<table>
<tr><td colspan="3">**Name of Student: Pushkar Sane**</td></tr>
<tr><td>**Roll Number: 45**</td><td colspan="2">**Lab Assignment Number: 5 & 6**</td></tr>
<tr><td colspan="3">**Title of Lab Assignment: Feature Engineering, one hot Encoding, Normalization, Standardization, EDA using SageMaker DataWrangler Linear and Multiple Linear Regression using SageMaker.**</td></tr>
<tr><td>**DOP: 16-03-2024**</td><td colspan="2">**DOS: 16-03-2024**</td></tr>
<tr><td>**CO Mapped:**
CO3</td><td>**PO Mapped:**
PO2, PO3, PO4, PO5, PO6, PO7, PSO1, PSO2</td><td>**Signature:**</td></tr>
</table>

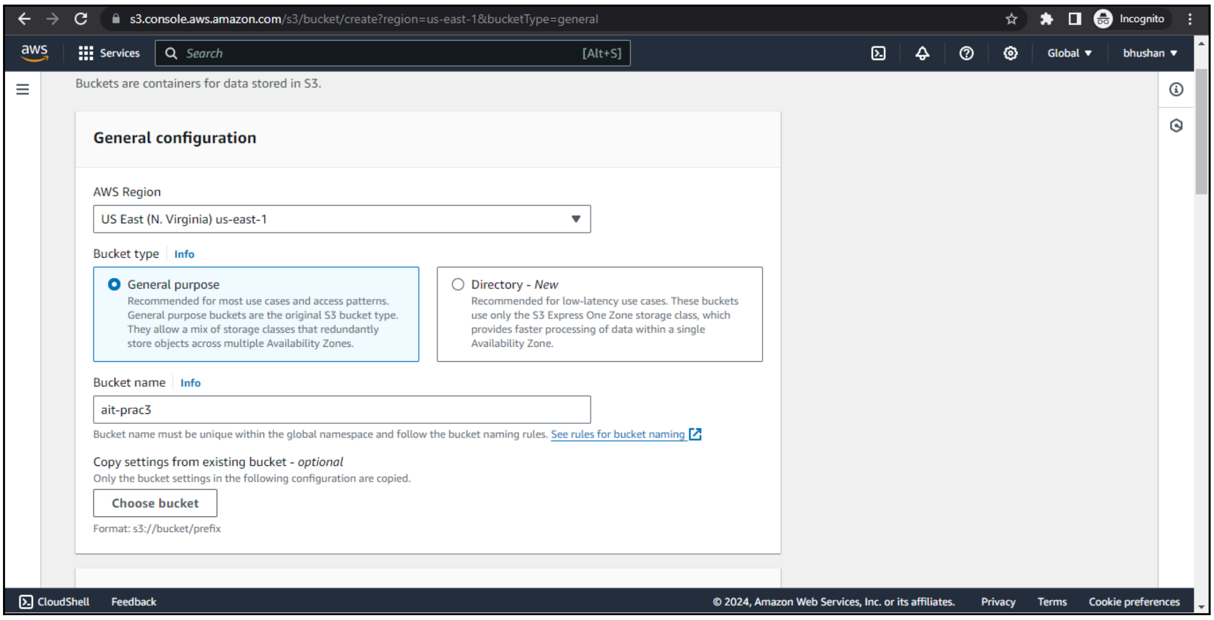
Practical 5 & 6

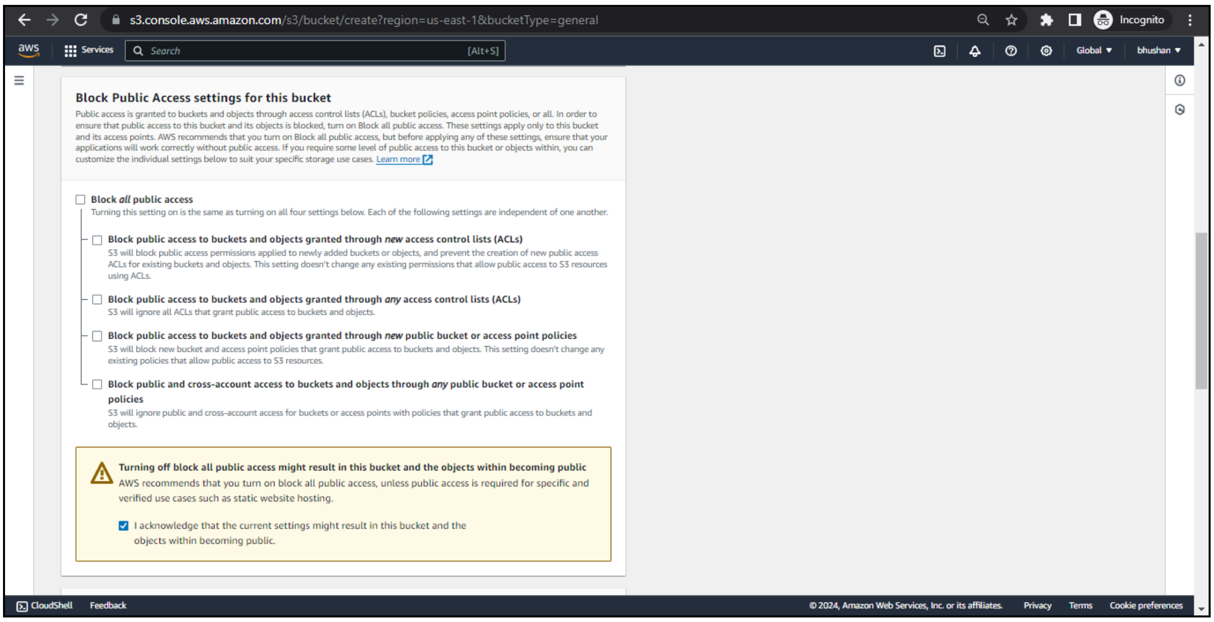
Aim: Feature Engineering, one hot Encoding, Normalization, Standardization, EDA using SageMaker DataWrangler Linear and Multiple Linear Regression using SageMake**r.**

Description:

Prerequisite

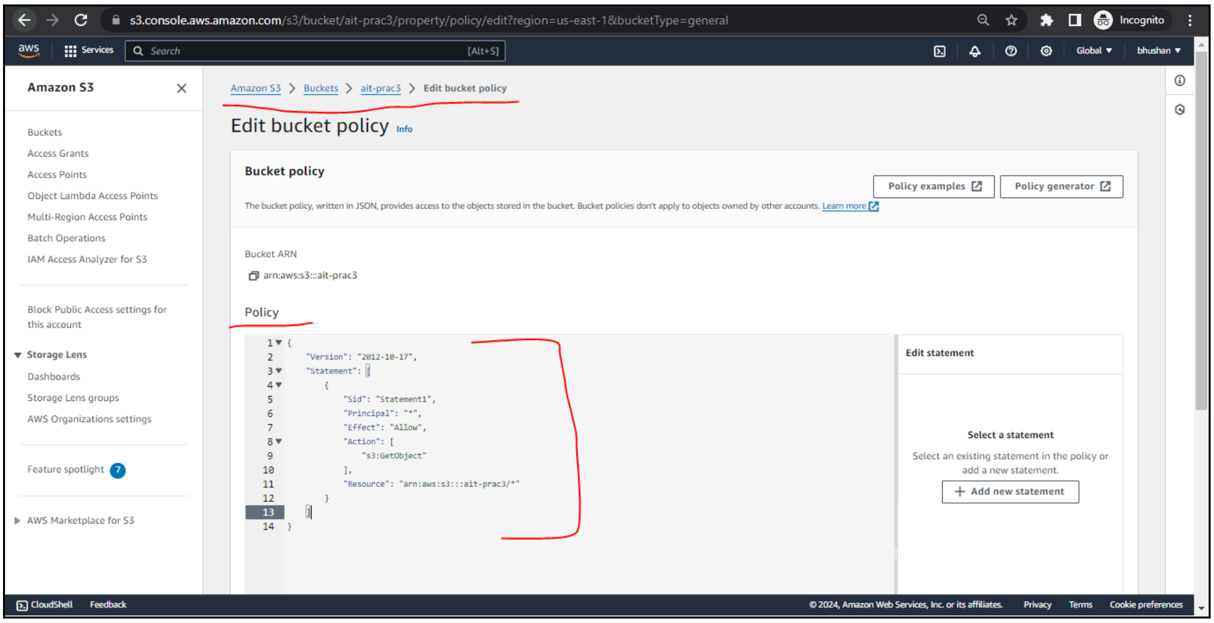
Create an S3 bucket and keep aside (it will be used later)



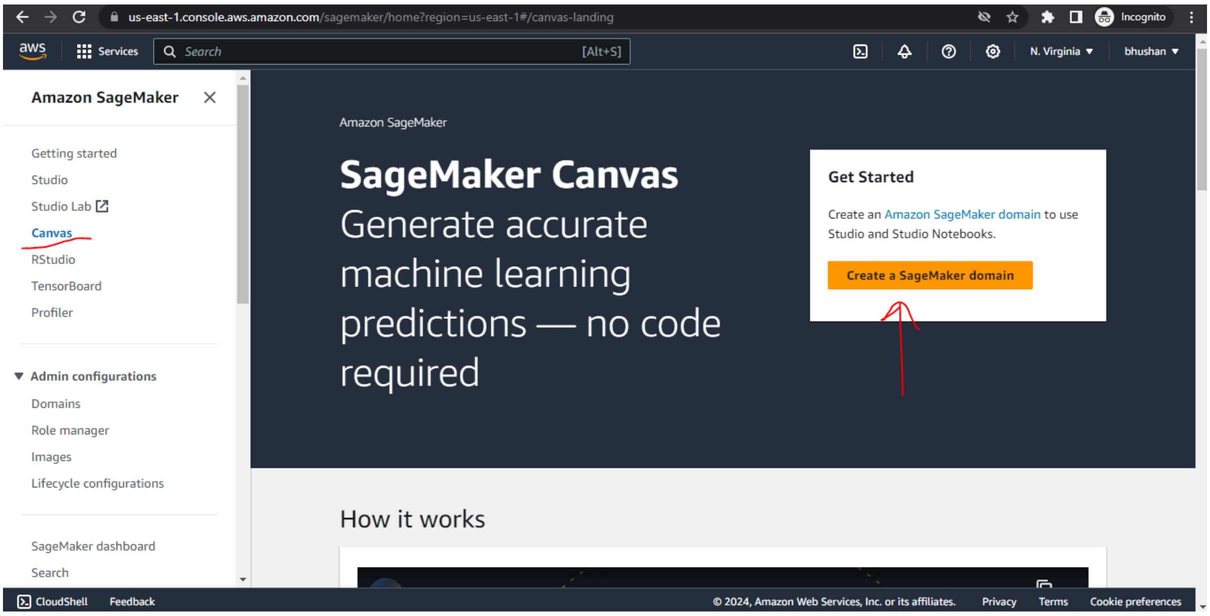


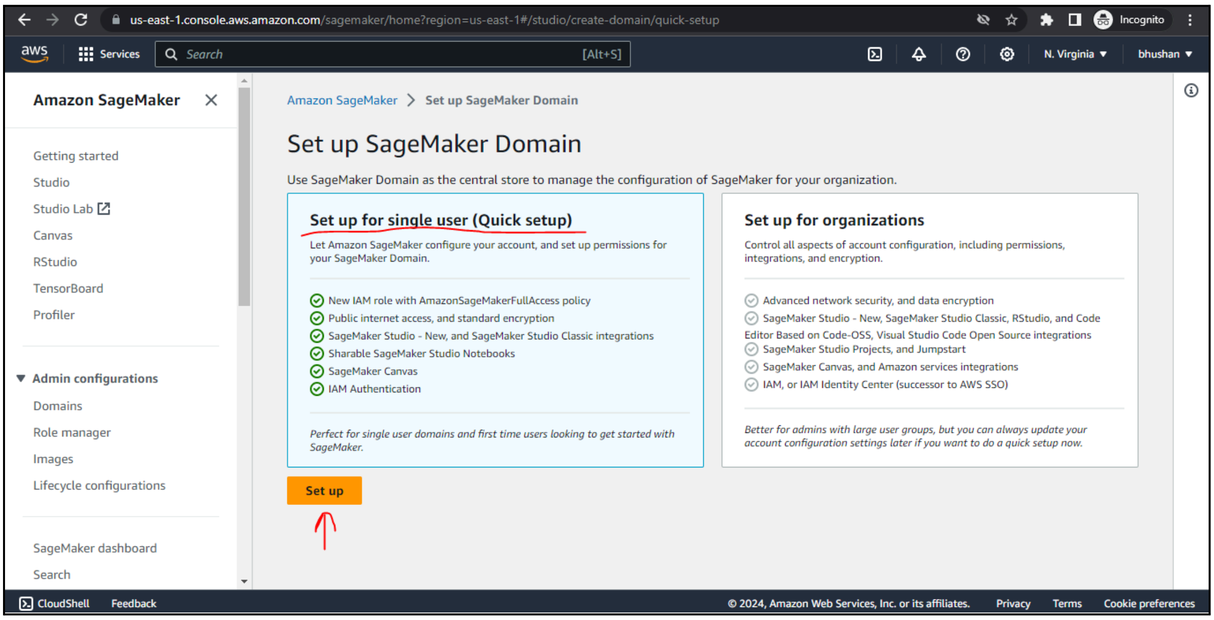
Click on “create” to create the bucket

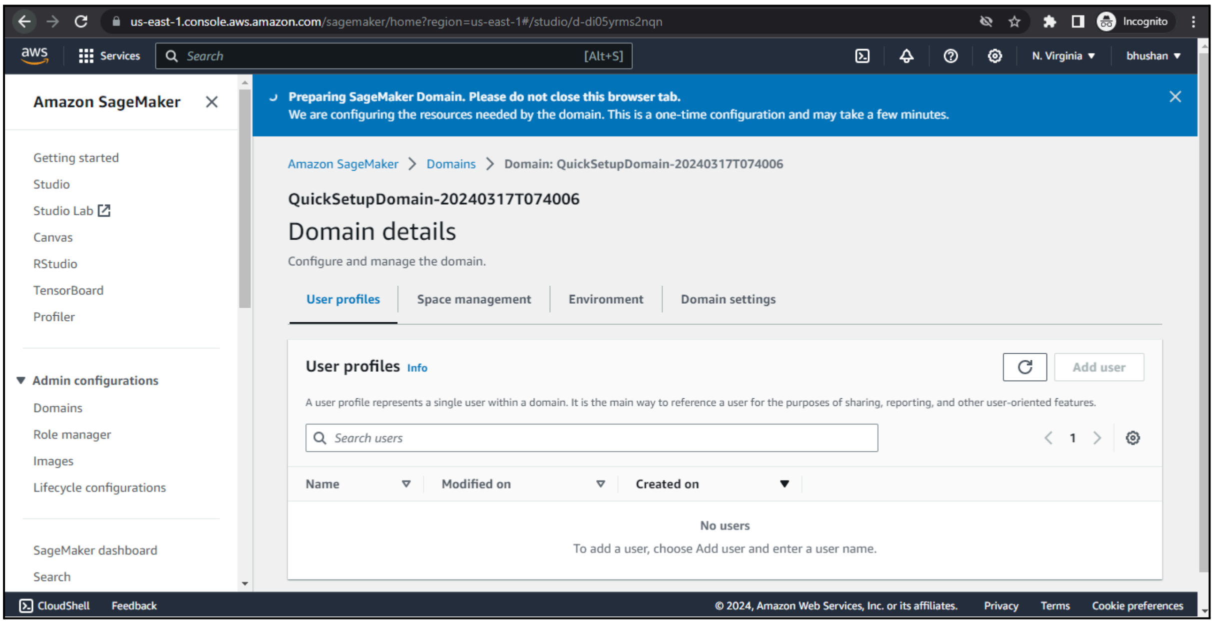
Change the permissions

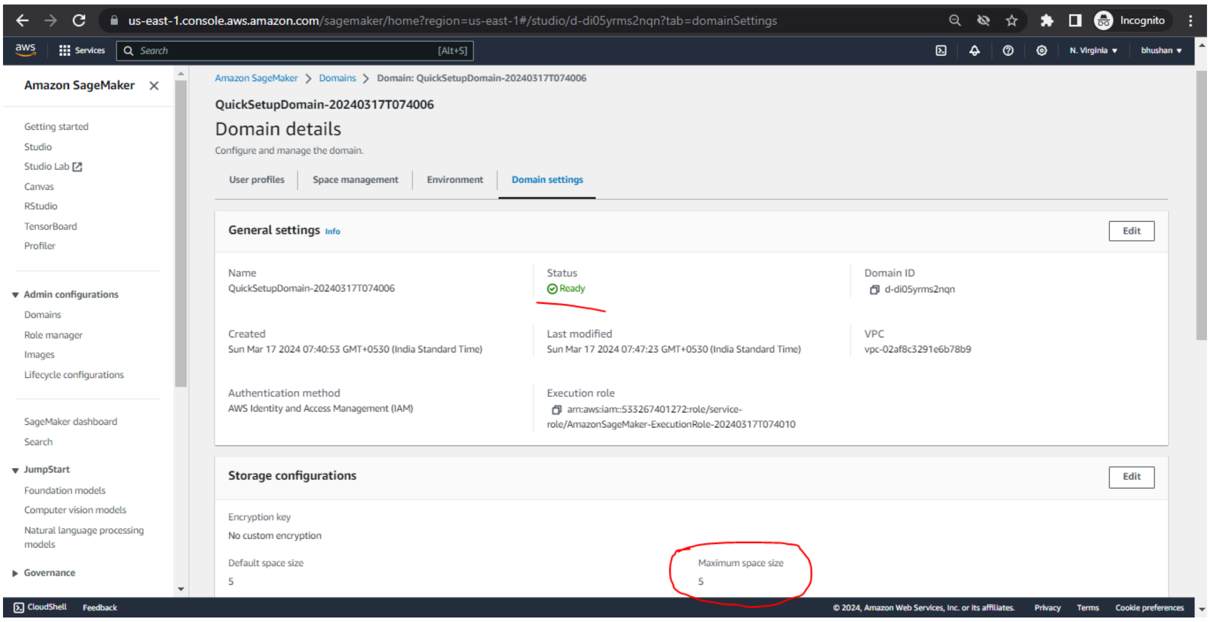


SageMaker Canvas

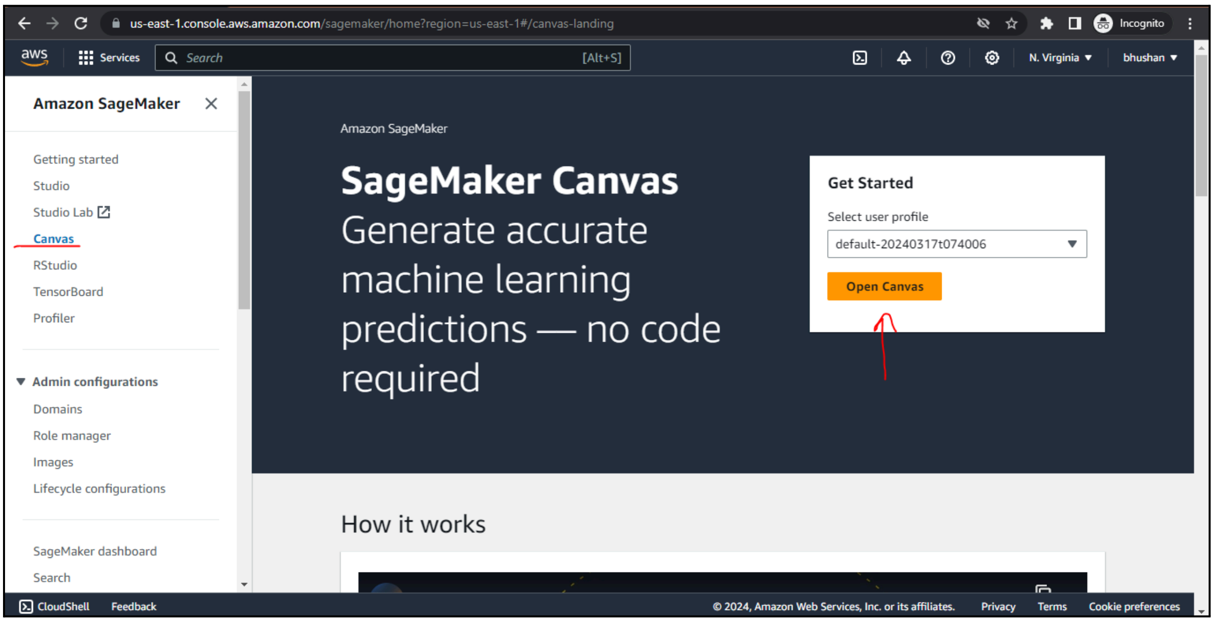



us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/studio/create-domain/quick-setup
Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
TensorBoard
Profiler
Admin configurations
Domains
Role manager
Images
Lifecycle configurations
SageMaker dashboard
Search
Amazon SageMaker > Set up SageMaker Domain
Set up SageMaker Domain
Use SageMaker Domain as the central store to manage the configuration of SageMaker for your organization.
Set up for single user (Quick setup)
Let Amazon SageMaker configure your account, and set up permissions for your SageMaker Domain.
New IAM role with AmazonSageMakerFullAccess policy
Public internet access, and standard encryption
SageMaker Studio - New, and SageMaker Studio Classic integrations
Sharable SageMaker Studio Notebooks
SageMaker Canvas
IAM Authentication
Perfect for single user domains and first time users looking to get started with SageMaker.
Set up for organizations
Control all aspects of account configuration, including permissions, integrations, and encryption.
Advanced network security, and data encryption
SageMaker Studio - New, SageMaker Studio Classic, RStudio, and Code Editor Based on Code-OSS, Visual Studio Code Open Source integrations
SageMaker Studio Projects, and Jumpstart
SageMaker Canvas, and Amazon services integrations
IAM, or IAM Identity Center (successor to AWS SSO)
Better for admins with large user groups, but you can always update your account configuration settings later if you want to do a quick setup now.
Set up
© 2024, Amazon Web Services, Inc. or its affiliates.

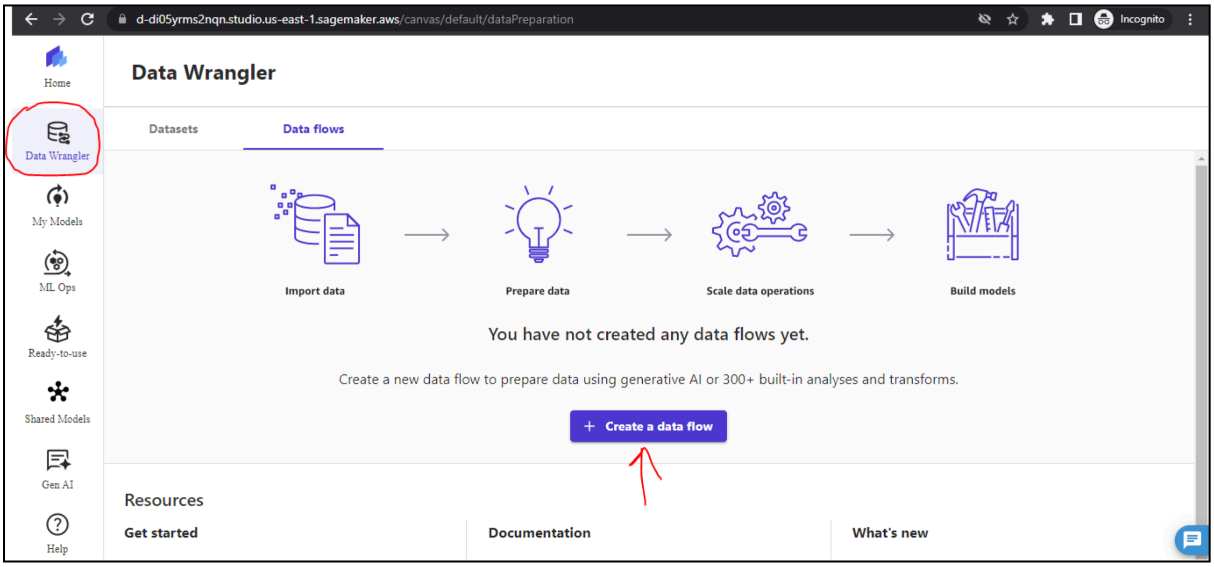

us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/studio/d-di05yrms2nqn
Preparing SageMaker Domain. Please do not close this browser tab.
We are configuring the resources needed by the domain. This is a one-time configuration and may take a few minutes.
Amazon SageMaker > Domains > Domain: QuickSetupDomain-20240317T074006
QuickSetupDomain-20240317T074006
Domain details
Configure and manage the domain.
User profiles
Space management
Environment
Domain settings
User profiles Info
Add user
A user profile represents a single user within a domain. It is the main way to reference a user for the purposes of sharing, reporting, and other user-oriented features.
Search users
Name
Modified on
Created on
No users
To add a user, choose Add user and enter a user name.

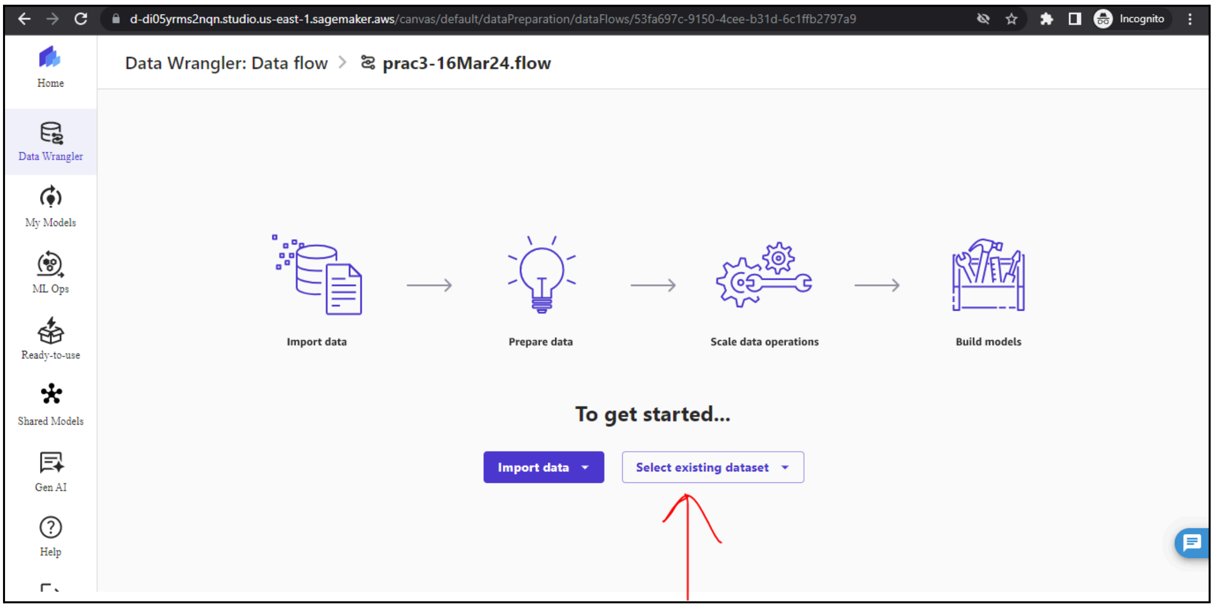


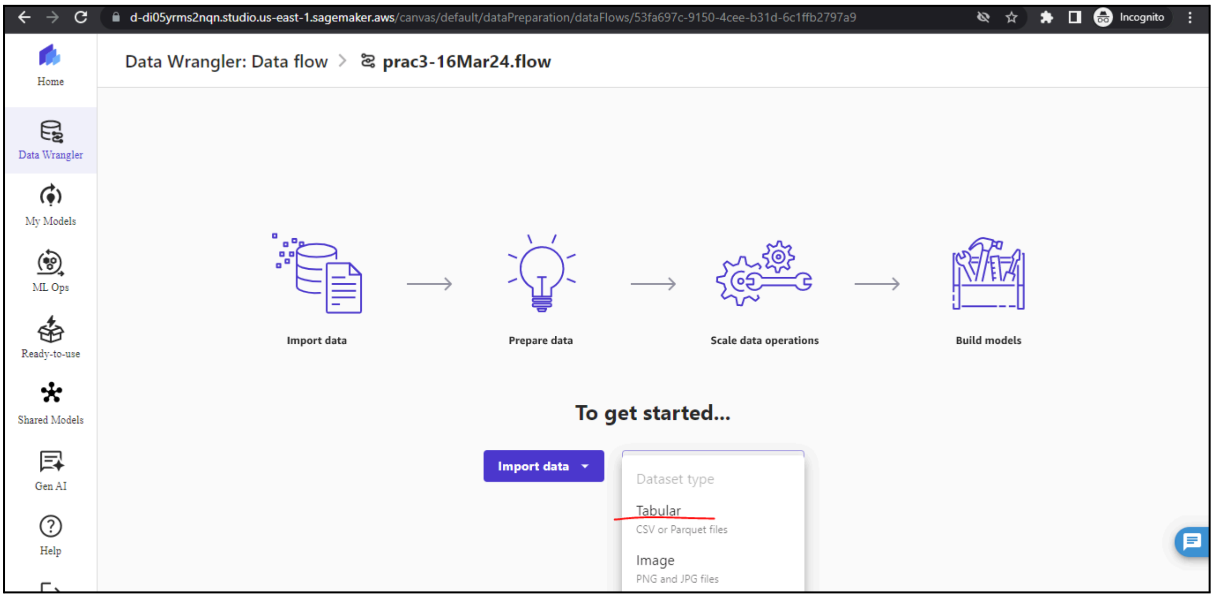
Data Preparation using SageMaker DataWrangler



Import Dataset

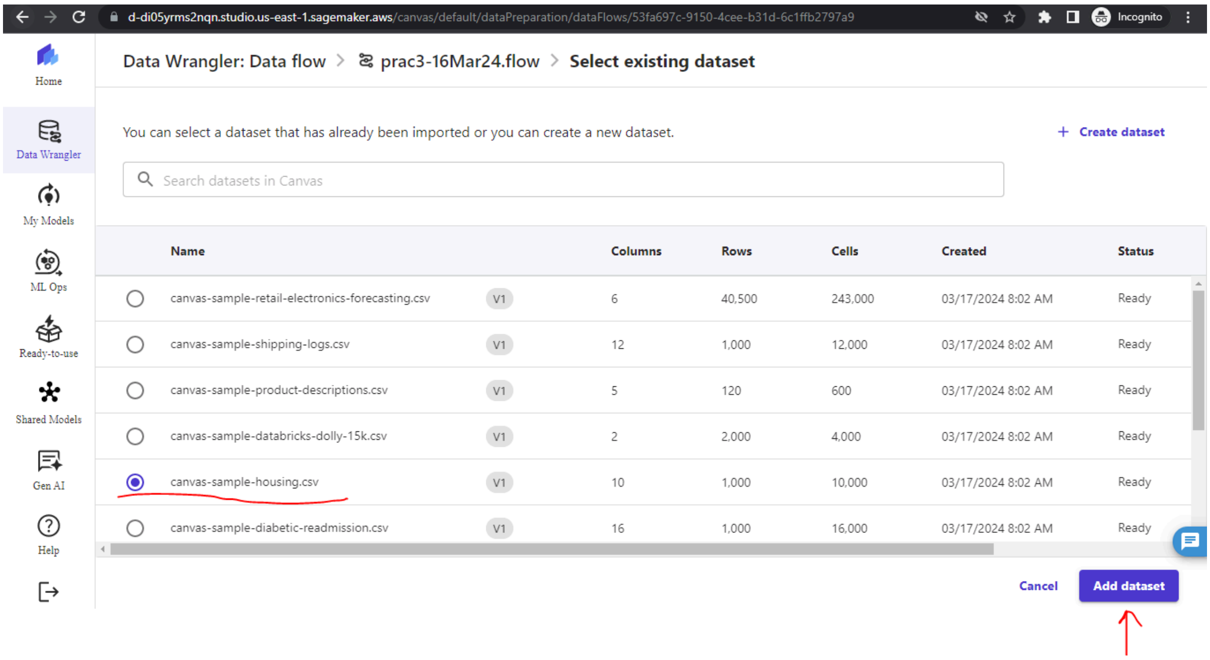




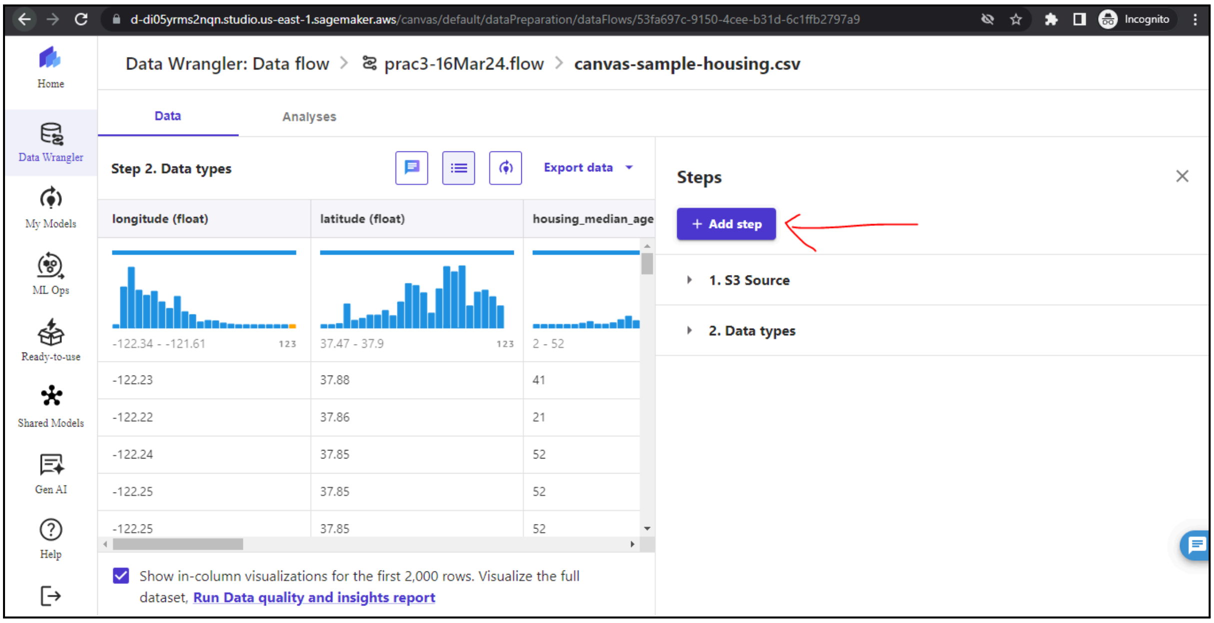


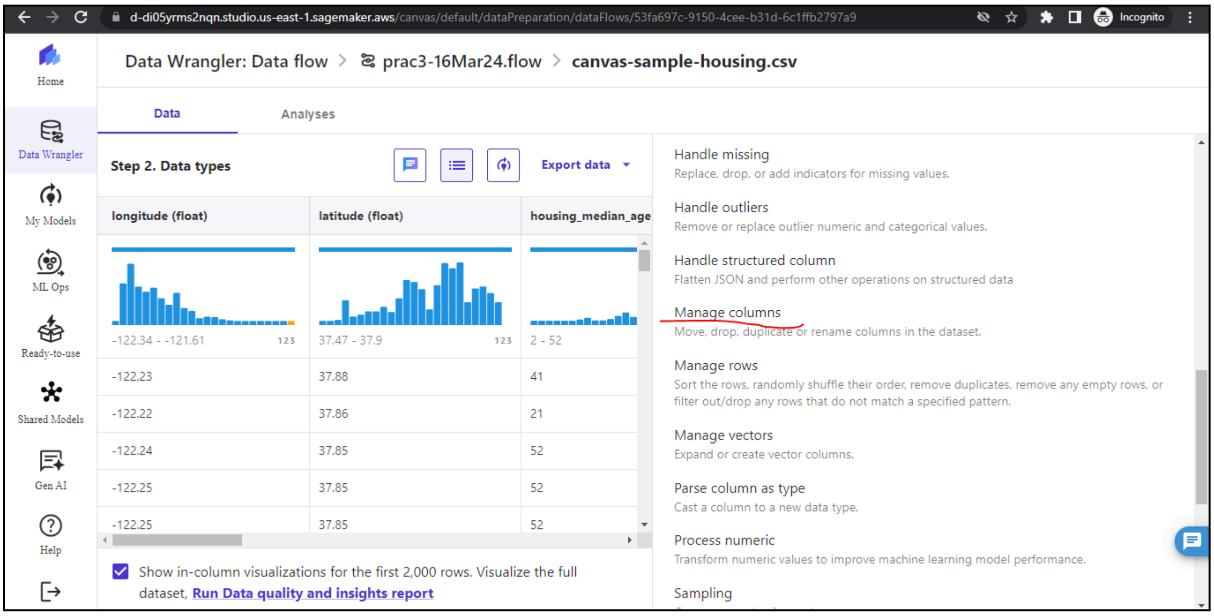
Reference for sample datasets explanation:
https://docs.aws.amazon.com/sagemaker/latest/dg/canvas-sample-datasets.html

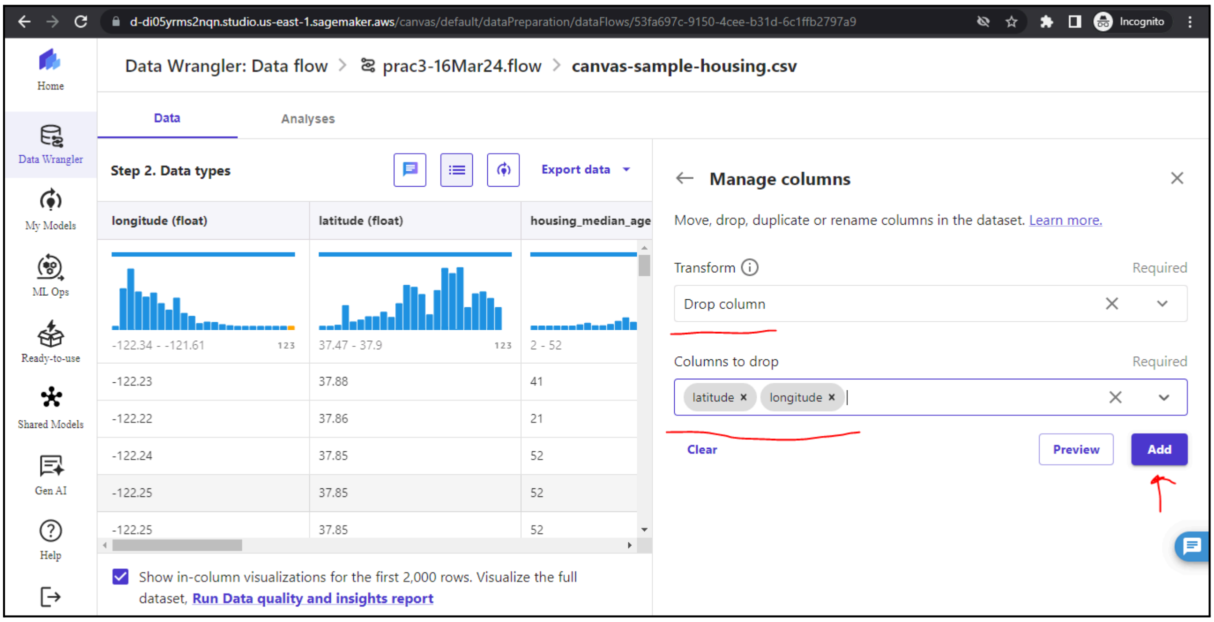
canvas-sample-housing.csv: This dataset contains data on the characteristics tied to a given housing price. You can use this dataset to predict housing prices. Use the **median_house_value** column as the target column, and use the numeric prediction model type with this dataset. To learn more about building a model with this dataset, see the SageMaker Canvas workshop page. This is the California housing dataset obtained from the StatLib repository.



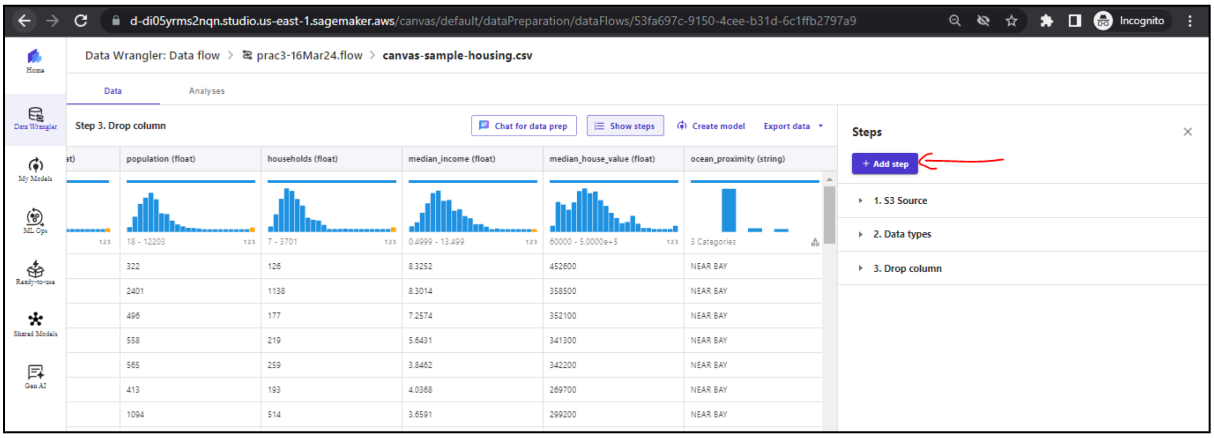
Delete irrelevant columns

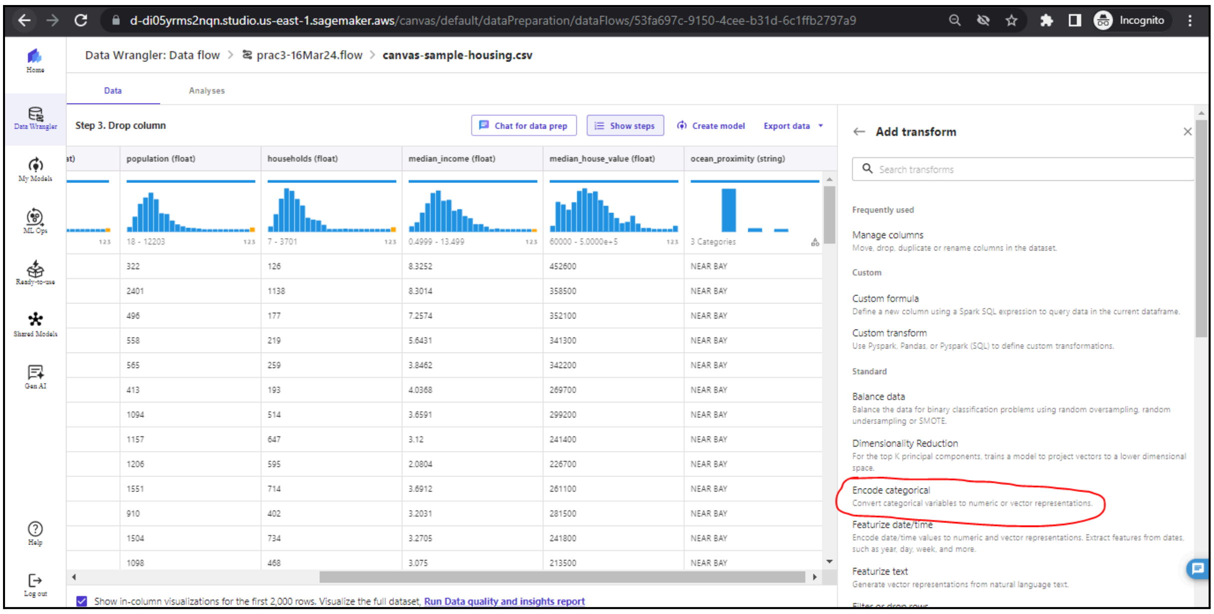


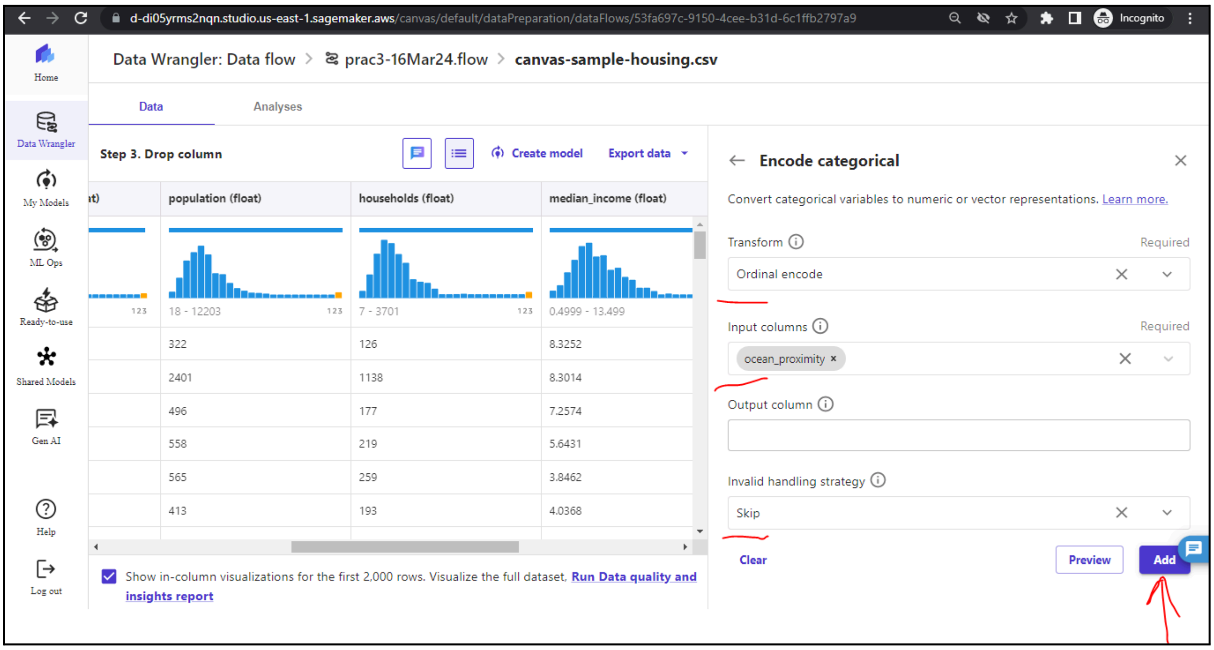




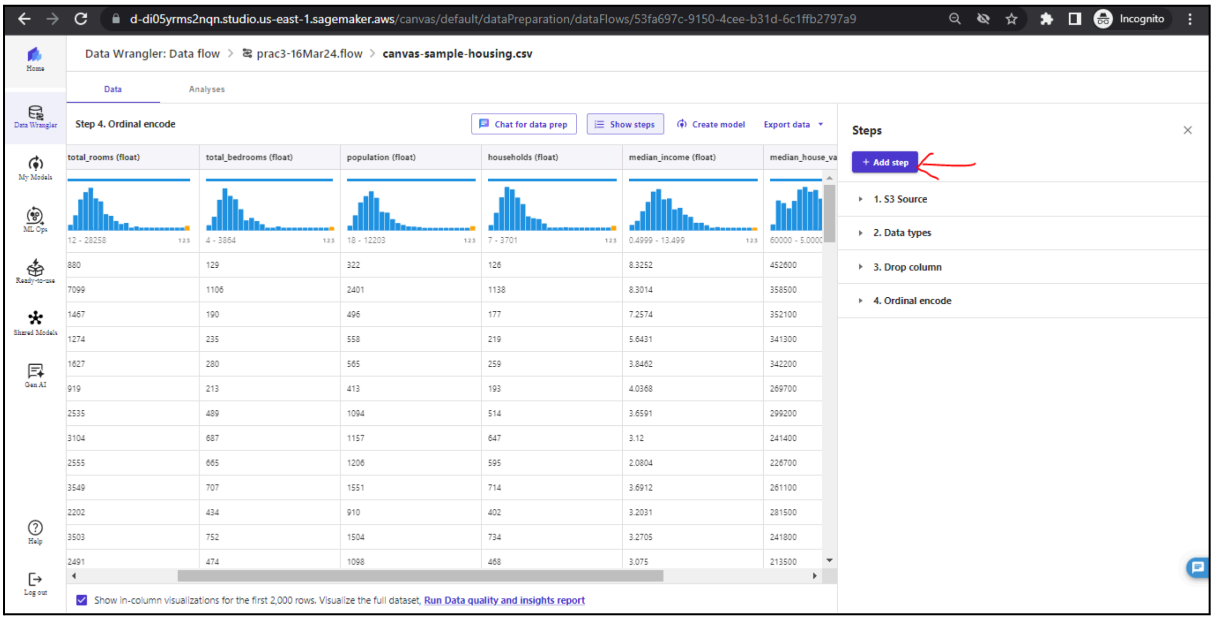
Ordinal Encoding for categorical column

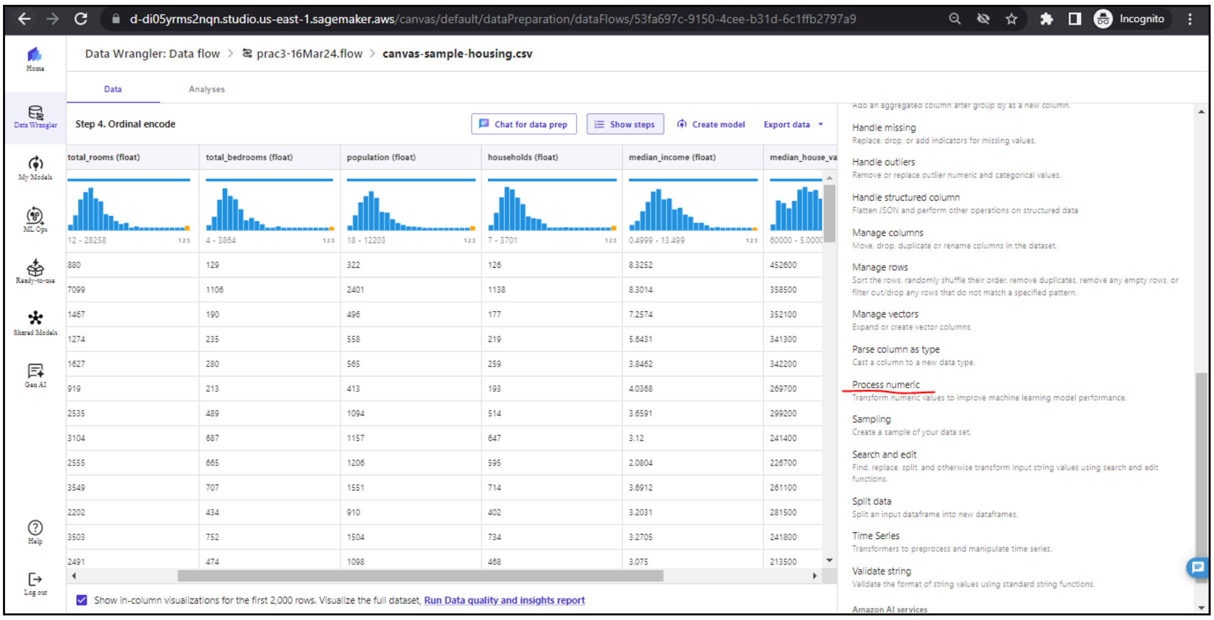


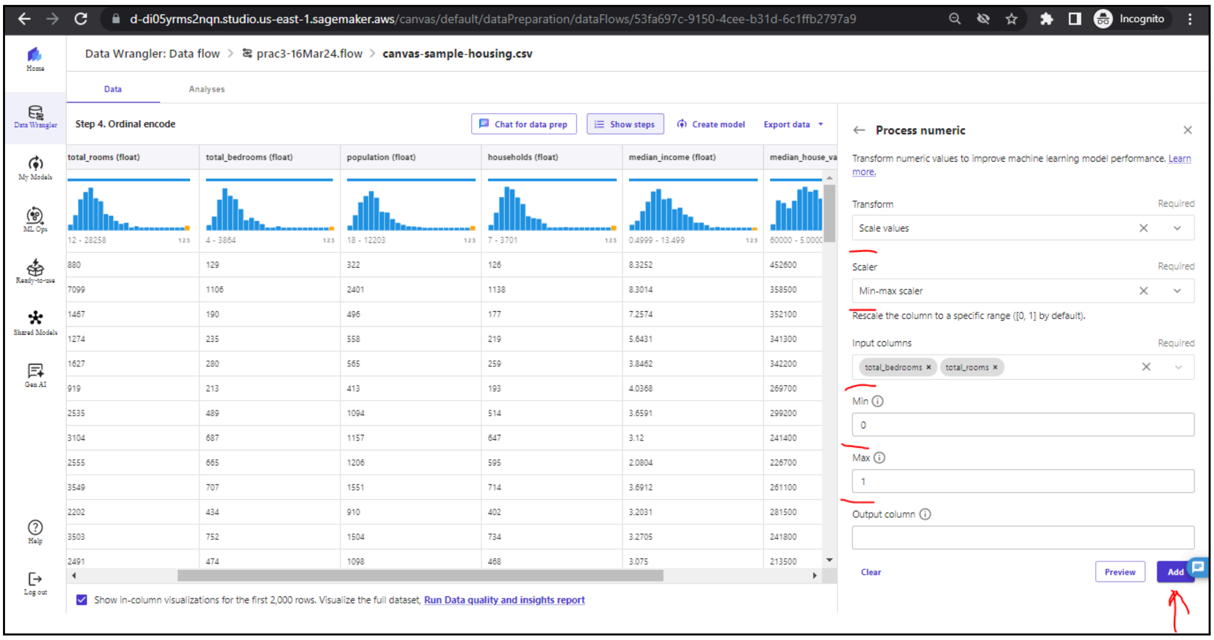




Min-Max Scaling

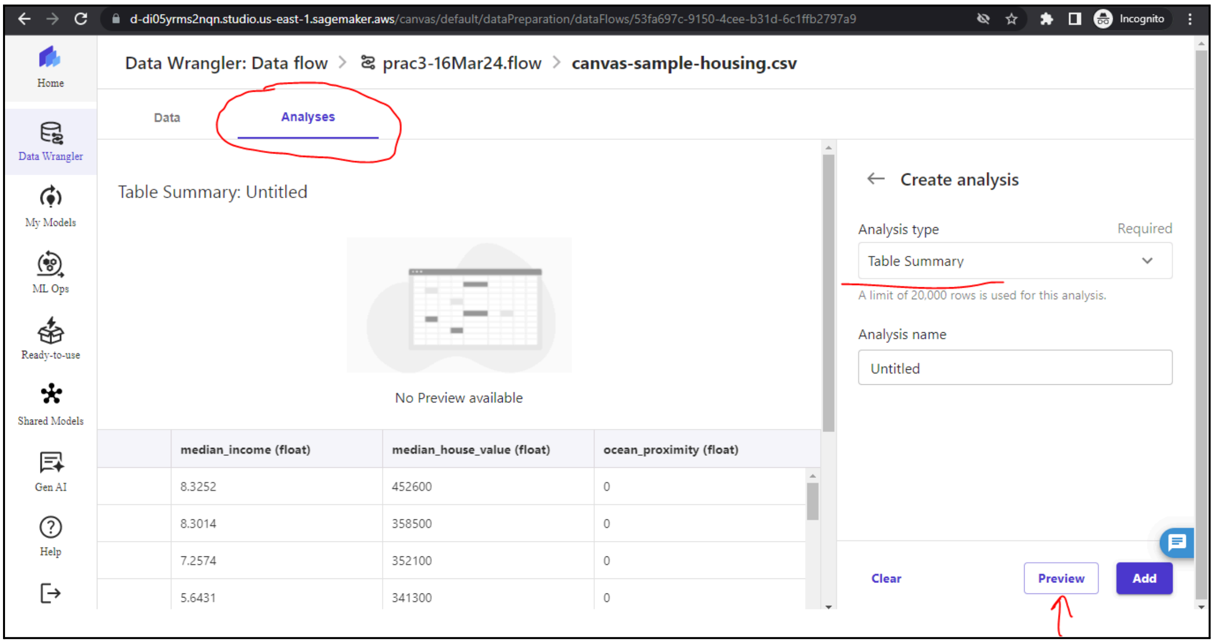


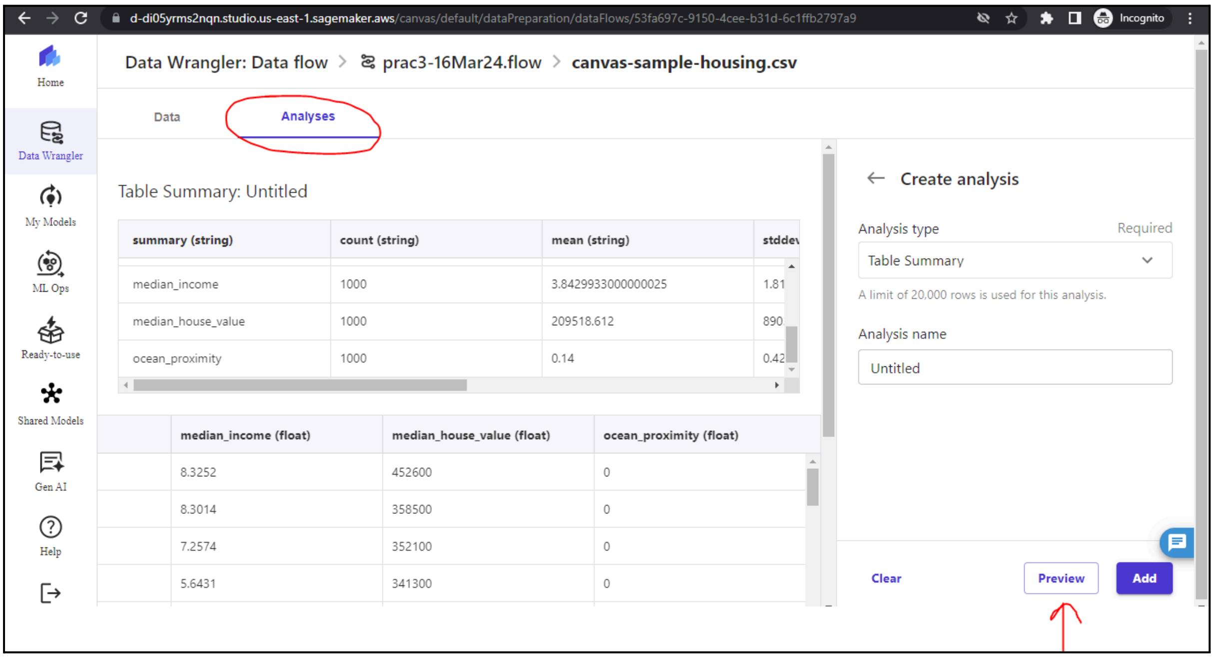

Data Wrangler: Data flow
prac3-16Mar24.flow
canvas-sample-housing.csv
Data
Analyses
Step 4. Ordinal encode
Chat for data prep
Show steps
Create model
Export data
Handle missing
Handle outliers
Handle structured column
Manage columns
Manage rows
Manage vectors
Parse column as type
Process numeric
Sampling
Search and edit
Split data
Time Series
Validate string


Data Wrangler: Data flow
prac3-16Mar24.flow
canvas-sample-housing.csv
Step 4. Ordinal encode
Process numeric
Transform
Scale values
Scaler
Min-max scaler
Input columns
Min
Max
Output column
Clear
Preview
Add

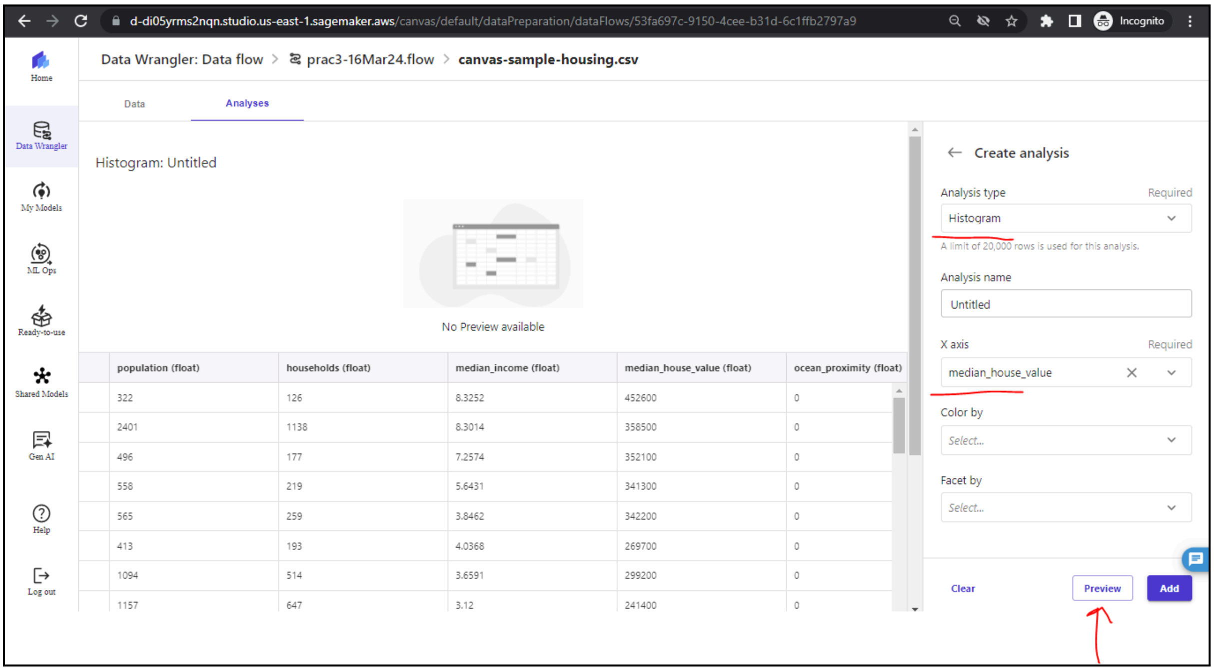
EDA using SageMaker DataWrangler

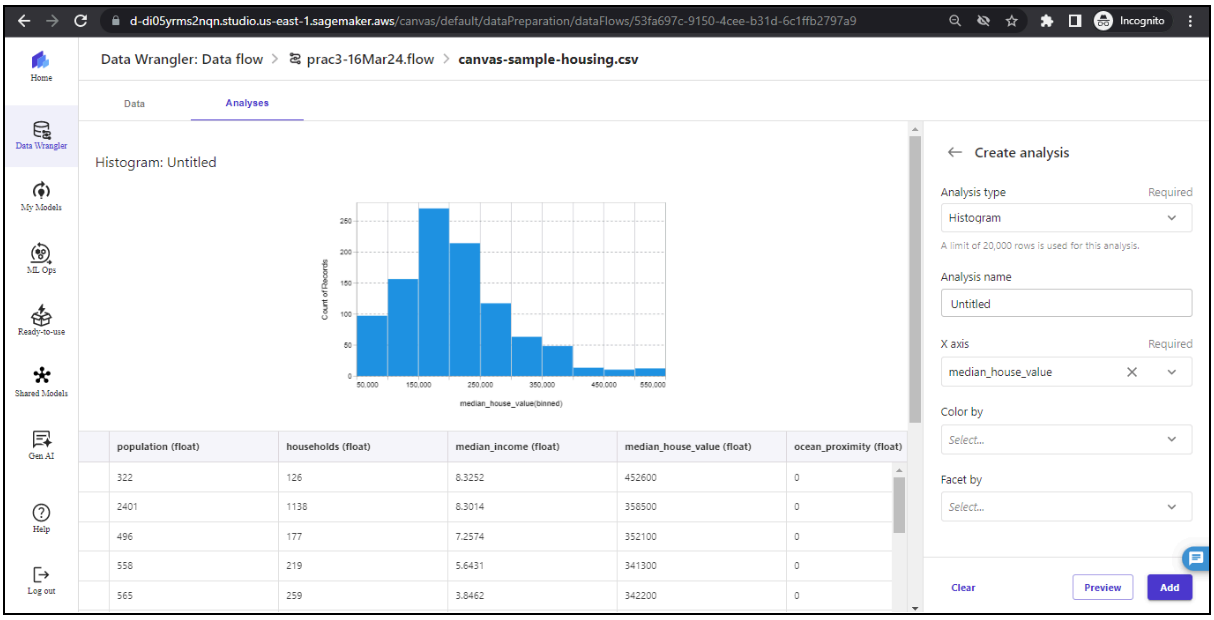
Table Summary





Checking if the target column is normally distributed (bell shaped graph)





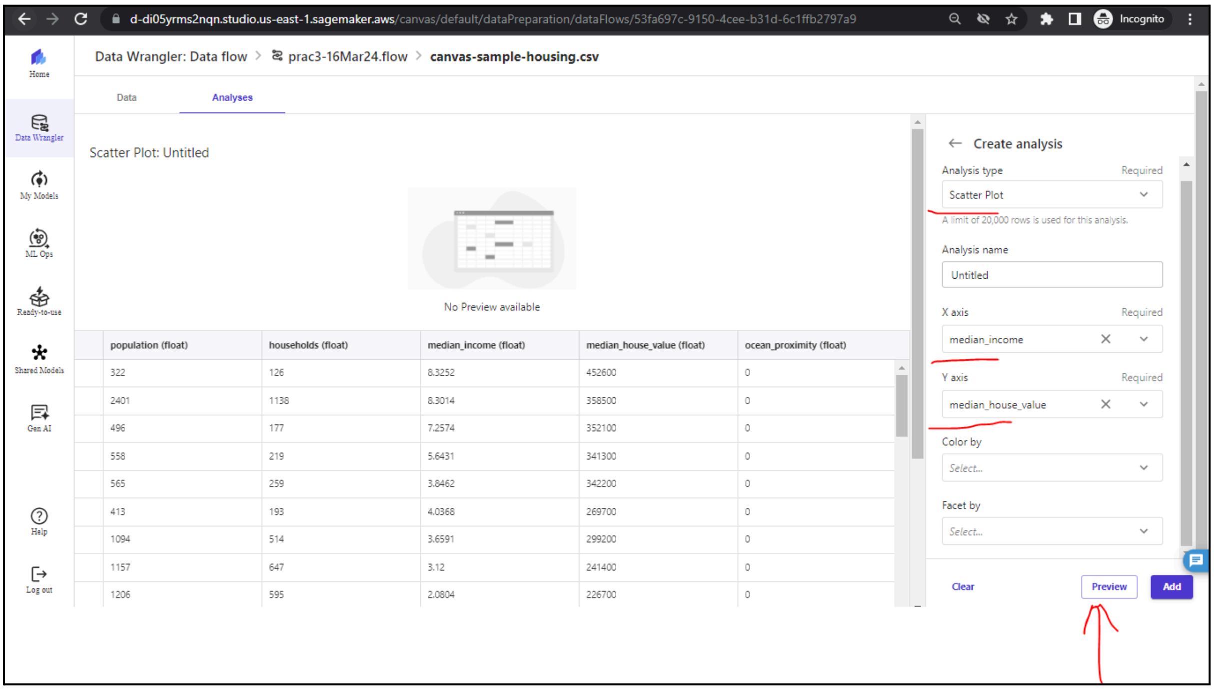
Data is normally distributed but slightly skewed towards the left

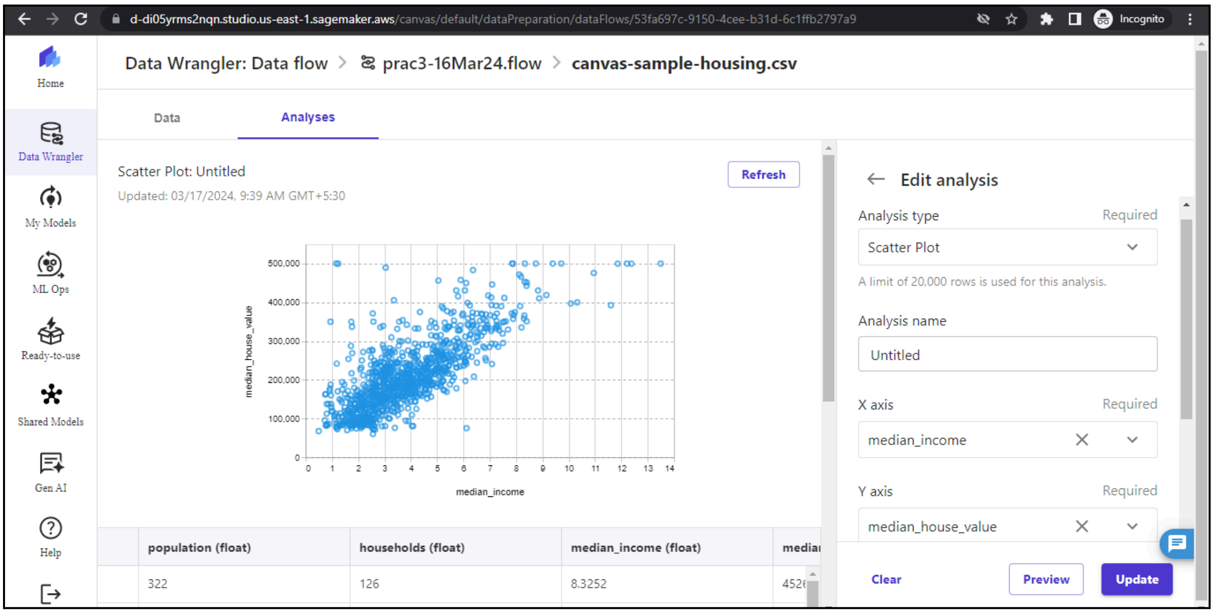
Since this is a sample dataset, we can proceed with a slightly skewed data also

Checking if the relationship between the predictor and target variable is linear

The **dependent variable** is called the **target** and the **independent variable** is called the **predictor**

The **dependent variable (target) depends** on the **independent variable (predictor)**

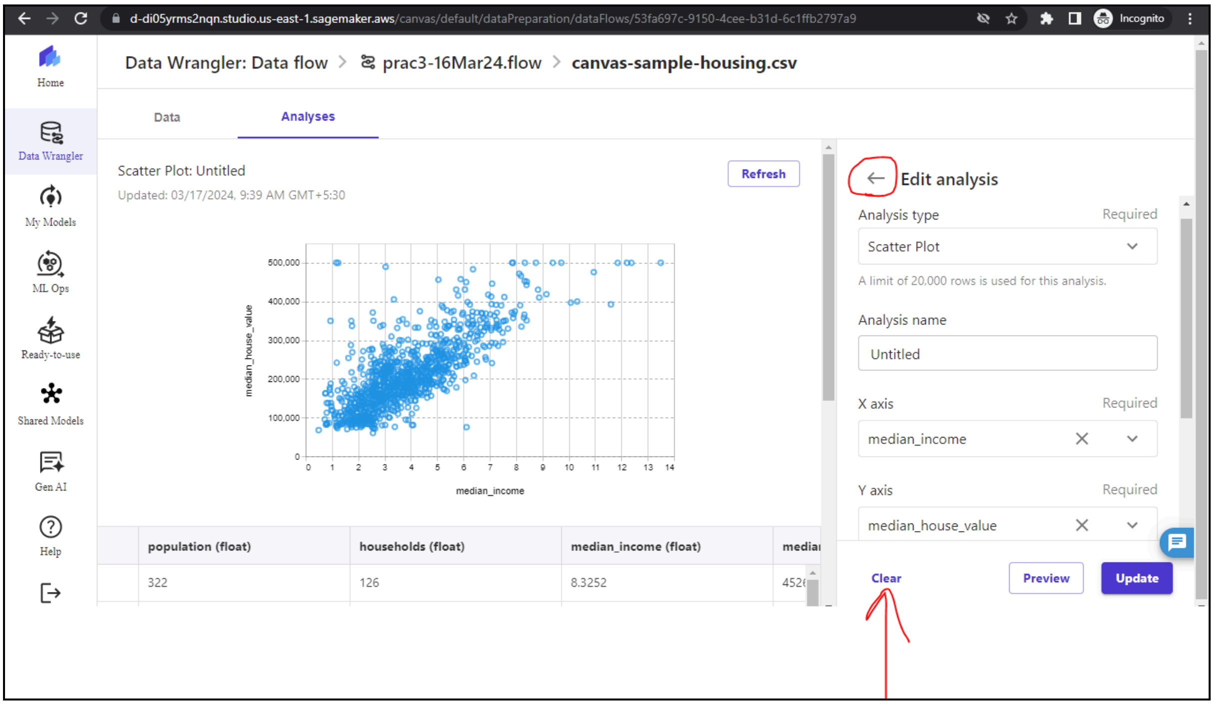


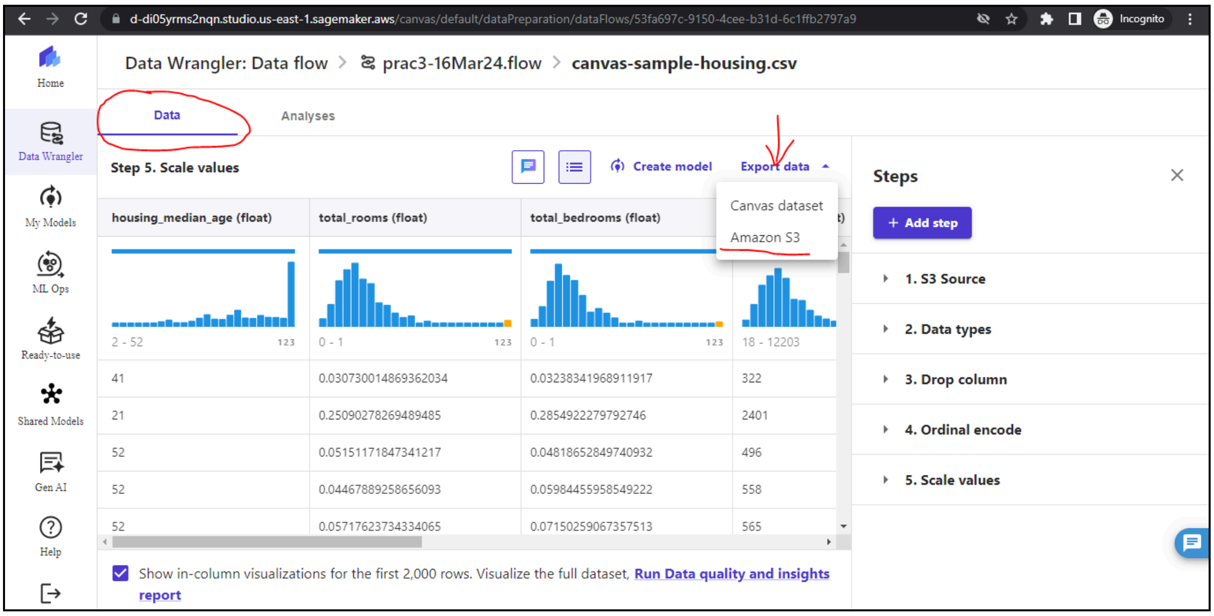


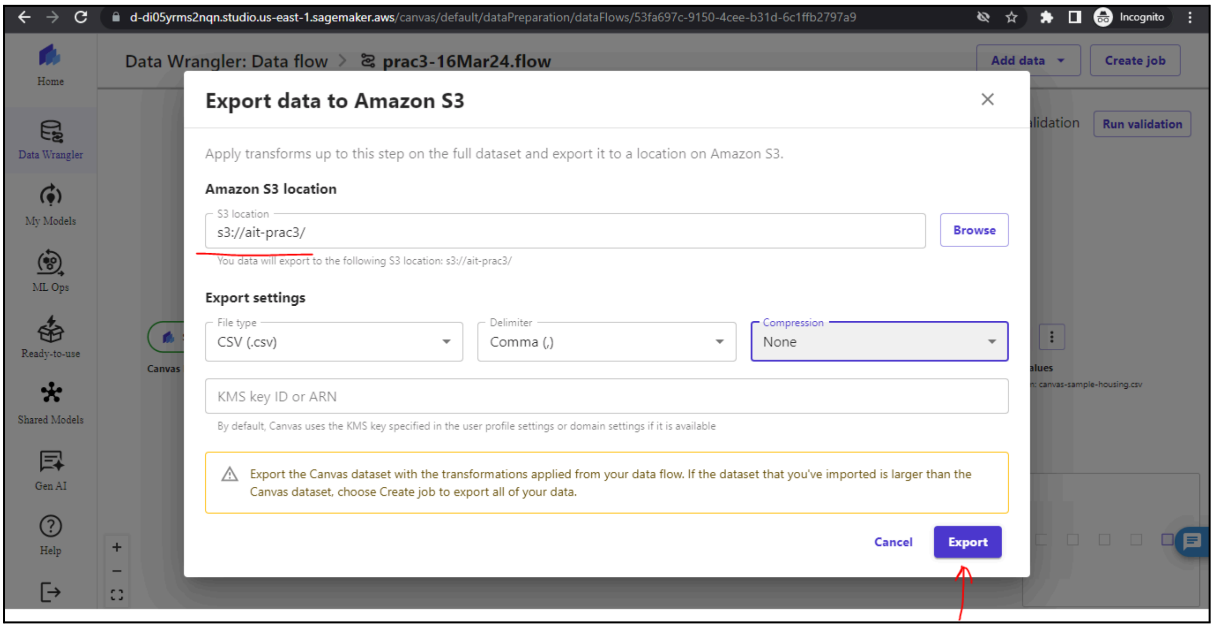
The columns' relation is roughly linear

The relationship looks roughly linear, so we can proceed with the linear regression.

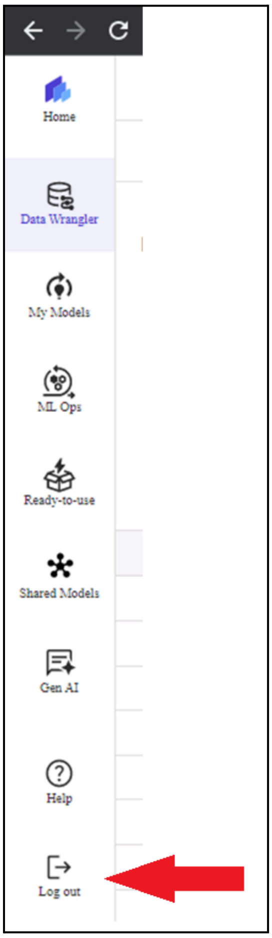
Export prepared data





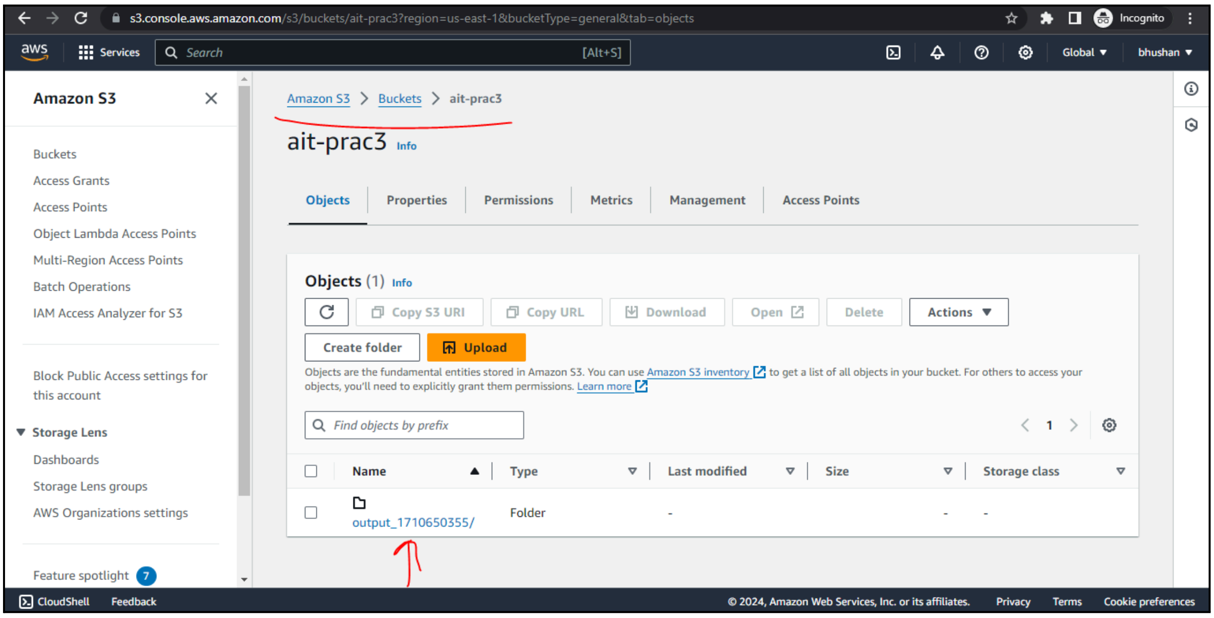


Logout of the Canvas after the data is exported

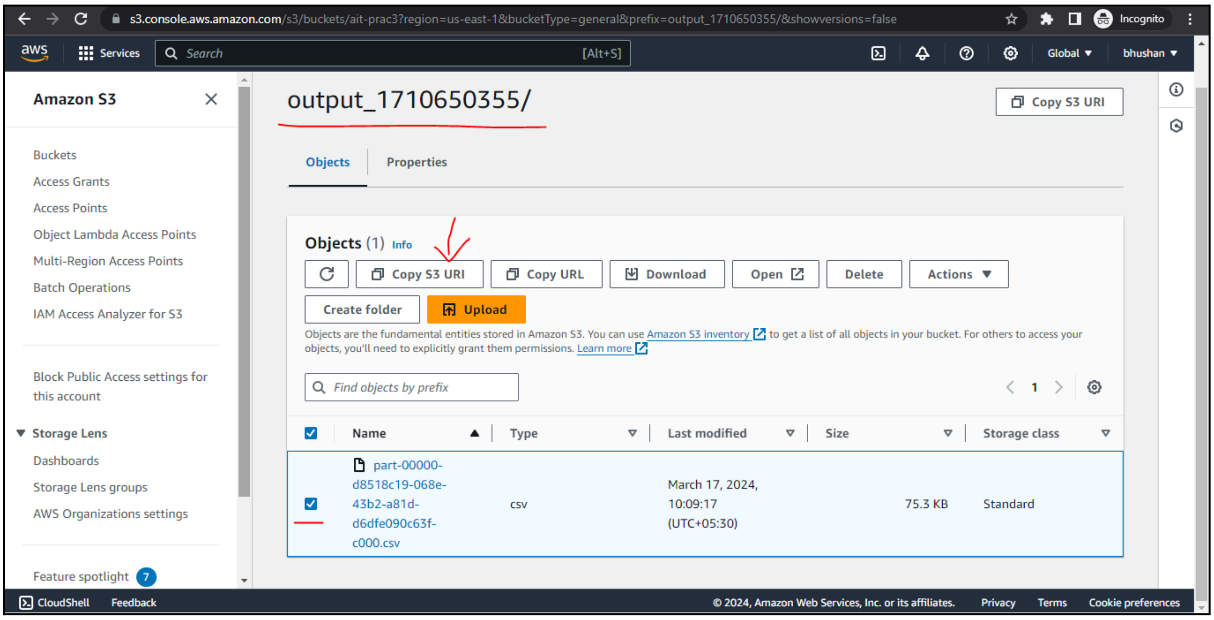


Important: Delete the SageMaker-domain-user and the SageMaker-domain

Locate the exported CSV file in the bucket

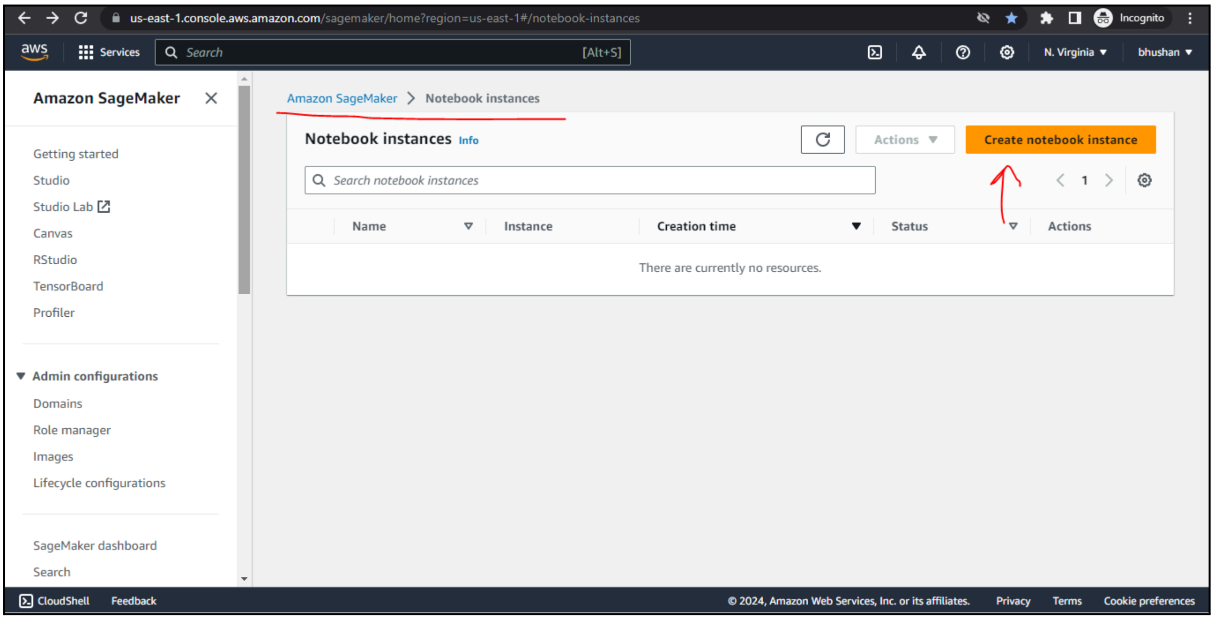


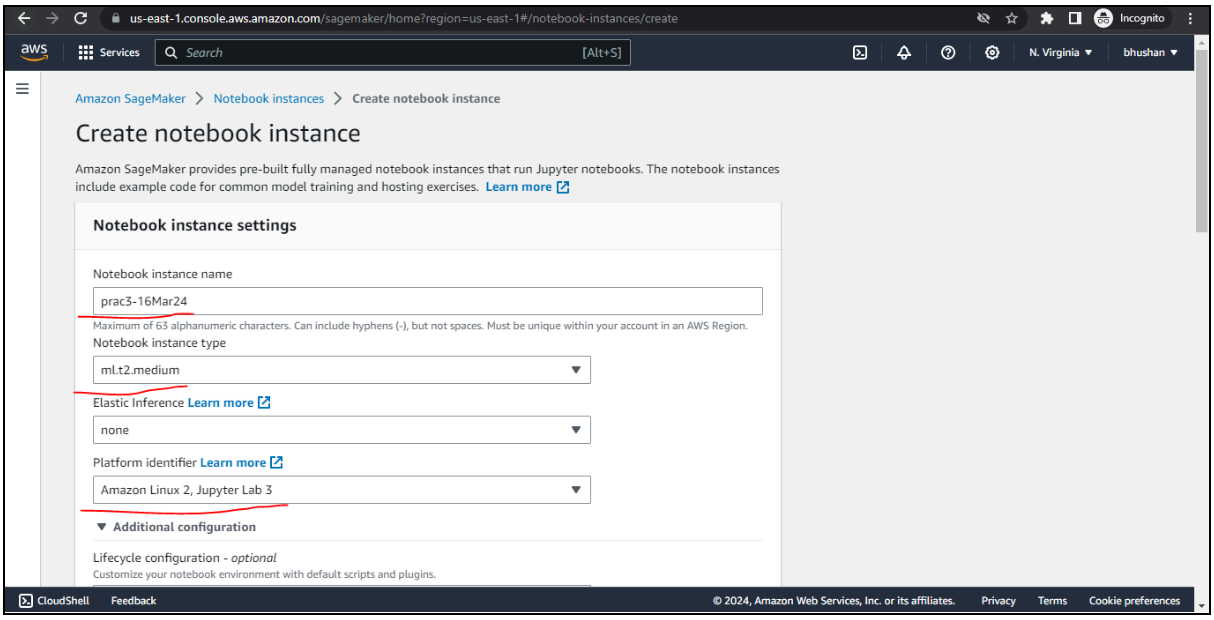
Copy the S3 URI of the csv file (to be used later in the python program)



Simple Linear Regression using SageMaker

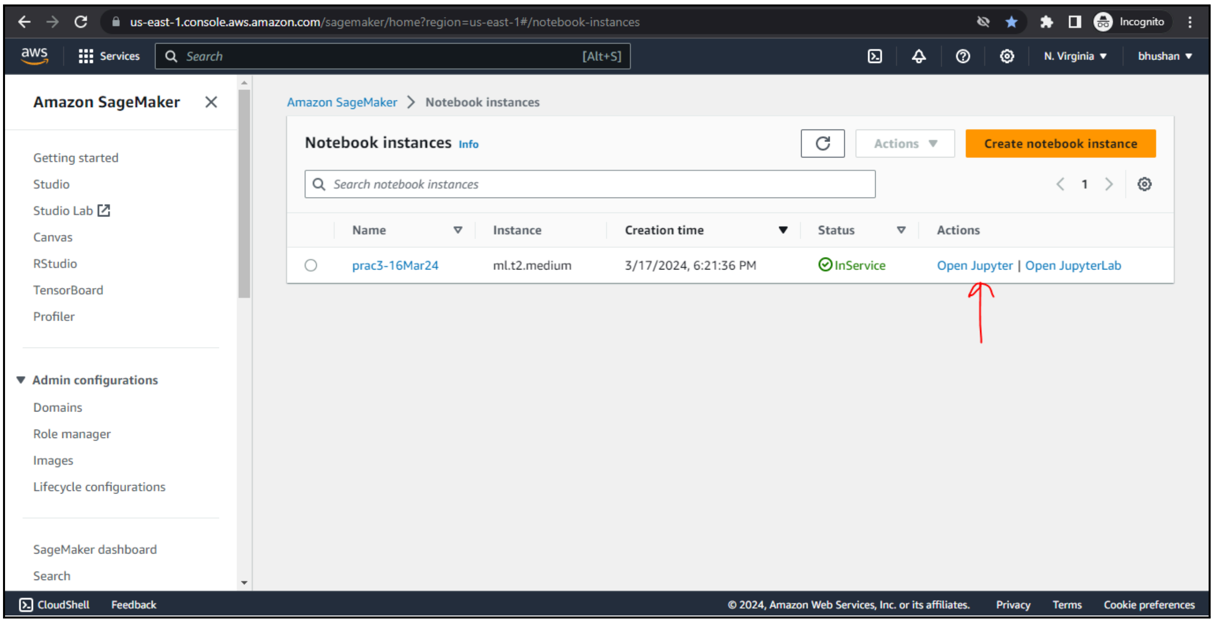
Create a Notebook

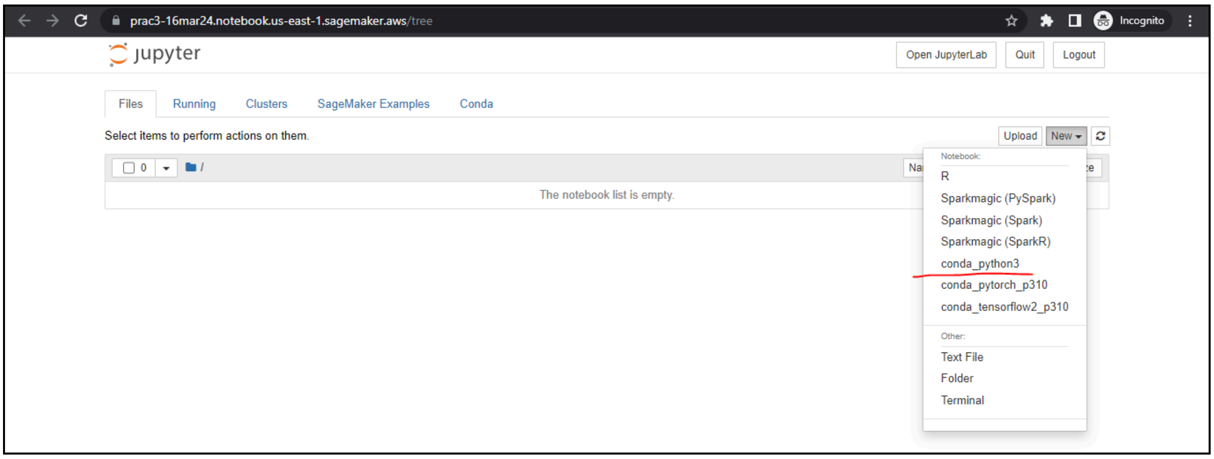




Click on “create” to create the notebook

Open in Jupyter to add the code





Add the following code

\# Using linear regression algorithm from sklearn library

import pandas as pd

import boto3

from io import StringIO

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import r2_score

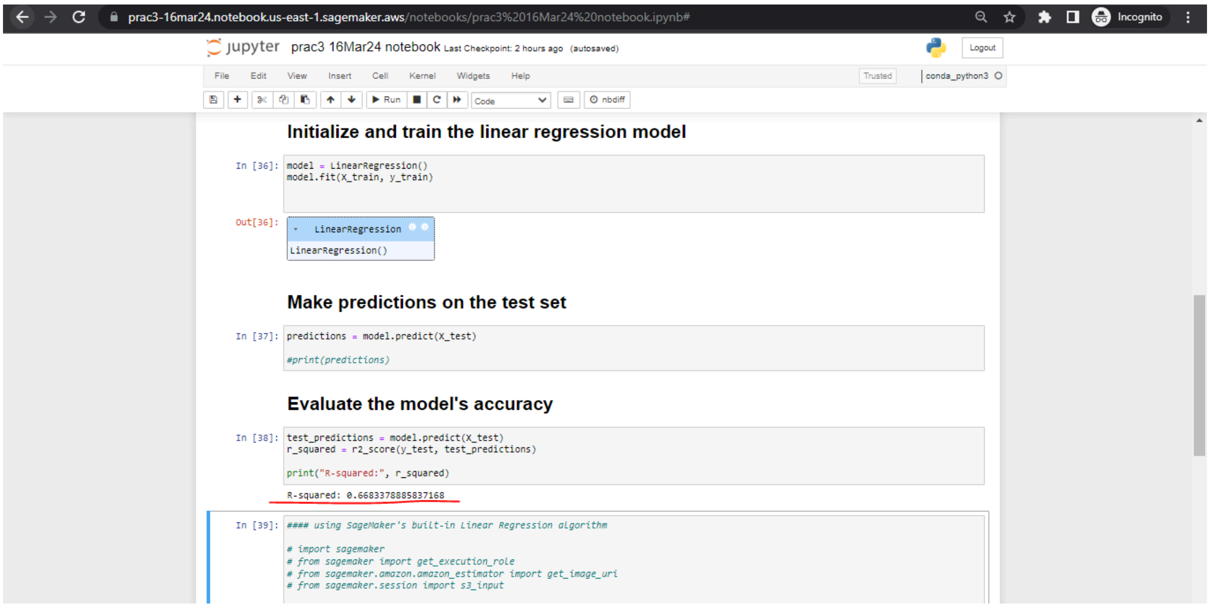
\# Load the dataset

s3_bucket = 'ait-prac3'

```
s3_key =
'output_1710650355/part-00000-d8518c19-068e-43b2-a81d-d6dfe090c63f-c000.csv'
s3_client = boto3.client('s3')
response = s3_client.get_object(Bucket=s3_bucket, Key=s3_key)
data = pd.read_csv(response['Body'])
# data = pd.read_csv(response['Body'], na_values=['NA', 'N/A', 'NaN', ''])
# Split the data into features (X) and target variable (y)
X = data[['median_income']]
y = data[['median_house_value']] # Notice the double square brackets to keep it as a
DataFrame
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions on the test set
predictions = model.predict(X_test)
#print(predictions)
# Extract the coefficients
# slope = model.coef_[0][0]
# intercept = model.intercept_[0]
# Print the equation of the linear regression line
# print(f"Linear Regression Equation: y = {slope:.2f}x + {intercept:.2f}")
# # Evaluate the model's accuracy
test_predictions = model.predict(X_test)
r_squared = r2_score(y_test, test_predictions)
print("R-squared:", r_squared)
# using SageMaker's built-in Linear Regression algorithm
# import sagemaker
# from sagemaker import get_execution_role
# from sagemaker.amazon.amazon_estimator import get_image_uri
# from sagemaker.session import s3_input
# Define your S3 bucket and prefix
# bucket = 'your-s3-bucket'
# prefix = 'your-prefix'
```

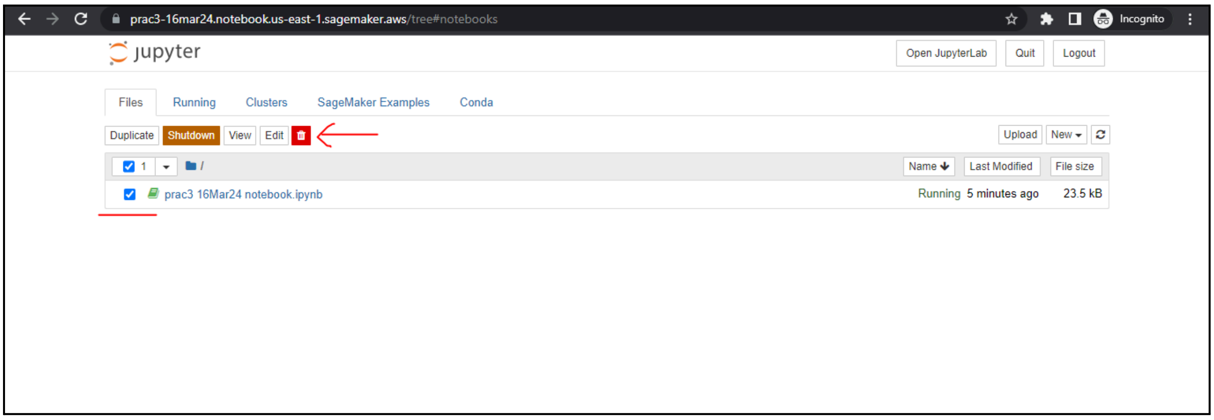
```
# Set up SageMaker session and role
# sagemaker_session = sagemaker.Session()
# role = get_execution_role()
# Specify the location of your dataset in S3
# train_data = 's3://{}/{}/{}'.format(bucket, prefix, 'canvas-sample-housing.csv')
# Create a LinearLearner estimator
# linear_container = get_image_uri(sagemaker_session.boto_region_name, 'linear-learner')
# linear = sagemaker.estimator.Estimator(linear_container,
#                                 role,
#                                 train_instance_count=1,
#                                 # train_instance_type='ml.m4.xlarge',
#                                 train_instance_type='',
#                                 output_path='s3://{}/{}/output'.format(bucket, prefix),
#                                 sagemaker_session=sagemaker_session)
# Set hyperparameters
# linear.set_hyperparameters(predictor_type='regressor',
#                     mini_batch_size=50,
#                     epochs=3)
# Train the model
# linear.fit({'train': s3_input(train_data, content_type='text/csv')})
```

Output

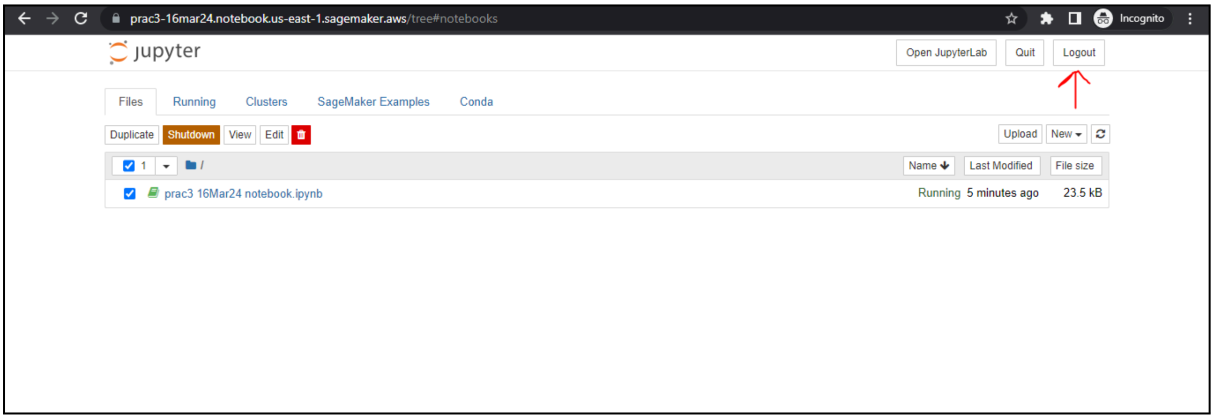


Since the R^2 value is 0.66, we can say that the linear regression model is 66% accurate

Delete the Jupyter Notebook



Logout from Jupyter



CLEAN UP

1. Stop all the services that are running.
2. Delete all the resources that were created for this practical (including the buckets and the notebooks, etc.)

Conclusion: We saw data Preparation and EDA using AWS SageMaker DataWrangler, linear regression using SageMaker.