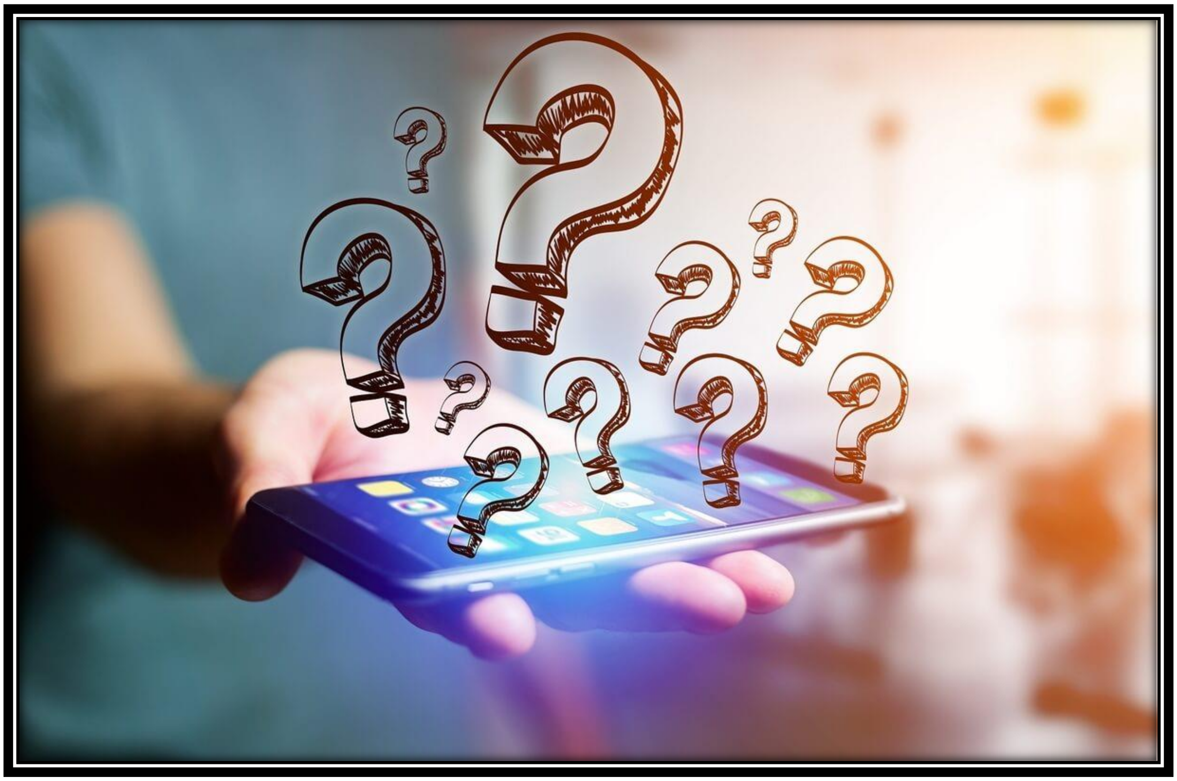
University *of* New Haven

TAGLIATELA COLLEGE OF ENGINEERING

Master of Science in Data Science (MSDS)

Cloud-Based Mobile Price Prediction



SPRING 2024

Contents

1. Abstract:

This report presents the development and deployment of a machine learning model for predicting mobile phone prices. The model utilizes datasets containing information about mobile phone features and prices, with the goal of assisting consumers and businesses in making informed decisions about purchasing and pricing mobile devices.

2. Introduction:

The proliferation of mobile devices has led to a diverse range of features and price points in the market. Predicting mobile phone prices accurately can be valuable for consumers looking to purchase a device within their budget and for businesses seeking to optimize pricing strategies. In this project, we aim to develop a machine learning model capable of predicting mobile phone prices based on their features.

3. Objective:

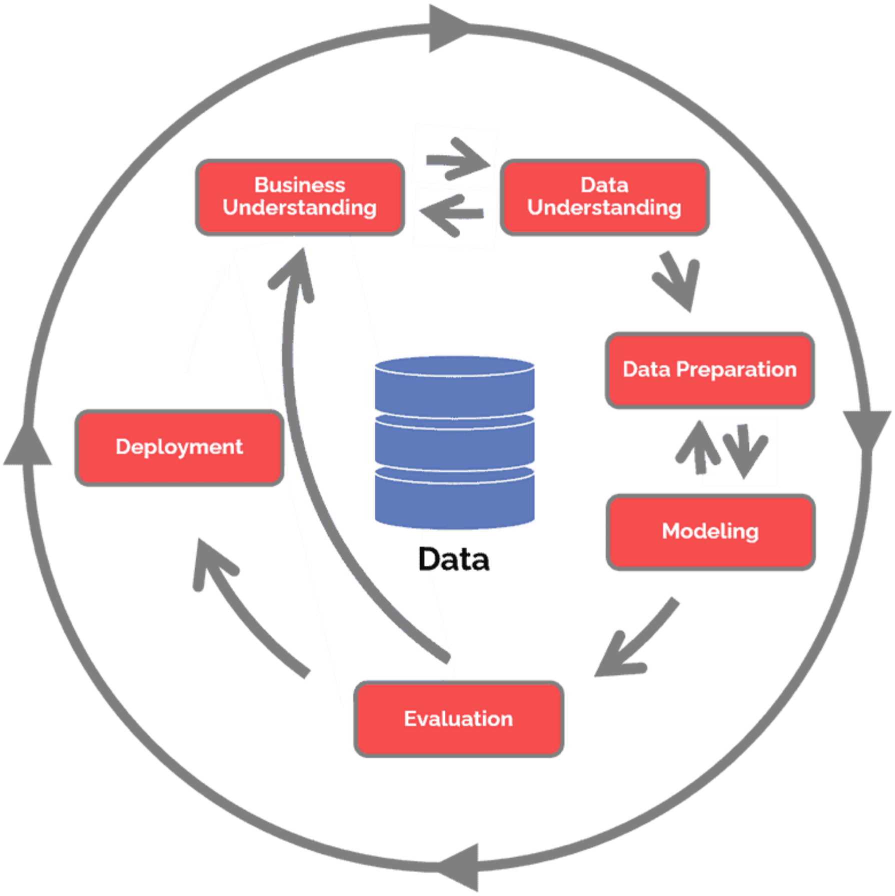
The primary objective of this project is to develop a machine learning model that accurately predicts mobile phone prices based on their features. By leveraging datasets containing information about mobile phone specifications and prices, we aim to create a model that can assist consumers and businesses in making informed decisions about mobile device purchases and pricing strategies.

4. Overview:

The project follows a structured approach, encompassing data gathering, data preparation, exploratory data analysis (EDA), feature selection, model training, model evaluation, and model deployment. Key tools and services utilized include Amazon SageMaker for model development and deployment, Amazon S3 for data storage, and AWS CloudWatch for monitoring pipeline activities.

5. CRISP-DM:

The Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology serves as the framework for this project, guiding the iterative process of data exploration, model development, and deployment.



6. Data Gathering:

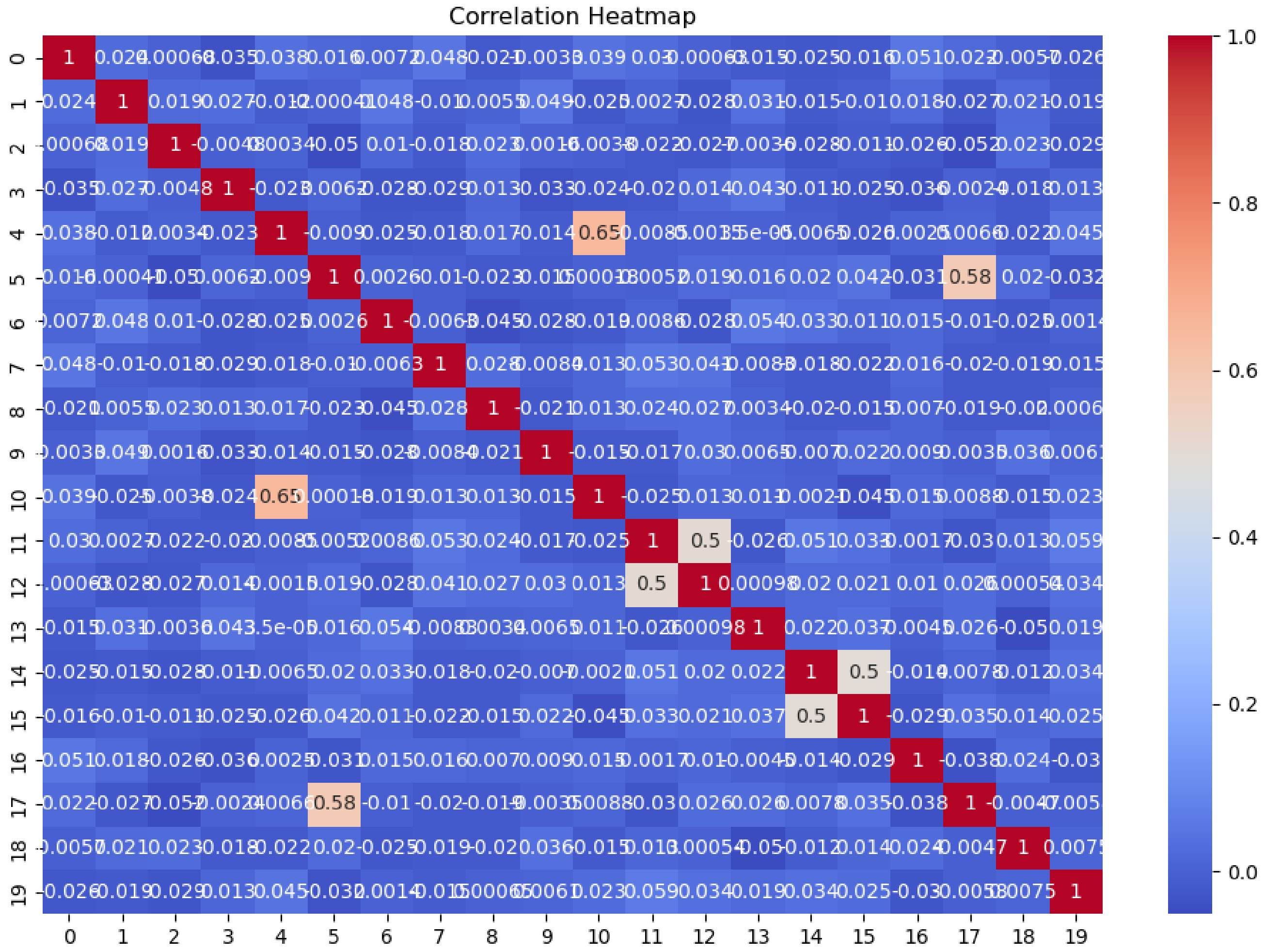
Data gathering involves collecting datasets containing information about mobile phone features and prices. These datasets are stored in Amazon S3 buckets for easy access and scalability. AWS CloudWatch is used to monitor S3 buckets for data availability, access patterns, and storage usage.

7. Data Preparation:

Data preparation includes cleaning and formatting the datasets to ensure they are suitable for model training. This process involves handling missing values, encoding categorical variables, and normalizing numerical features. Python libraries such as Pandas and NumPy are used for data preprocessing tasks, with AWS CloudWatch monitoring the data preparation process.

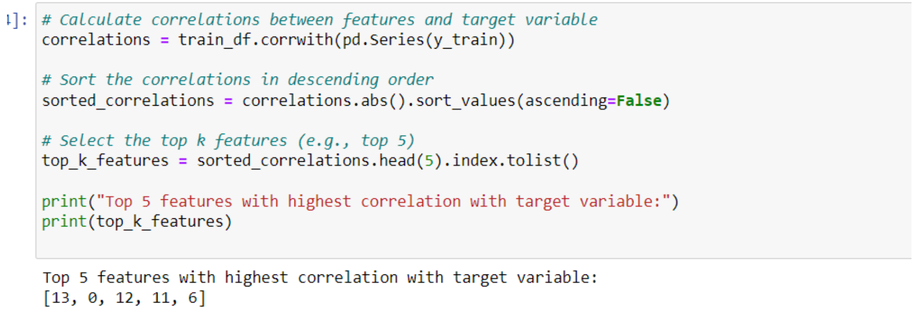
8. Exploratory Data Analysis (EDA):

EDA is performed to gain insights into the datasets and understand the relationships between different features and the target variable (mobile phone prices). Jupyter Notebooks and Python libraries such as Matplotlib and Seaborn are used for visualization and statistical analysis. AWS CloudWatch monitors the execution of EDA scripts, tracking metrics such as notebook execution time and memory usage.



9. Feature Selection:

Feature selection is conducted to improve model performance and efficiency by selecting the most relevant features. Features are chosen based on their importance, as determined by EDA and domain knowledge.

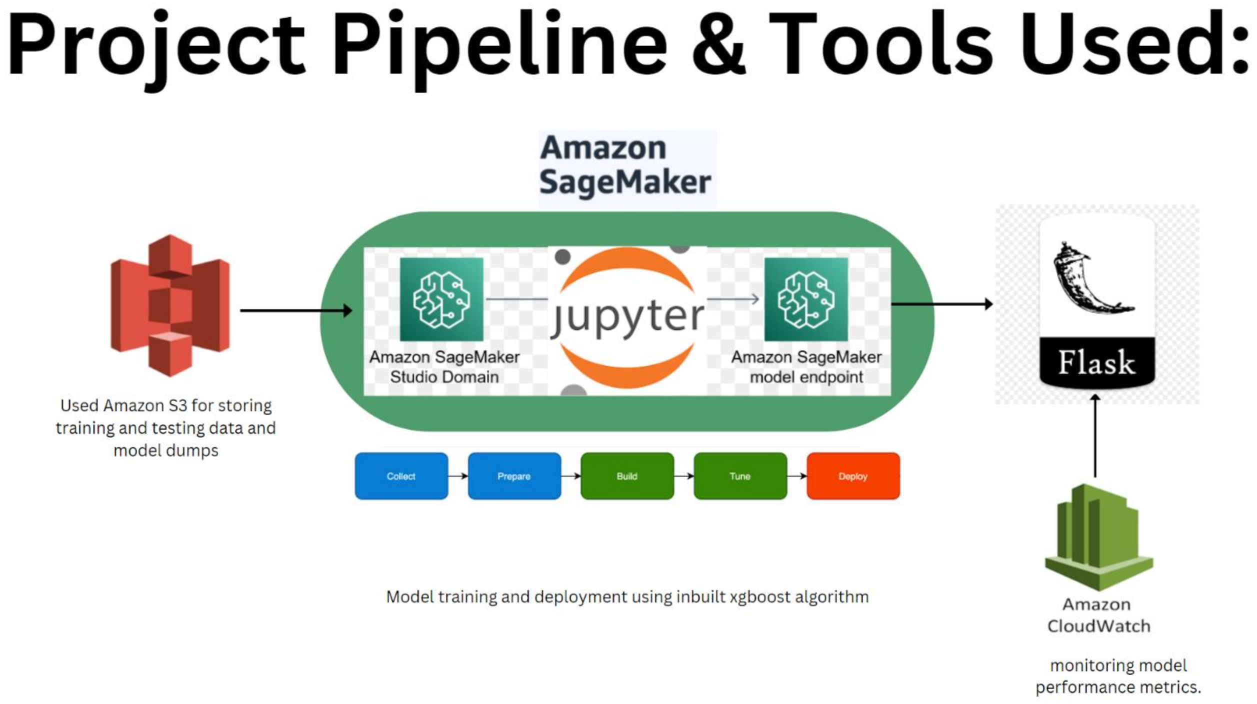


```
# Calculate correlations between features and target variable
correlations = train_df.corrwith(pd.Series(y_train))

# Sort the correlations in descending order
sorted_correlations = correlations.abs().sort_values(ascending=False)

# Select the top k features (e.g., top 5)
top_k_features = sorted_correlations.head(5).index.tolist()

print("Top 5 features with highest correlation with target variable:")
print(top_k_features)
```

```
Top 5 features with highest correlation with target variable:
[13, 0, 12, 11, 6]
```

10. Data Engineering Pipeline:

The data engineering pipeline leverages several AWS services, including Amazon S3 for data storage, Amazon SageMaker for model training and deployment, and AWS CloudWatch for monitoring pipeline activities.



Project Pipeline & Tools Used:

Used Amazon S3 for storing training and testing data and model dumps

Model training and deployment using inbuilt xgboost algorithm

monitoring model performance metrics.

11. Model Training:

The XGBoost algorithm is chosen for model training, a popular choice for regression and classification tasks. Amazon SageMaker provides built-in support for XGBoost, making it easy to train models at scale using distributed computing resources.

11.1 XGBoost Classifier Overview:

XGBoost (Extreme Gradient Boosting) is a widely-used algorithm for classification problems like predicting mobile phone prices. It sequentially builds an ensemble of decision trees, correcting errors made by previous trees to improve accuracy. Key features include regularization to prevent overfitting, parallelization for scalability, and built-in cross-validation for parameter tuning.

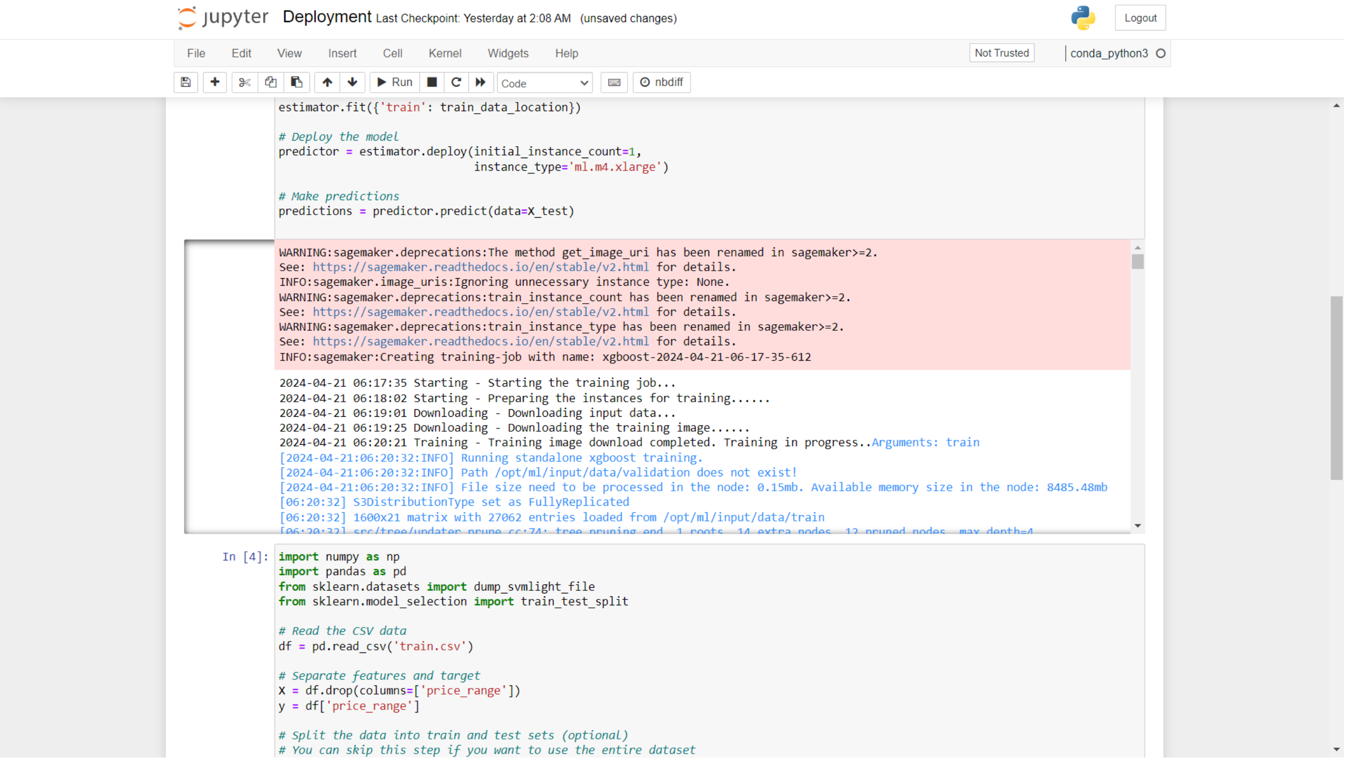
11.2 Advantages:

- Efficiency: XGBoost is fast and scalable, making it suitable for large datasets like mobile phone features.

- Robustness: It handles outliers well and provides insights into feature importance, aiding in interpreting model results.

Use Case:

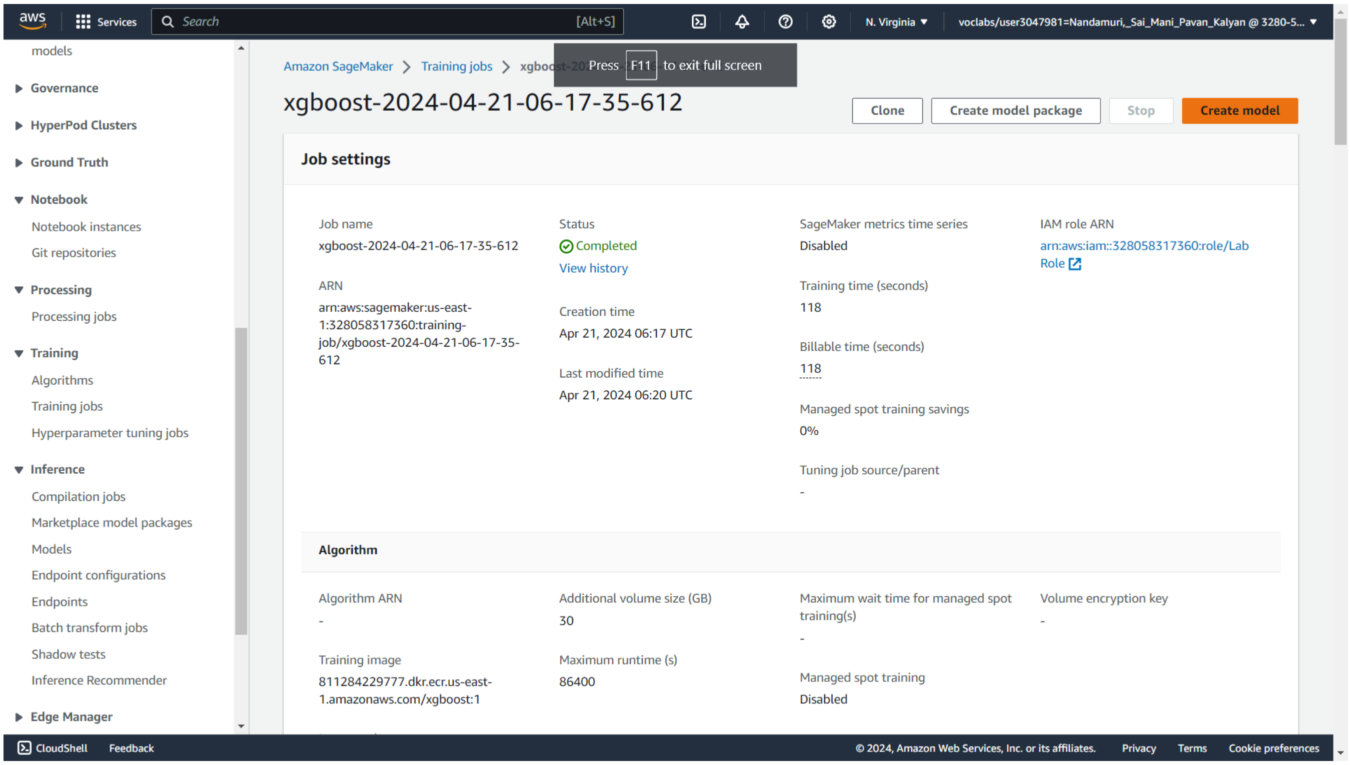
- Mobile Price Prediction: XGBoost can classify mobile phone prices based on features like battery power, camera specs, and connectivity options.



```
estimator.fit({'train': train_data_location})

# Deploy the model
predictor = estimator.deploy(initial_instance_count=1,
                             instance_type='ml.m4.xlarge')

# Make predictions
predictions = predictor.predict(data=X_test)
```

```
WARNING:sagemaker.deprecations:The method get_image_uri has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
WARNING:sagemaker.deprecations:train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
WARNING:sagemaker.deprecations:train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
INFO:sagemaker:Creating training-job with name: xgboost-2024-04-21-06-17-35-612

2024-04-21 06:17:35 Starting - Starting the training job...
2024-04-21 06:18:02 Starting - Preparing the instances for training......
2024-04-21 06:19:01 Downloading - Downloading input data...
2024-04-21 06:19:25 Downloading - Downloading the training image......
2024-04-21 06:20:21 Training - Training image download completed. Training in progress..Arguments: train
[2024-04-21:06:20:32:INFO] Running standalone xgboost training.
[2024-04-21:06:20:32:INFO] Path /opt/ml/input/data/validation does not exist!
[2024-04-21:06:20:32:INFO] File size need to be processed in the node: 0.15mb. Available memory size in the node: 8485.48mb
[06:20:32] S3DistributionType set as FullyReplicated
[06:20:32] 1600x21 matrix with 27062 entries loaded from /opt/ml/input/data/train
[06:20:32] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 14 extra nodes, 12 pruned nodes, max_depth=4
```

```
In [4]: import numpy as np
        import pandas as pd
        from sklearn.datasets import dump_svmlight_file
        from sklearn.model_selection import train_test_split

        # Read the CSV data
        df = pd.read_csv('train.csv')

        # Separate features and target
        X = df.drop(columns=['price_range'])
        y = df['price_range']

        # Split the data into train and test sets (optional)
        # You can skip this step if you want to use the entire dataset
```



xgboost-2024-04-21-06-17-35-612

Job settings

Job name: xgboost-2024-04-21-06-17-35-612

ARN: arn:aws:sagemaker:us-east-1:328058317360:training-job/xgboost-2024-04-21-06-17-35-612

Status: Completed

Creation time: Apr 21, 2024 06:17 UTC

Last modified time: Apr 21, 2024 06:20 UTC

SageMaker metrics time series: Disabled

Training time (seconds): 118

Billable time (seconds): 118

Managed spot training savings: 0%

Tuning job source/parent: -

IAM role ARN: arn:aws:iam::328058317360:role/LabRole

Algorithm

Algorithm ARN: -

Training image: 811284229777.dkr.ecr.us-east-1.amazonaws.com/xgboost:1

Additional volume size (GB): 30

Maximum runtime (s): 86400

Maximum wait time for managed spot training(s): -

Managed spot training: Disabled

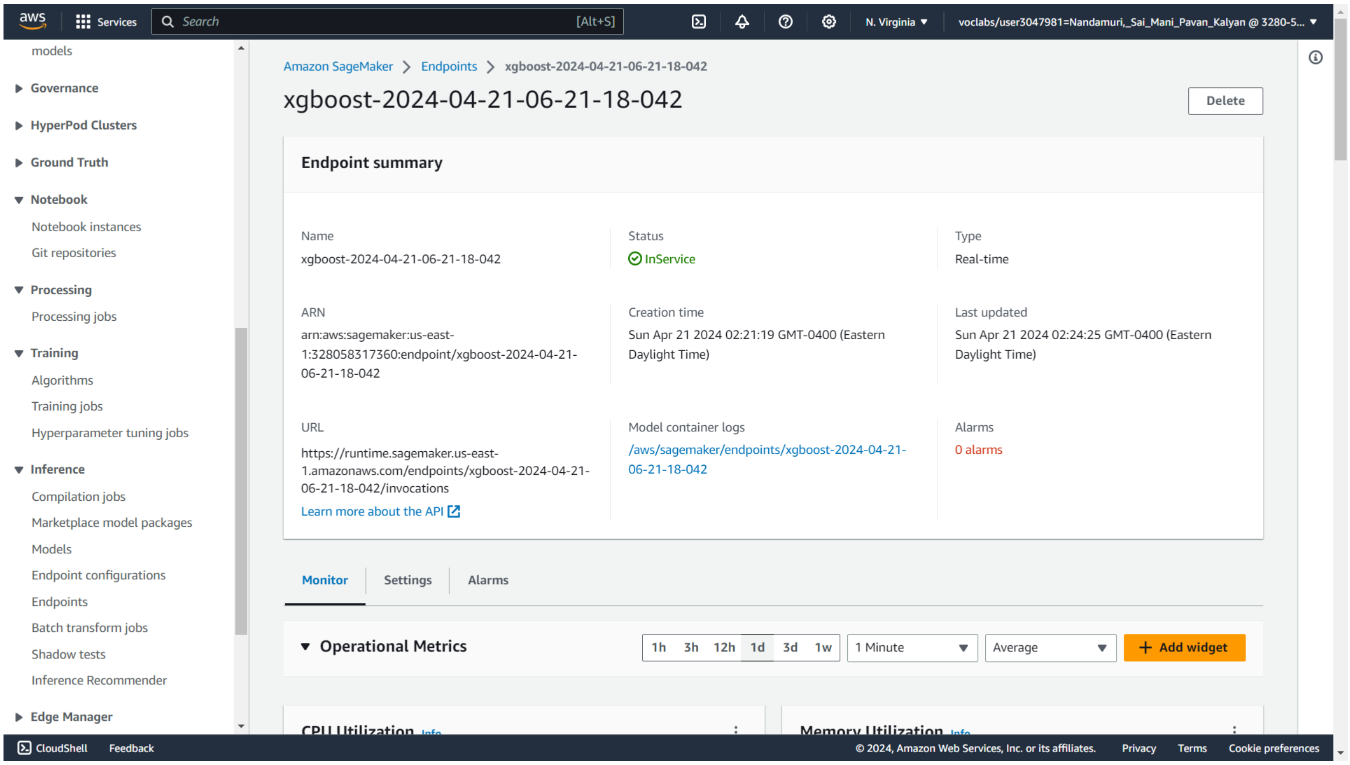
Volume encryption key: -

12. Model Evaluation:

The trained model is evaluated using validation data to assess its performance and generalization capabilities. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure the effectiveness of the model in predicting mobile phone prices.

13. Model Deployment:

Once the model is trained and evaluated, it is deployed to a SageMaker endpoint, allowing real-time inference on new data. The endpoint is configured with appropriate settings to ensure optimal performance and cost efficiency.



xgboost-2024-04-21-06-21-18-042

Endpoint summary

Name
xgboost-2024-04-21-06-21-18-042

Status
InService

Type
Real-time

ARN
arn:aws:sagemaker:us-east-1:328058317360:endpoint/xgboost-2024-04-21-06-21-18-042

Creation time
Sun Apr 21 2024 02:21:19 GMT-0400 (Eastern Daylight Time)

Last updated
Sun Apr 21 2024 02:24:25 GMT-0400 (Eastern Daylight Time)

URL
https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/xgboost-2024-04-21-06-21-18-042/invocations
Learn more about the API

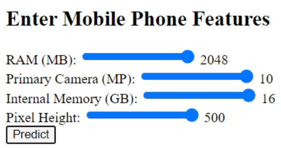
Model container logs
/aws/sagemaker/endpoints/xgboost-2024-04-21-06-21-18-042

Alarms
0 alarms

▼ Operational Metrics

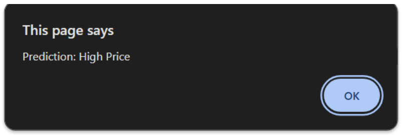
1 Minute ▼
Average ▼

Deployed Flask App: URL



Enter Mobile Phone Features

RAM (MB): 2048
Primary Camera (MP): 10
Internal Memory (GB): 16
Pixel Height: 500

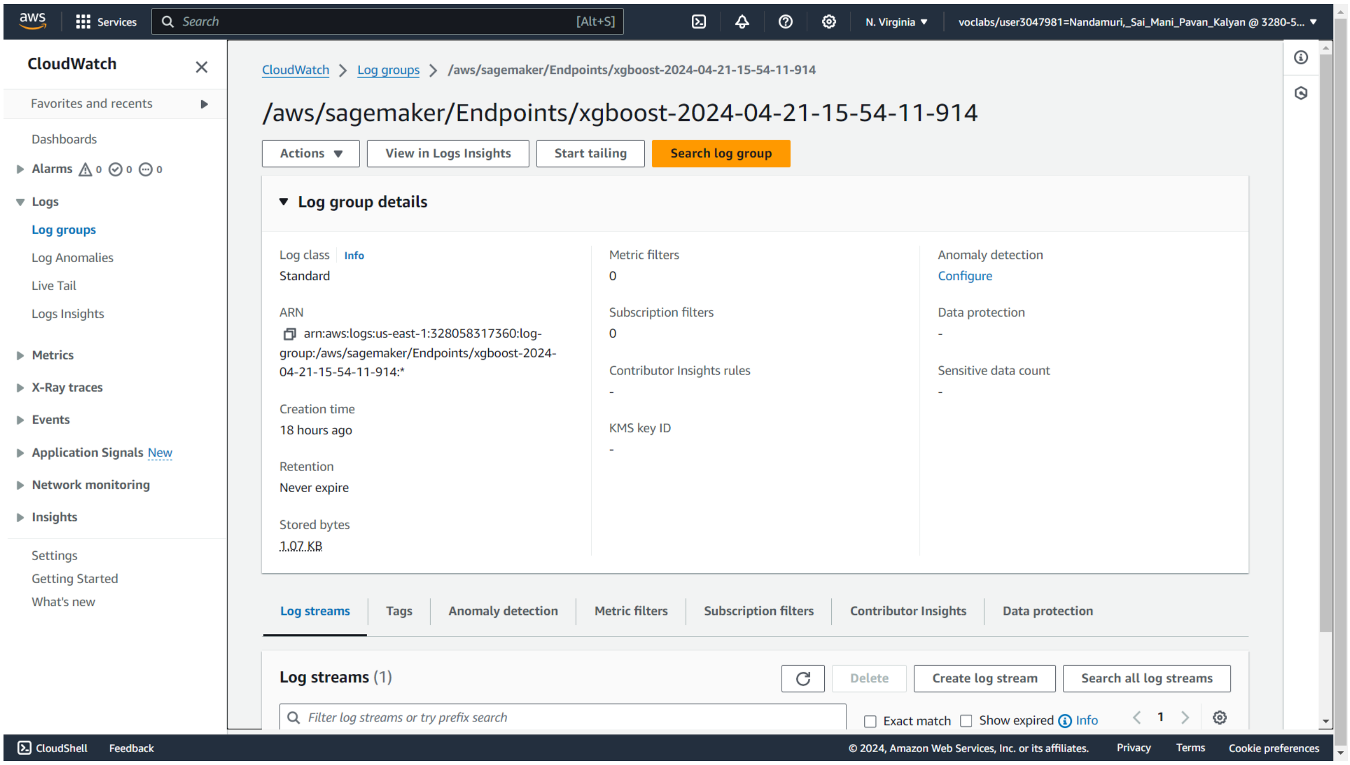


This page says

Prediction: High Price

14. Monitoring and Management:

The deployed endpoint and pipeline activities are monitored using Amazon CloudWatch, providing insights into performance, resource utilization, and system health. CloudWatch enables proactive monitoring and alerting for any issues that arise during model training, evaluation, and deployment.



CloudWatch > Log groups > /aws/sagemaker/Endpoints/xgboost-2024-04-21-15-54-11-914

/aws/sagemaker/Endpoints/xgboost-2024-04-21-15-54-11-914

Log group details

Log class Info
Standard

ARN
arn:aws:logs:us-east-1:328058317360:log-group:/aws/sagemaker/Endpoints/xgboost-2024-04-21-15-54-11-914:*

Creation time
18 hours ago

Retention
Never expire

Stored bytes
1.07 KB

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

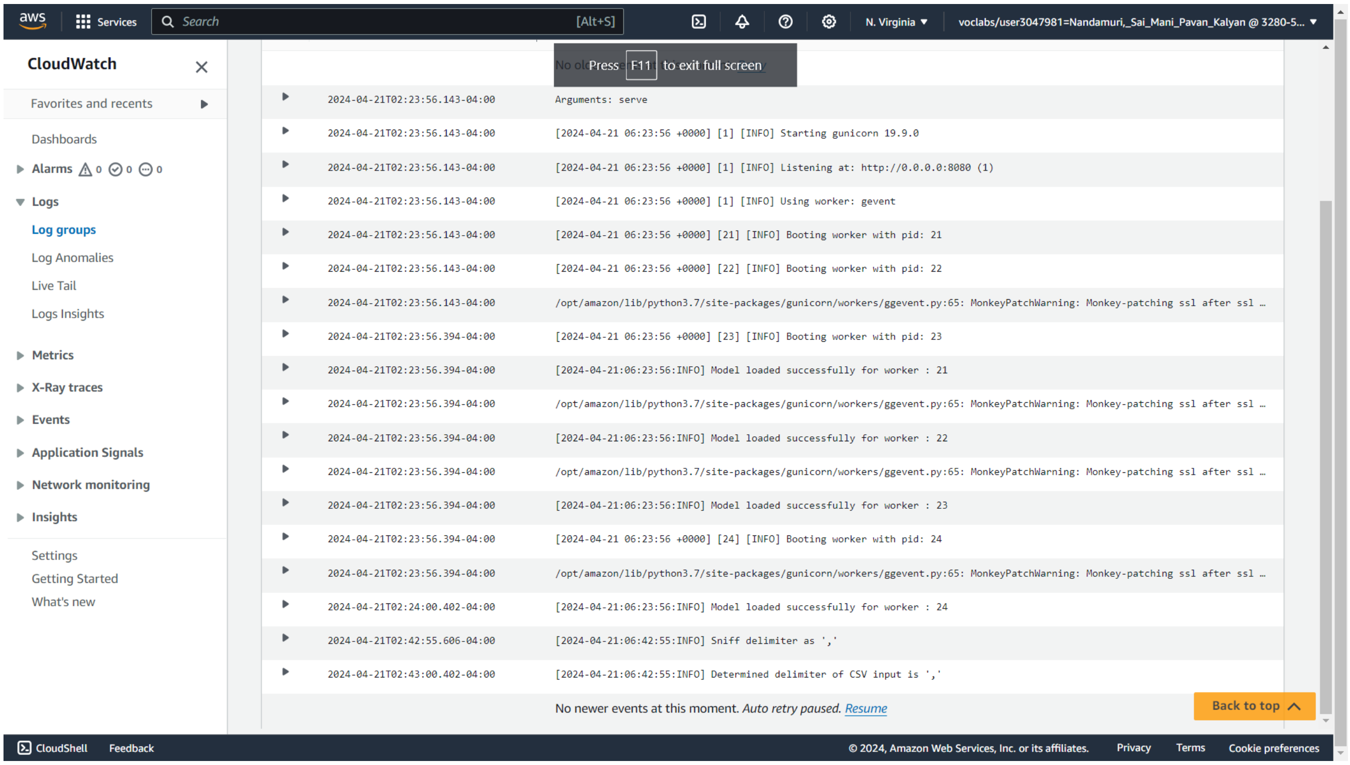
KMS key ID
-

Anomaly detection
Configure

Data protection
-

Sensitive data count
-

Log streams (1)



```
2024-04-21T02:23:56.143-04:00    Arguments: serve
2024-04-21T02:23:56.143-04:00    [2024-04-21 06:23:56 +0000] [1] [INFO] Starting gunicorn 19.9.0
2024-04-21T02:23:56.143-04:00    [2024-04-21 06:23:56 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
2024-04-21T02:23:56.143-04:00    [2024-04-21 06:23:56 +0000] [1] [INFO] Using worker: gevent
2024-04-21T02:23:56.143-04:00    [2024-04-21 06:23:56 +0000] [21] [INFO] Booting worker with pid: 21
2024-04-21T02:23:56.143-04:00    [2024-04-21 06:23:56 +0000] [22] [INFO] Booting worker with pid: 22
2024-04-21T02:23:56.143-04:00    /opt/amazon/lib/python3.7/site-packages/gunicorn/workers/ggevent.py:65: MonkeyPatchWarning: Monkey-patching ssl after ssl …
2024-04-21T02:23:56.394-04:00    [2024-04-21 06:23:56 +0000] [23] [INFO] Booting worker with pid: 23
2024-04-21T02:23:56.394-04:00    [2024-04-21:06:23:56:INFO] Model loaded successfully for worker : 21
2024-04-21T02:23:56.394-04:00    /opt/amazon/lib/python3.7/site-packages/gunicorn/workers/ggevent.py:65: MonkeyPatchWarning: Monkey-patching ssl after ssl …
2024-04-21T02:23:56.394-04:00    [2024-04-21:06:23:56:INFO] Model loaded successfully for worker : 22
2024-04-21T02:23:56.394-04:00    /opt/amazon/lib/python3.7/site-packages/gunicorn/workers/ggevent.py:65: MonkeyPatchWarning: Monkey-patching ssl after ssl …
2024-04-21T02:23:56.394-04:00    [2024-04-21:06:23:56:INFO] Model loaded successfully for worker : 23
2024-04-21T02:23:56.394-04:00    [2024-04-21 06:23:56 +0000] [24] [INFO] Booting worker with pid: 24
2024-04-21T02:23:56.394-04:00    /opt/amazon/lib/python3.7/site-packages/gunicorn/workers/ggevent.py:65: MonkeyPatchWarning: Monkey-patching ssl after ssl …
2024-04-21T02:24:00.402-04:00    [2024-04-21:06:23:56:INFO] Model loaded successfully for worker : 24
2024-04-21T02:42:55.606-04:00    [2024-04-21:06:42:55:INFO] Sniff delimiter as ','
2024-04-21T02:43:00.402-04:00    [2024-04-21:06:42:55:INFO] Determined delimiter of CSV input is ','
```

No newer events at this moment. *Auto retry paused.* Resume

15. Conclusion:

In conclusion, the project demonstrates the end-to-end process of developing and deploying a machine learning model for predicting mobile phone prices. By leveraging AWS services and tools, we create a scalable and efficient solution that can assist consumers and businesses in making informed decisions about mobile device purchases and pricing strategies.

16. Future Work:

Future work may include exploring additional features and refining the model further to improve prediction accuracy. Additionally, optimization of the deployment architecture for enhanced scalability and cost efficiency could be pursued.

17. References:

1. [Amazon SageMaker Documentation](https://docs.aws.amazon.com/sagemaker/index.html)
2. [CRISP-DM Methodology Documentation](https://www.the-modeling-agency.com/crisp-dm.pdf)
3. [XGBoost Documentation](https://xgboost.readthedocs.io/en/latest/)
4. [AWS CloudWatch Documentation](https://docs.aws.amazon.com/cloudwatch/index.html)
5. [AWS CloudTrail Documentation](https://docs.aws.amazon.com/cloudtrail/index.html)
6. [AWS SageMaker SGBoost] https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html