook instance type
ml.t2.medium
ARN
arn:aws:sagemaker:us-east-1:604484162453:notebook-instance/introtocloudproject
Creation time
May 04, 2021 16:18 UTC
Elastic Inference
-
Lifecycle configuration
-
Last updated
May 06, 2021 02:23 UTC
Volume Size
5GB EBS
Git repositories
Name
Repository URL
Type
There are currently no resources.
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
Amazon SageMaker
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
N. Virginia
Support
Amazon SageMaker
Amazon SageMaker Studio
Dashboard
Search
Images
Ground Truth
Notebook
Notebook instances
Lifecycle configurations
Git repositories
Processing
Training
Inference
Edge Manager
Augmented AI
AWS Marketplace
Amazon SageMaker > Notebook instances
Notebook instances
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
Introtocloudproject
ml.t2.medium
May 04, 2021 16:18 UTC
Deleting
-
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
Amazon SageMaker
Services
Search for services, features, marketplace products, and docs [Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
N. Virginia
Support
Amazon SageMaker
Amazon SageMaker Studio
Dashboard
Search
Images
Ground Truth
Notebook
Notebook instances
Lifecycle configurations
Git repositories
Processing
Training
Inference
Edge Manager
Augmented AI
AWS Marketplace
Amazon SageMaker > Notebook instances
Notebook instances
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
There are currently no resources.
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Logging out:


console.aws.amazon.com
Bitcoin Historical Data | Kaggle
My Classrooms
Workbench
Amazon SageMaker
aws
Services
Search for services, features, marketplace products, and docs
[Option+S]
vocstartsoft/user969782=jkumar1@stevens.edu @ 6044-8416-2453
N. Virginia
Support
Amazon SageMaker
Amazon SageMaker Studio
Dashboard
Search
Images
Ground Truth
Notebook
Notebook instances
Lifecycle configurations
Git repositories
Processing
Training
Inference
Edge Manager
Augmented AI
AWS Marketplace
Amazon SageMaker > Notebook instances
Notebook instances
Search notebook instances
Name
Instance
Creation time
Actions
Federated Login: vocstartsoft/user969782=jkumar1@stevens.edu
My Account 604484162453
My Organization
My Service Quotas
My Billing Dashboard
Switch Roles
Sign Out
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Conclusion:

We have used AWS S3 bucket for storage of data, AWS IAM for encryption, AWS sagemaker for creating the project and using the Deep learning libraries, AWS sagemaker Jupyter notebook instance for creating Python files and training our model with variousDeep learning algorithms.

As we see LSTM and RNN diagrams on Page 20 and 21, the LSTM is all the more firmly related with the real value expectation than the RNN model. At the point when the preparation information spreads in reverse then it gets more vulnerable and more fragile, when it gets to those neurons that address more seasoned information focused presently arrangement it has no juice to change them appropriately. This issue is called Vanishing Gradient.

LSTM is also one of the kind or say extensions of RNN.The difference is that it keeps the vital data around the past and overlooks the insignificant data. It won't be devoured by pointless data when gradient back-propagates. We are able to increment the number of epochs to refine our show execution and it also improves the prediction of close value of Bitcoin price results overall.

We can say that, when we move from RNN to LSTM (Long Short-Term Memory), we are presenting more and additional controlling handles, which control the stream and blending of Inputs according to prepared Weights. What's more, hence, acquiring greater adaptability in controlling the yields. Along these lines, **LSTM gives us the most Control-capacity and hence, Better Results and when used with the AWS S3 bucket provides faster data fetching and AWS sagemaker libraries gives faster computation.**

AWS Sagemaker has been an incredible arrangement for our predictive analysis algorithm to achieve a really start to finish ML arrangement. It dealt with abstracting a huge load of programming improvement abilities important to achieve the errand while being exceptionally powerful and adaptable and practical. Above all, it helped us focus on the core Deep Learning experiments and supplements the remaining necessary skills with easy abstracted tools similar to our existing workflow.

References:

1. https://www.kaggle.com/mczielinski/bitcoin-historical-datahttps://aws.amazon.com/blogs/machine-learning/training-debugging-and-running-time-series-forecasting-models-with-the-gluonts-toolkit-on-amazon-sagemaker/
2. https://github.com/aws/amazon-sagemaker-examples/tree/master/introduction_to_amazon_algorithms
3. https://www.linkedin.com/learning/learning-amazon-sagemaker?trk=share_ios_course_learning&shareId=A769egtZRB6UhHuZGBIF6A
4. https://www.linkedin.com/learning/learning-bitcoin-and-other-cryptocurrencies/how-bitcoin-works?u=56742337