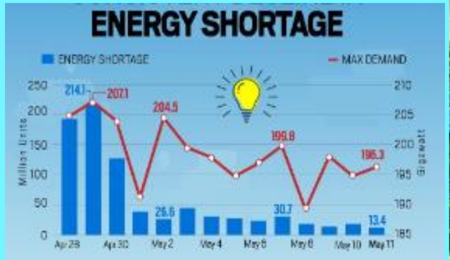
Energy Demand Forecasting

Business Understanding

Business Problem

- Inaccuracies in estimating energy needs lead to blackouts and economic loss due to underestimation, and wasteful expenditure and higher emissions due to overestimation, disrupting businesses and the economy.



Business Objective

- Minimizing inaccuracies in energy consumption estimations leading to economic disruptions, blackouts and environmental impact through predictive modelling.

Business Constraints

- Data Availability
- Model Complexity: Building a model that accurately captures complex patterns in the data can be difficult.
- Translation to policy decisions

- Time-series forecasting model to predict 'Total Energy Consumption' at a monthly frequency. The model will consider historical trends and patterns in energy generation sources to forecast short-term – **1 month** and long-term **- 6 months** energy consumption.
- The project will also explore the relationship between different energy generation sources and the total energy met. By simulating various scenarios, we can assess the impact of changes in generation mix on future energy consumption. This can help in planning and optimizing the energy mix for future needs.

Success Criterion

Business Success

- Achieve precise energy forecasting to cut down on waste and costs, by 5%.
- Strengthen supply chain reliability and operational effectiveness; base strategic choices on accurate energy data, on a sustainable basis.

ML Success Criteria

- Consistent high forecast accuracy and model performance with expected less than 10% ML - RMSE.
- Positive feedback loop effectiveness, with model adjustments improving forecast precision over time.

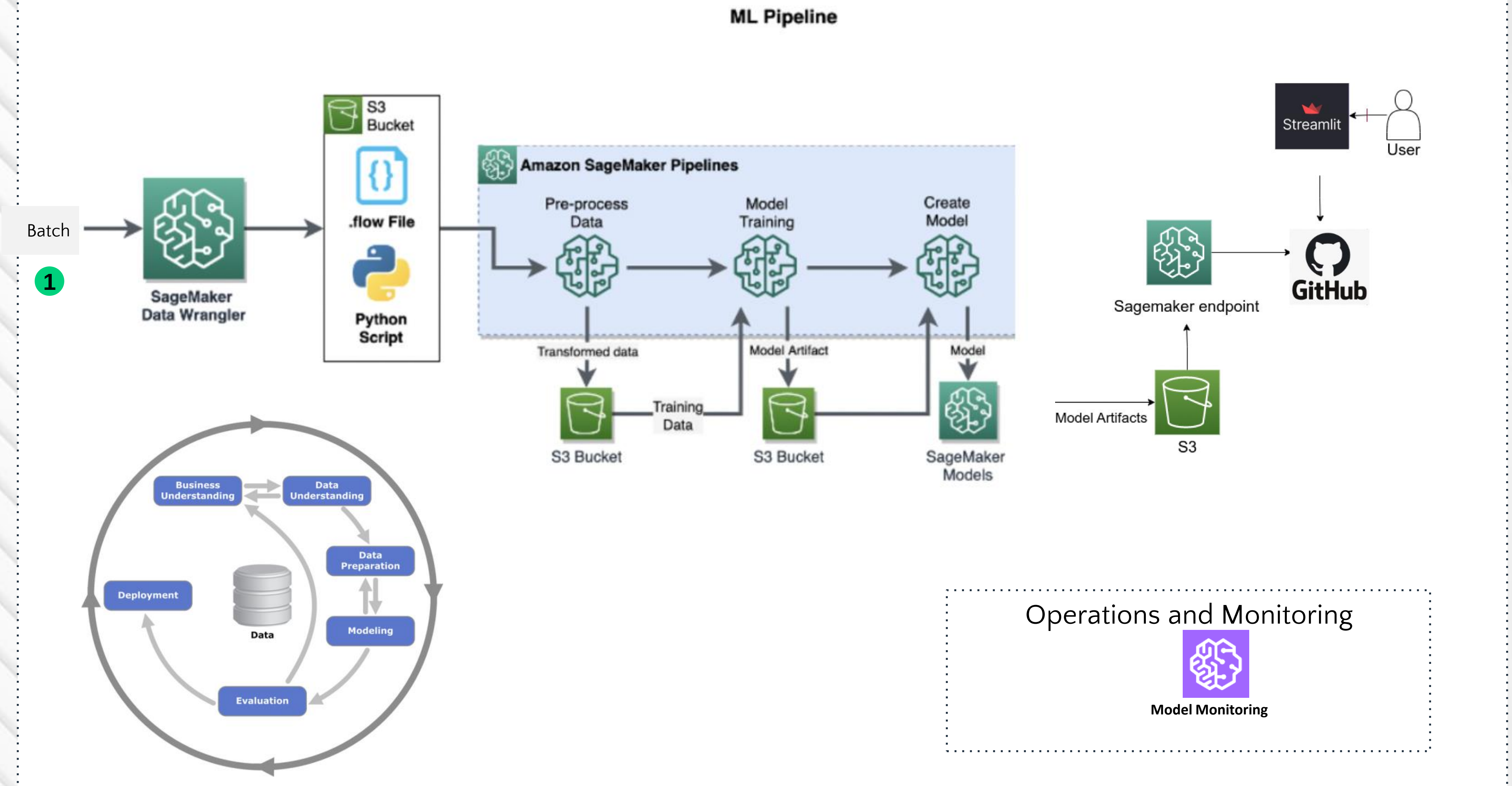
Economic Success Criteria

- Accurate energy consumption forecasts can reduce expenses, boost energy efficiency, fostering a sustainable and prosperous economy.
- Aiming for at least a 20% decrease in energy procurement costs

Architecture

Data Sources

Energy Consumption Data from POSOCO



Data Preparation

Data Collection

- Power System Operation Corporation Limited (POSOCO) daily reports, specifically from the publicly available daily reports section on their website:
- https://posoco.in/reports

Feature selection

- Correlation analysis to identify strongly related features to energy consumption.
- Time-series decomposition to select relevant temporal features.
- Domain expertise consultation for choosing features impacting energy usage.

Data selection

- Inclusion of historical data relevant to forecasting windows (short-term, long-term).
- Seasonal periods covering various demand patterns (summer, monsoon, winter).

Noise reduction

- Identify and smooth out spikes or dips that do not correlate with known trends or events, using rolling averages or median filtering.
- Utilize anomaly detection algorithms to identify and correct or remove outliers that could distort the model.

Data imputation

- Implement model-based imputation techniques like K-Nearest Neighbours (KNN).
- Time-series specific methods like forward fill, backward fill, or interpolation can be used, considering the temporal nature of the data.

Feature engineering

- Normalization / Standardisation
- Feature transformation
- Interaction Features: Create new features that are combinations of two or more.

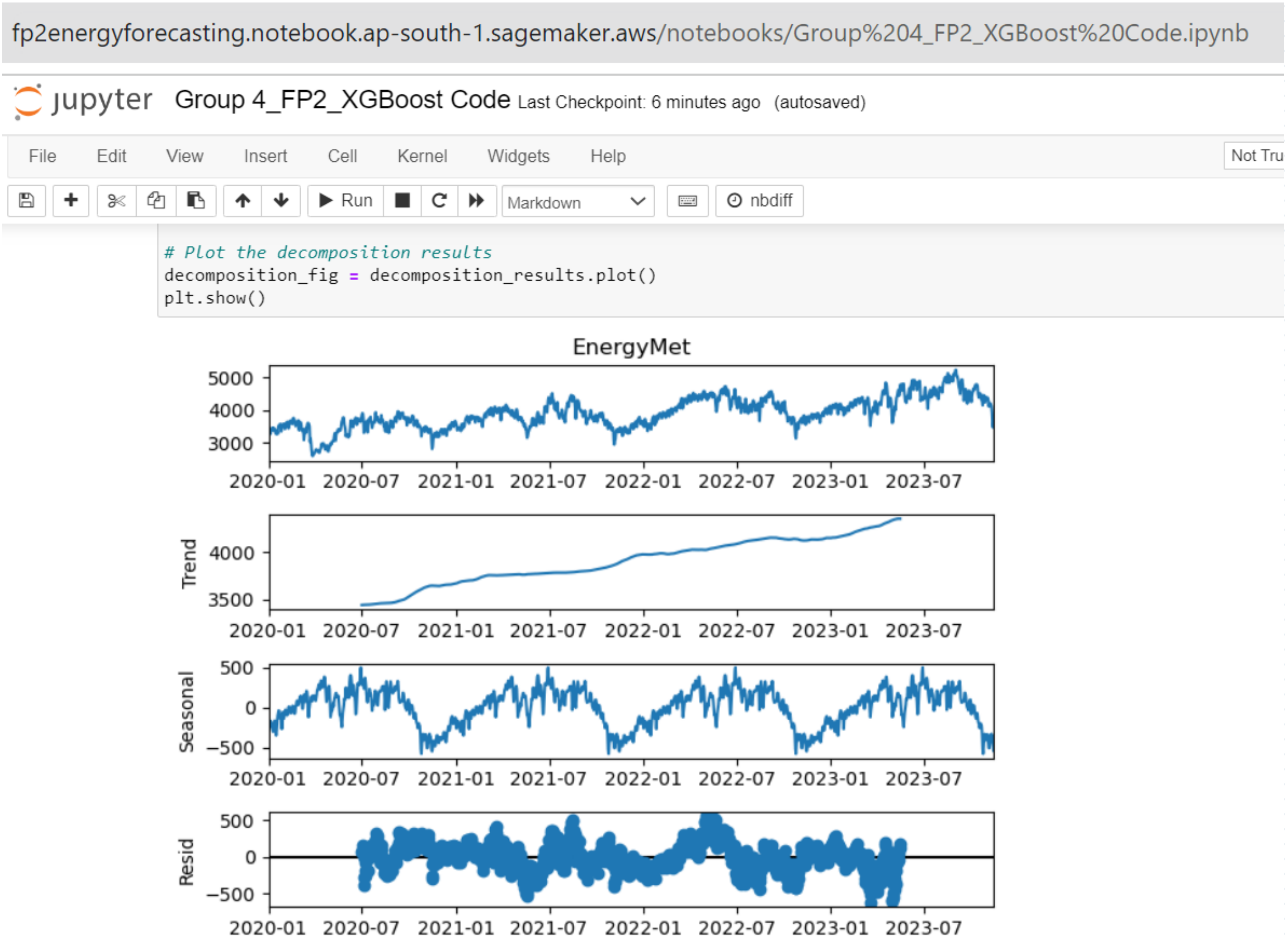
Daily Report Weekly Report / PSP Report / 2023-2024 / Oct 2023 Search

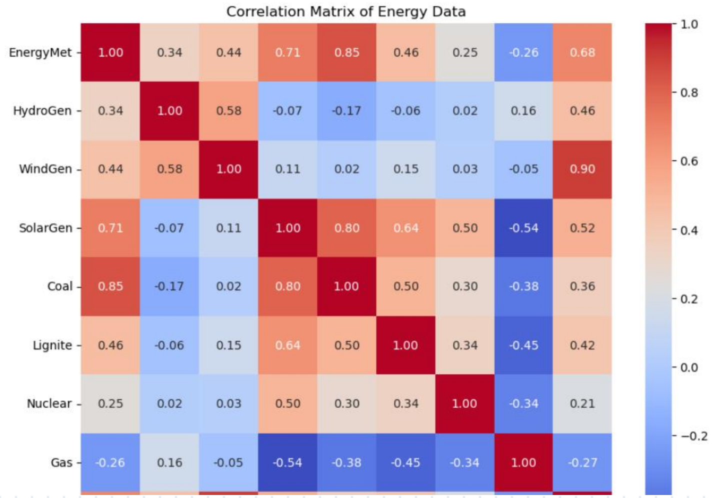
Name	Size	Modified	Actions
..			
Weekly 021023 to 081023.pdf	768.37 KB	13.10.23 18:04	
Weekly 091023 to 151023.pdf	797.85 KB	20.10.23 17:20	
Weekly 161023 to 221023.pdf	767.4 KB	03.11.23 15:36	
Weekly 231023 to 291023.pdf	767.55 KB	03.11.23 17:31	
Weekly 250923 to 011023.pdf	823.09 KB	06.10.23 17:25	

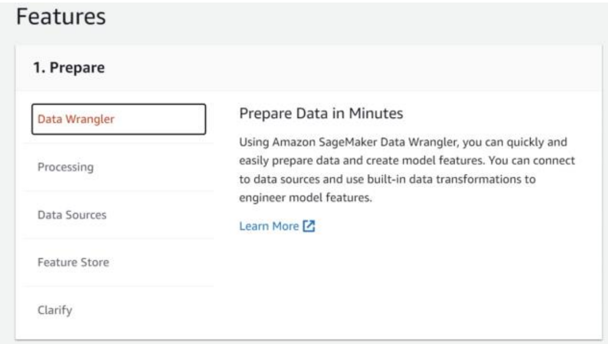
Date	EnergyMet	HydroGen	WindGen	SolarGen	Coal	Lignite	Nuclear	Gas	RES
01-01-2020	3380	280	123	111	2571	78	110	106	320
02-01-2020	3383	298	91	109	2601	81	110	104	287
03-01-2020	3390	295	64	118	2627	86	101	106	267
04-01-2020	3388	299	53	122	2680	81	101	107	211
05-01-2020	3271	273	91	112	2531	84	97	96	282
06-01-2020	3397	307	124	105	2593	78	97	97	316
07-01-2020	3442	307	98	96	2663	76	97	101	286
08-01-2020	3391	271	83	122	2651	77	97	99	289
09-01-2020	3403	289	102	139	2601	75	95	98	330
10-01-2020	3455	274	171	143	2582	74	95	105	409
11-01-2020	3457	273	128	138	2619	76	98	107	355
12-01-2020	3364	256	82	137	2607	74	92	103	301
13-01-2020	3421	297	59	128	2666	81	92	99	272
14-01-2020	3376	305	42	149	2609	80	98	96	272
15-01-2020	3301	269	94	150	2510	77	108	94	328

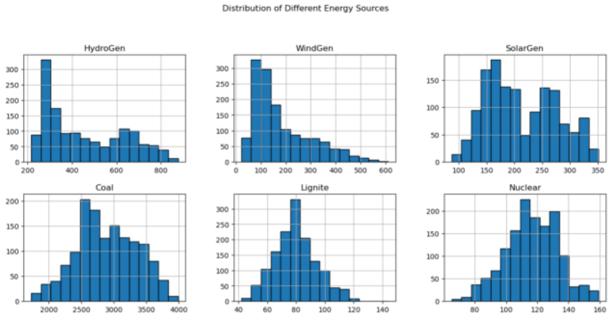
Target Column

Data Preparation









Modelling

MODEL SELECTION:

- Evaluate various time series models (e.g., SARIMAX , DeepAR , RNN, LSTM and XGboost.) based on the defined quality measures.
- Choose a model that best captures seasonal trends and patterns inherent in energy consumption data.

MODEL TRAINING:

- Train the selected model using AWS Sagemaker, carefully tuning parameters to optimize forecast performance.
- Validate the model against a hold-out set or use cross-validation to ensure generalization.

ASSURE REPRODUCIBILITY:

- Method Reproducibility: Document the entire modelling process in AWS Sagemaker, including data preprocessing steps, feature engineering techniques, and model parameters.
- Result Reproducibility: Use version control for both data and code to ensure that results can be replicated, and maintain a log of model performance metrics for each run.

Quality Measures:

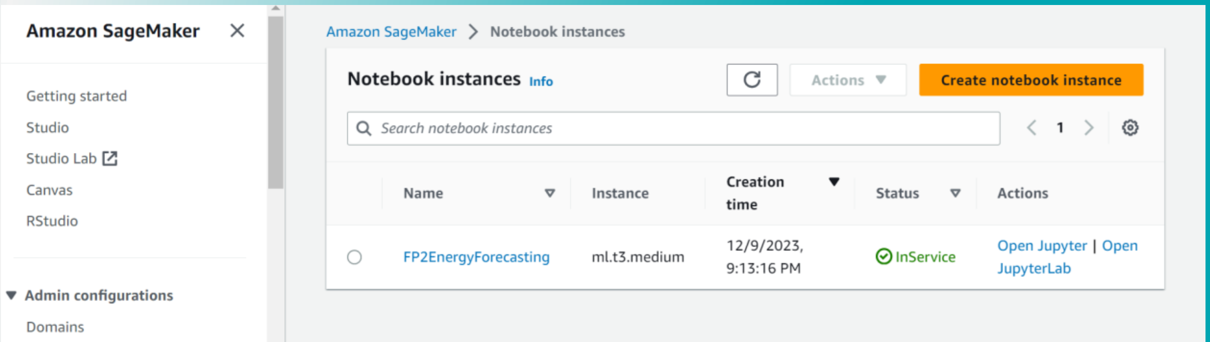
Performance: RMSE Score

Robustness: Ensure the model remains stable across different time frames and outlier events.

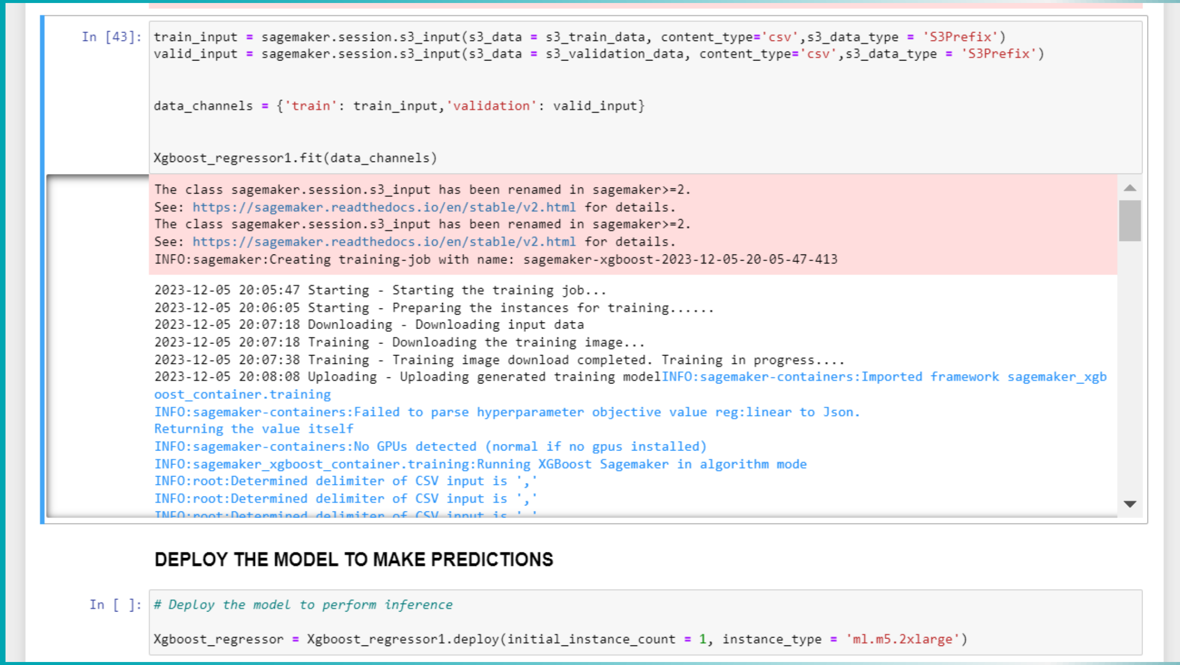
Scalability: The model should handle increasing data volumes as more historical data accumulates.

Model Complexity: Aim for the simplest model that still provides accurate predictions to facilitate maintenance and updates.

Modelling



Amazon SageMaker
Getting started
Studio
Studio Lab
Canvas
RStudio
Admin configurations
Domains
Amazon SageMaker > Notebook instances
Notebook instances Info
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
FP2EnergyForecasting
ml.t3.medium
12/9/2023, 9:13:16 PM
InService
Open Jupyter | Open JupyterLab



In [43]: train_input = sagemaker.session.s3_input(s3_data = s3_train_data, content_type='csv',s3_data_type = 'S3Prefix')
valid_input = sagemaker.session.s3_input(s3_data = s3_validation_data, content_type='csv',s3_data_type = 'S3Prefix')
data_channels = {'train': train_input,'validation': valid_input}
Xgboost_regressor1.fit(data_channels)
The class sagemaker.session.s3_input has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
The class sagemaker.session.s3_input has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
INFO:sagemaker:Creating training-job with name: sagemaker-xgboost-2023-12-05-20-05-47-413
2023-12-05 20:05:47 Starting - Starting the training job...
2023-12-05 20:06:05 Starting - Preparing the instances for training......
2023-12-05 20:07:18 Downloading - Downloading input data
2023-12-05 20:07:18 Training - Downloading the training image...
2023-12-05 20:07:38 Training - Training image download completed. Training in progress....
2023-12-05 20:08:08 Uploading - Uploading generated training modelINFO:sagemaker-containers:Imported framework sagemaker_xgboost_container.training
INFO:sagemaker-containers:Failed to parse hyperparameter objective value reg:linear to Json.
Returning the value itself
INFO:sagemaker-containers:No GPUs detected (normal if no gpus installed)
INFO:sagemaker_xgboost_container.training:Running XGBoost Sagemaker in algorithm mode
INFO:root:Determined delimiter of CSV input is ','
INFO:root:Determined delimiter of CSV input is ','
DEPLOY THE MODEL TO MAKE PREDICTIONS
In [ ]: # Deploy the model to perform inference
Xgboost_regressor = Xgboost_regressor1.deploy(initial_instance_count = 1, instance_type = 'ml.m5.2xlarge')

Model Evaluation

Validate Performance: Measure forecast accuracy against actual consumption using AWS Sagemaker built-in metrics. Ensure error margins are within acceptable thresholds for project goals.

Determine Robustness: Conduct scenario-based testing to confirm model reliability across diverse data conditions. Monitor long-term stability to validate model consistency over time.

Increase Explainability: Human-in-the-Loop (HITL): Involving human oversight in the decision-making process to ensure that the model's decisions are reasonable and justifiable. Develop intuitive visualization for end-user clarity on forecast insights.

Compare Results with Success Criteria: Benchmark model outputs against predefined economic and business metrics. Refine model in response to any discrepancies from target success benchmarks.

Model Deployment

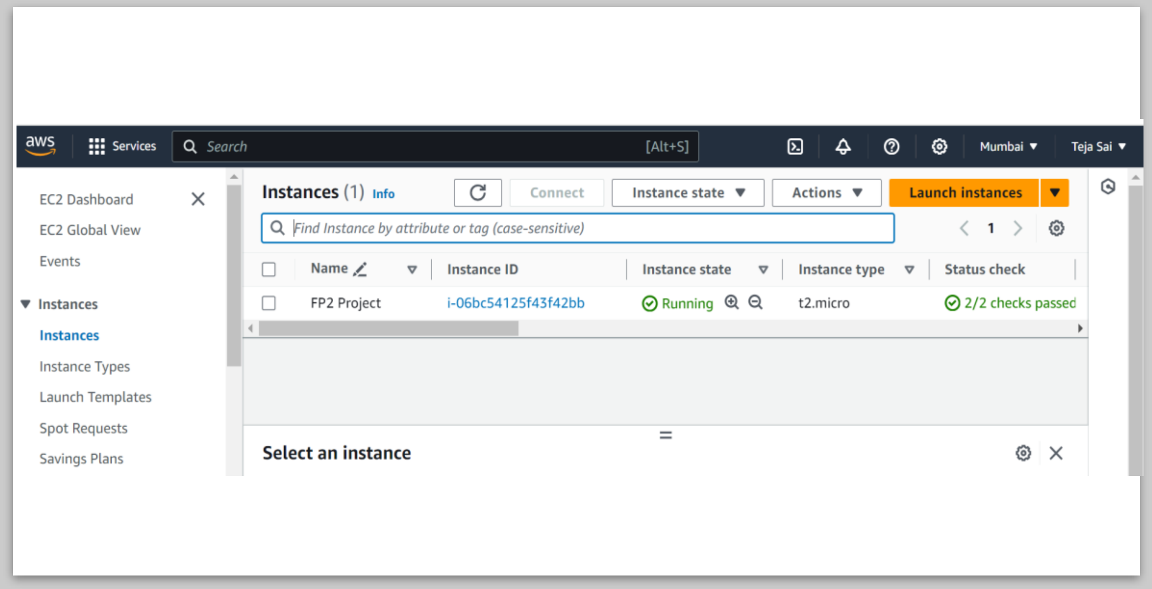
Define Inference Hardware: Opt for AWS Functions for cost-effective scaling or AKS for high-load, complex model inferences.

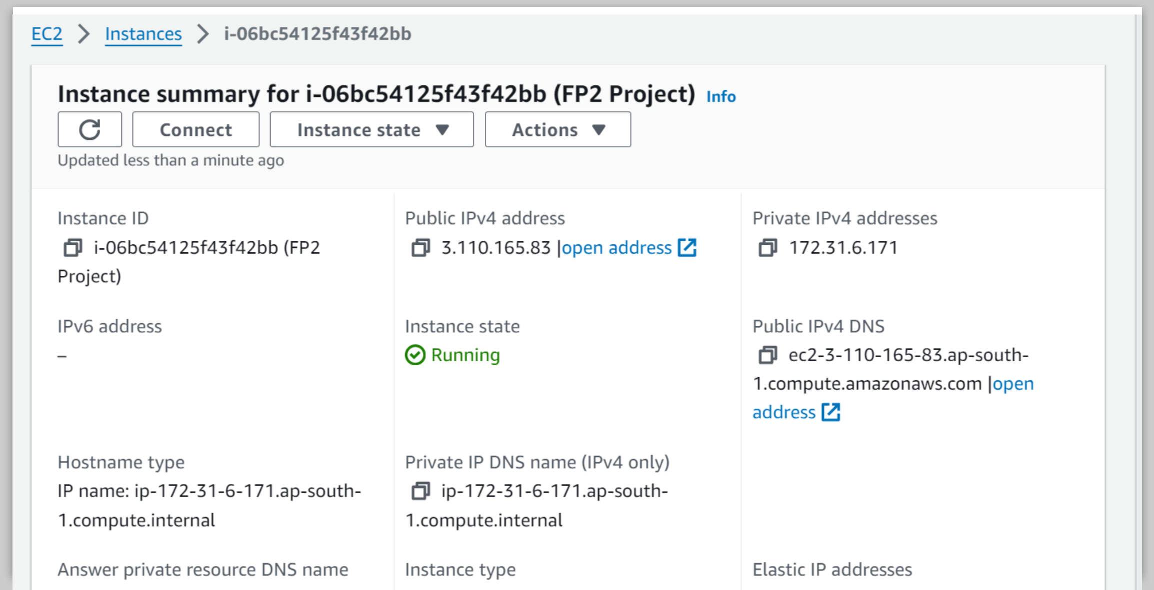
Model Evaluation under Production Condition: Implement shadow deployment and A/B testing within AWS Sagemaker to validate model performance with production traffic.

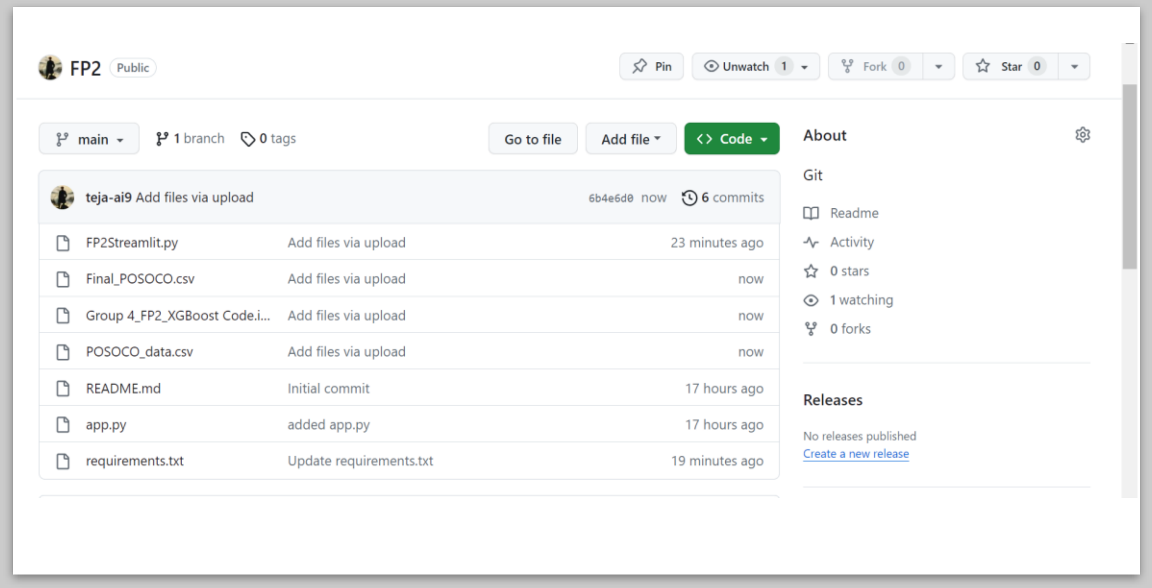
Assure User Acceptance and Usability: Integrate feedback mechanisms in applications and provide clear documentation for end-user engagement.

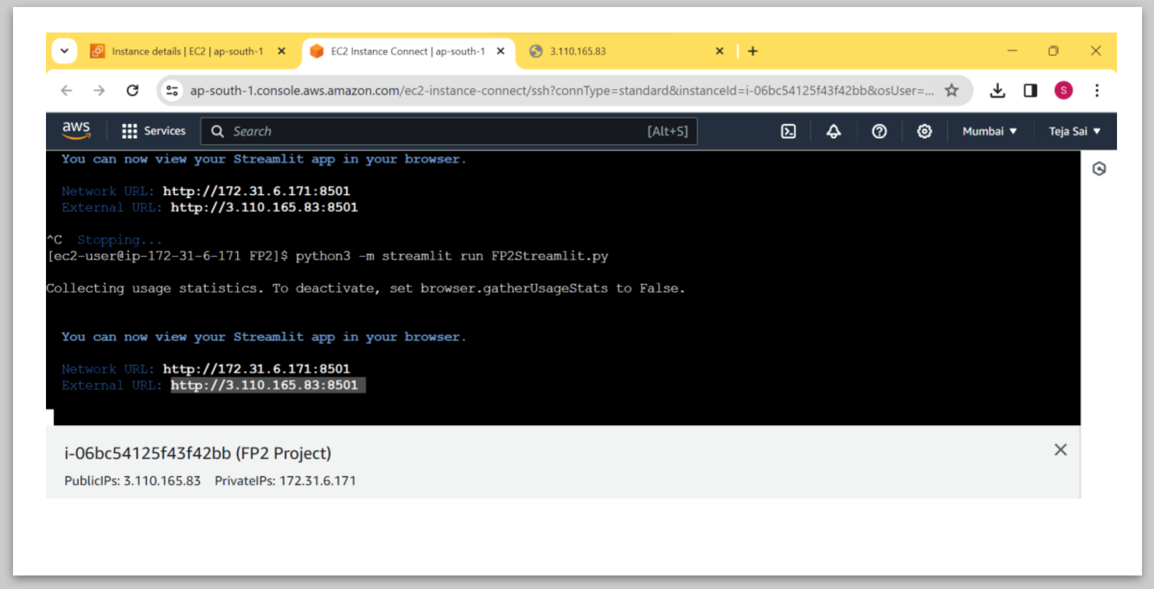
Minimize the Risks of Unforeseen Errors: Set up automated alerts for error logging and data drift with Sagemaker model Monitor to ensure quick rollback or hotfixes.

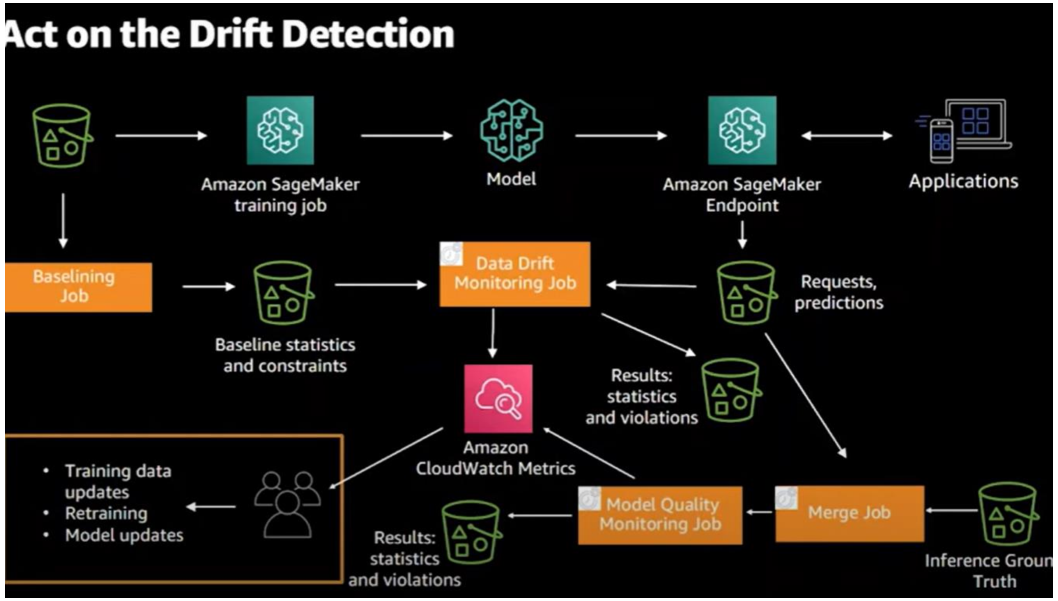
Deployment Strategy: Use AWS EC2 instance to create Streamlit pipeline, enabling smooth deployment for iterative and safe model rollouts.

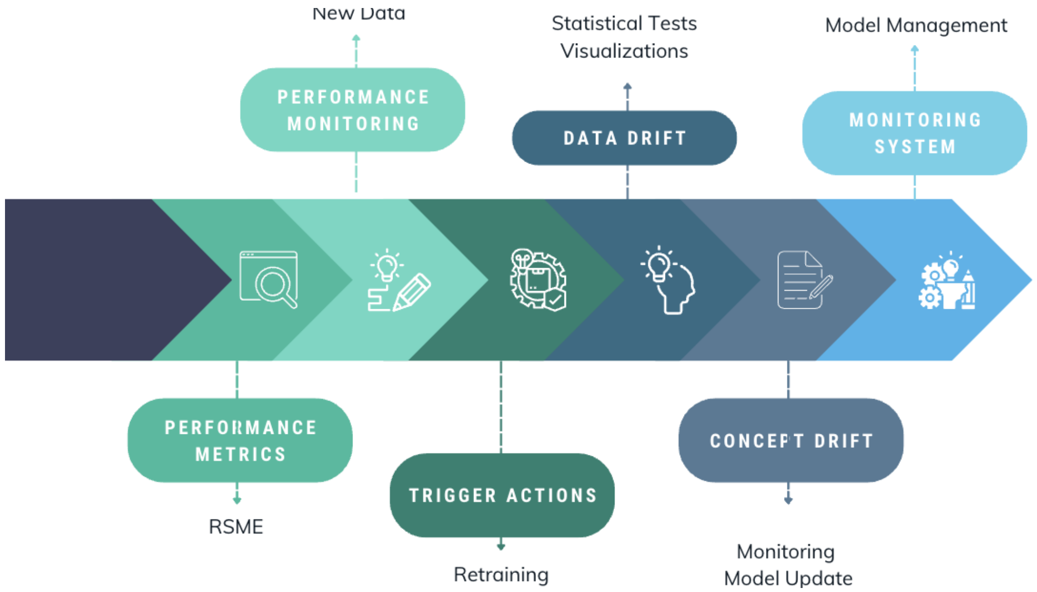

aws
Services
Search
[Alt+S]
Mumbai
Teja Sai
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
FP2 Project
i-06bc54125f43f42bb
Running
t2.micro
2/2 checks passed
Select an instance


EC2 > Instances > i-06bc54125f43f42bb
Instance summary for i-06bc54125f43f42bb (FP2 Project) Info
Connect
Instance state
Actions
Updated less than a minute ago
Instance ID
i-06bc54125f43f42bb (FP2 Project)
Public IPv4 address
3.110.165.83 |open address
Private IPv4 addresses
172.31.6.171
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-3-110-165-83.ap-south-1.compute.amazonaws.com |open address
Hostname type
IP name: ip-172-31-6-171.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-6-171.ap-south-1.compute.internal
Answer private resource DNS name
Instance type
Elastic IP addresses


FP2 Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 branch
0 tags
Go to file
Add file
Code
teja-ai9 Add files via upload
6b4e6d0 now
6 commits
FP2Streamlit.py
Add files via upload
23 minutes ago
Final_POSOCO.csv
Add files via upload
now
Group 4_FP2_XGBoost Code.i...
Add files via upload
now
POSOCO_data.csv
Add files via upload
now
README.md
Initial commit
17 hours ago
app.py
added app.py
17 hours ago
requirements.txt
Update requirements.txt
19 minutes ago
About
Git
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release


Instance details | EC2 | ap-south-1
EC2 Instance Connect | ap-south-1
3.110.165.83
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-06bc54125f43f42bb&osUser=...
aws
Services
Search
[Alt+S]
Mumbai
Teja Sai
You can now view your Streamlit app in your browser.
Network URL: http://172.31.6.171:8501
External URL: http://3.110.165.83:8501
^C Stopping...
[ec2-user@ip-172-31-6-171 FP2]$ python3 -m streamlit run FP2Streamlit.py
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.
You can now view your Streamlit app in your browser.
Network URL: http://172.31.6.171:8501
External URL: http://3.110.165.83:8501
i-06bc54125f43f42bb (FP2 Project)
PublicIPs: 3.110.165.83 PrivateIPs: 172.31.6.171





Monitor & Maintenance

Performance Tracking: Use AWS Sagemaker Model Monitor to oversee model accuracy and trigger alerts for anomalies or performance dips, ensuring quick response to any deviations from expected outcomes.

Automated Retraining: Implement AWS Sagemaker pipeline for automated retraining cycles, ensuring the model evolves with the latest consumption patterns and remains accurate over time.

User Feedback Integration: Create a structured process within the application to collect and analyze user feedback, which is critical for ongoing model refinement and addressing usability concerns.

Version Control: Leverage ML Ops for systematic model versioning, allowing for efficient management of iterations and facilitating smooth rollbacks if needed.