

Customer Churn Prediction using AWS Kinesis and SageMaker

Ali Guzelyel
A20454373

Avinash Sankar
A20525471

H. V. Veerakkumaar
A20528175

Taufeeq A. Mohammed
A20512082

CSP 554 - Big Data Technologies

{aguzelyel, asankar5, hvelampalayamveerakk, tmohammed1} @hawk.iit.edu

ABSTRACT

Big Data has become part of our lives. Along with it, came the increase in supply of many highly accurate Artificial Intelligence models, which use existing data to generate new data or classify, predict, or estimate existing data. As such, many companies have also shown interest in integrating AI into their applications and company tech stack. With this project, we explore how a company with a continuous data stream can utilize Amazon Web Services systems such as Kinesis Data Streams, S3, and SageMaker to create an end-to-end Machine Learning pipeline, starting from data transfer-compilation until model prediction and analysis. In this paper, we've managed to gather 7000+ user churn data¹ in separate JSON files through Kinesis Data Streams into S3, pre-process with PySpark, further pre-process with Data Wrangler, and build an ML model with SageMaker. Using the quick-build option, we were able to get a classification model within ten minutes (build time) with an overall 76% accuracy, and an F-score of 0.642, without having to tweak hyperparameters or define the type of ML model. The main purpose of this paper is to show how we approached the problem, the AWS architecture we came up with, and how the AWS systems make this convenient. Model accuracy is not a metric for this project.

Screenshots of the progress are given at the end of the paper.

¹<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

1. INTRODUCTION

In this project, we are tackling how to use AWS systems to create an end-to-end Machine Learning pipeline, starting from gathering data in streams, to compiling the data, profiling it, and then using it as training data for an ML model. We are working on a user churn dataset published by TELCO that consists of 7000+ user data with 21 columns in labels such as monthly charges, tenure (in months), payment method, etc. The dataset has the target label as the column “churn”, which means whether the user has canceled their subscription or not. In the following sections, we talk in-depth about the technologies used from AWS to create the end-to-end pipeline, how we simulated a data stream, and how we built the machine learning model.

2. TECHNOLOGIES

2.1 AWS Command Line Interpreter (CLI)

The Amazon Web Services Command Line Interpreter, or AWS CLI², is a centralized tool that allows users to communicate with different AWS services straight from the command line. It provides a command-line interface that enables users to manage and administer AWS services and resources in a variety of ways without requiring the usage of the AWS Management Console.

The real-time churn prediction model was developed and deployed using Amazon SageMaker, with essential components controlled

² <https://aws.amazon.com/cli/>

via the AWS CLI. Throughout the project, the AWS CLI made many jobs easier, improving automation and administration simplicity. The AWS CLI made it easier to deploy the churn prediction model on SageMaker. The model was seamlessly integrated into the prediction system thanks to the ease with which configurations and artifacts from the model could be uploaded to SageMaker and the creation of real-time endpoints using a set of CLI commands.

2.2 AWS Kinesis Data Streams

Building custom applications for the real-time processing of streaming data is made possible by Amazon Web Services (AWS) through its managed solution, Amazon Kinesis Data Streams³. Accompanying it is Kinesis Video Streams, Kinesis Data Analytics, and Kinesis Data Firehose, as well as the larger family of Amazon Kinesis services. Applications, analytics, and monitoring systems are a few examples of situations where Kinesis Data Streams come in handy when you need to ingest, process, and analyze data in real-time.

Our real time customer churn prediction system's architecture was greatly influenced by Amazon Kinesis Data Streams. By giving us the basic framework for consuming and analyzing streaming data, it enabled us to model dynamic data flows and show off the possibilities of our real-time prediction model. AWS services and Kinesis Data Streams work together smoothly to build a full pipeline for real-time data processing. As consumers, AWS Lambda functions processed data records and initiated actions according to the churn projections.

2.3 AWS Kinesis Firehose

AWS's completely managed solution for ingesting, converting, and loading streaming data into different AWS storage and analytics services is called Amazon Kinesis Data Firehose⁴. It makes importing streaming data easier and makes near

real-time analytics possible. For situations where you need to consistently and economically transport streaming data to locations like Amazon S3, Amazon Redshift, Amazon Elasticsearch, and more, Kinesis Data Firehose is an essential component.

Our real-time customer churn prediction system relies heavily on Amazon Kinesis Data Firehose, which provides a dependable and scalable means of ingesting, converting, and sending streaming data to many AWS destinations. The main features of its usage and integration in our project are described in this section. Amazon Kinesis Data Firehose proved to be a valuable component in our real-time customer churn prediction system. Its ease of use, seamless integration with other AWS services, and capabilities for data transformation made it an ideal choice for ingesting and delivering streaming data, contributing to the success of our proof of concept.

2.4 AWS S3

Amazon Web Services (AWS) offers scalable and secure object storage through its Amazon Simple Storage Service (Amazon S3)⁵. Any quantity of data can be stored and retrieved from any location on the internet. A key component of cloud computing, Amazon S3 provides high availability, low latency, and durability for access to stored objects. Data archiving, data backup and recovery, data storage, and delivering static assets for web applications are just a few of the many use cases for which it is frequently employed.

The main storage option used by our real-time customer churn prediction system was Amazon S3. It was used to store processing results, model artifacts, and streaming data thanks to its scalable and robust object storage characteristics. The main features of our project's usage of Amazon S3 are described in this section. Amazon Kinesis Data Firehose delivery stream to

³ <https://aws.amazon.com/kinesis/>

⁴ <https://aws.amazon.com/kinesis/data-firehose/>

⁵ <https://aws.amazon.com/s3/>

send the data to an S3 bucket. This allows you to capture and persist the streaming data in a scalable and durable storage solution.

2.5 Apache Spark - PySpark

Apache Spark⁶ is an open-source distributed computing platform that offers a quick and flexible cluster computing framework for handling large amounts of data. The Python API for Apache Spark is called PySpark, and it lets Python programmers leverage Spark features with Python code. It makes it simpler for Python programmers to take advantage of Spark's capability by offering high-level Python APIs for distributed data processing.

With its strong distributed computing capabilities, Apache Spark was essential to the development of our real-time customer churn prediction system. PySpark, the Python API for Apache Spark, made it possible to seamlessly integrate Spark's features with Python, letting us take advantage of its machine learning libraries and processing power inside our Python-based system. For activities including data transformation and feature engineering, PySpark's DataFrame API was utilized. In order to improve the input data for the churn prediction model, this involved choosing pertinent features, dealing with missing values, and developing new features.

2.6 AWS SageMaker & Data Wrangler

A visual interface called Amazon SageMaker⁷ Data Wrangler⁸ is used to prepare data for machine learning (ML) projects. Data scientists and analysts may more effectively prepare datasets for machine learning model training thanks to its ability to streamline the processes of data exploration, cleansing, and transformation. Our real-time customer churn prediction system relies heavily on Amazon

SageMaker Data Wrangler, which offers an easy-to-use and effective environment for data preparation.

SageMaker Data Wrangler simplifies the process of importing datasets from Amazon S3, it enables intuitive data cleaning operations. Also, Data Wrangler facilitates export of preprocessed data to S3 bucket making it readily available for training ML models using SageMaker.

SageMaker is a fully managed machine learning platform that lets data scientists and developers quickly create, train, and deploy models using machine learning in a hosting environment that is fully functional and operational. The integration of a Jupyter notebook instance into SageMaker makes it easier to explore and analyze data sources and does away with the need for server management. It also has improved machine learning algorithms that can work well in a distributed environment with massive amounts of data. Numerous machine learning problems are covered by these techniques, including regression, classification, clustering, and anomaly detection. Along with native support for bespoke frameworks and algorithms, SageMaker also offers adaptable distributed training options tailored to specific processes.

3. IMPLEMENTATION

3.1 Server Architecture

The application that we implemented in this project consists of many distributed systems components. First, we implemented an AWS Kinesis Data Streams stream to obtain streams of data from the local application in time intervals. The Data Streams allowed us to collect data that was sent over in parts at different times. Second, we connected an AWS Kinesis Firehose to the Data Streams, which allowed us to direct the data received on the Data Streams to another platform. Third, the Kinesis Firehose was connected to an AWS S3 bucket. The S3 bucket allowed us to

⁶ <https://spark.apache.org/>

⁷ <https://aws.amazon.com/pm/sagemaker/>

⁸ <https://aws.amazon.com/sagemaker/data-wrangler/>

store the streaming data safely in objects/chunks. Fourth, by using a Spark EMR node, we were able to compile the data in chunks to a single CSV file. Finally, we imported the compiled CSV data file into the AWS SageMaker. Inside SageMaker, we used Data Wrangler to do data profiling and then model training. *(A screenshot of the architecture is given at the end of the paper, Figure 8.0)*

Additionally, we've implemented a company-like hierarchy inside the AWS platform, by creating "Data Scientist" users with restricted access, who were allowed to work on EMR, S3, Kinesis, and SageMaker, but were restricted from accessing any other technology within the platform.

3.2 Dataset

We explore the churn possibilities on a telecommunication service churn dataset, which consists of 7000+ service users with 21 columns: CustomerID, Gender, seniorCitizen, Partner, Dependents, Tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, Contract-type, PaperlessBill, PaymentMethod, MonthlyCharges, TotalCharges, and Churn (label) . Based on these columns, we can initiate a good machine-learning model that will be able to do classic classification for whether a user has the potential to stop using the service or not. Since this project is on real-time data, we first need to turn the single CSV file into multiple streaming data to be sent to the AWS servers. We can turn the single CSV file into a set of JSON files that consist of data of only one user per file. This allows us to simulate real-time streaming data by sending the data from a local machine (or a real-life application) to the AWS server in duration intervals.

3.3 Data Profiling

Given that the churn dataset shows real-life data, it is probable to have faulty rows of data rows inside. For this reason, we implemented data profiling on the dataset using PySpark from

SageMaker Data Wrangler. Data Wrangler allows us to work directly on the data with custom PySpark scripts from the console. First, we remove the customerID column from the dataset to remove bias from the training model. Then we removed the rows that contain null values to obtain consistency with the data. Additionally, we have observed that some users' data contained "tenure=0", which was also removed from the dataset. Finally, we standardized each column from string values to one-hot encoding-like formats. For instance, the column "InternetService" consists of labels: "No, DSL, Fiber optic", which were standardized into labels: "0, 1, 2" to allow better machine learning model training.

Each column was then put into a correlation table, to find columns that are correlated with each other, in order to decide which columns matter more for the model predictions.

4. EXPERIMENT & RESULTS

4.1 Quick Build

After setting up all the AWS architectures, and dealing with data, we directed the data into the AWS SageMaker Models by importing from the S3 bucket. From this console, we dropped some of the columns that had low correlation with the "churn" label, and normalized the columns: "total_charges, tenure, and monthly_charges". By doing so, we restricted the scale of the data inside these columns. AWS SageMaker allows for a very easy model building option, and we chose Quick Build to build the classification model. By just waiting for the model to finish training, we received a model that was able to predict with an *overall 76.26% accuracy and a 0.642 F-score*.

Here is the confusion matrix showing statistics for *true positive, false positive, true negative, and false negative (Figure X)*:

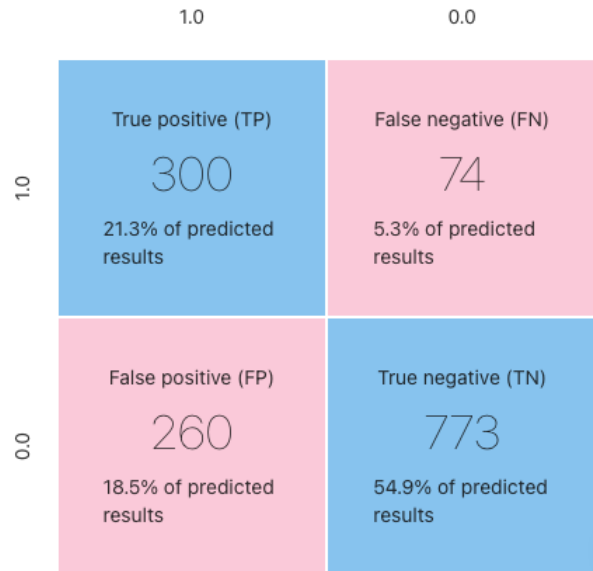


Figure X: Confusion matrix, 1: Churn-yes, 0: Churn-no

6. CONCLUSION & FUTURE WORK

In this project, we explored how AWS systems can be used to train a machine learning model using real-time streaming data. We have observed that the technologies residing in AWS system have great utilities for Data Scientists and Machine Learning Engineers who are interested in researching and/or analyzing on data that is produced by daily-life applications. There's great promise in being able to train models using streaming data, and AWS systems allow a company to establish an end-to-end solution from data gathering to data profiling to data preprocessing and model training. For future work, we'd like to see how some parts of the project could have been automated using AWS Lambda, and/or other systems in order to create a perfect seamless data flow from gathering to model training. We'd also like to try other normalization techniques for the columns, and tweaking the hyperparameters of the model.

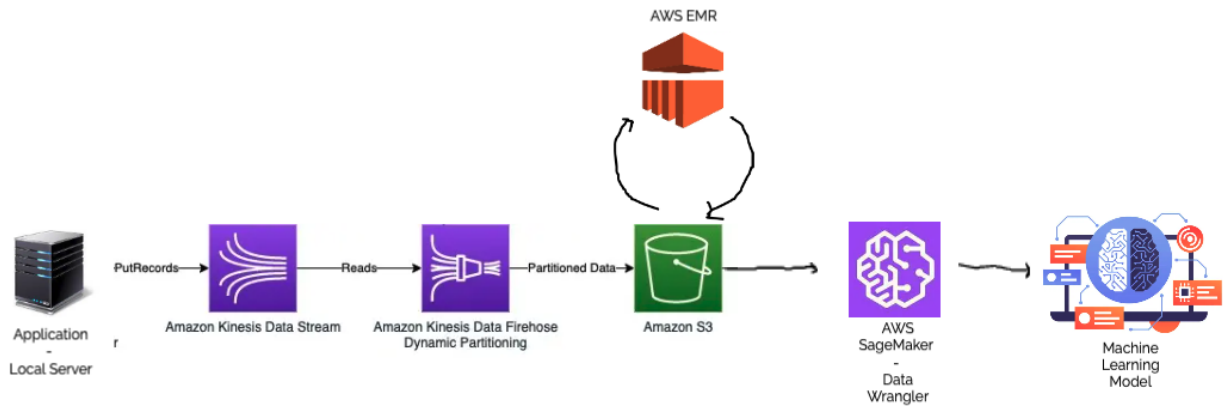
7. REFERENCES

- [1] <https://www.kaggle.com/datasets/blastchar/telco-customer-churn>
- [2] <https://aws.amazon.com/cli/>
- [3] <https://aws.amazon.com/kinesis/>
- [4] <https://aws.amazon.com/kinesis/data-firehose/>
- [5] <https://aws.amazon.com/s3/>
- [6] <https://spark.apache.org/>
- [7] <https://aws.amazon.com/pm/sagemaker/>
- [8] <https://aws.amazon.com/sagemaker/data-wrangler/>
- [9] <https://spark.apache.org/docs/latest/api/python/index.html>
- [10] <https://docs.aws.amazon.com/sagemaker/latest/dg/data-wrangler-data-flow.html>

8. SCREENSHOTS

Starts from the next page.

8.0 AWS - Utilized Systems Architecture



8.1 Organizational Roles

8.1.1 Adding a “Data Scientist” role to the organization

aws Services Search [Option+S]

[IAM Identity Center](#) > [Permission sets](#) > [DataScientist](#) > Edit inline policy

Edit the DataScientist permission set

You can edit this JSON document whenever you need to add or remove IAM permissions policies for this permissions set. [Learn more](#)

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Action": [
8         "s3:*"
9       ],
10      "Resource": [
11        "arn:aws:s3:::churn-check-s3"
12      ]
13    },
14    {
15      "Sid": "Statement2",
16      "Effect": "Allow",
17      "Action": [
18        "elasticmapreduce:*"
19      ],
20      "Resource": [
21        "*"
22      ]
23    },
24    {
25      "Sid": "Statement3",
26      "Effect": "Allow",
27      "Action": [
28        "sagemaker:*"
29      ]
30    }
31  ]
32 }
```

[+ Add new statement](#)

JSON Ln 25, Col 21

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Cancel [Save changes](#)

Edit statement Statement3

Remove

Add actions

All services > SageMaker

- ☒ All actions (sagemaker:*)



Access level - list


- ☒ ListActions Info
- ☒ ListAlgorithms Info
- ☒ ListAliases Info
- ☒ ListAppImageConfigs Info
- ☒ ListApps Info
- ☒ ListArtifacts Info
- ☒ ListAssociations Info
- ☒ ListAutoMLJobs Info
- ☐ ListCandidatesForAutoMLJob Info

Add a resource [Add](#)

Add a condition (optional) [Add](#)

8.1.2 “Data Scientist” role cannot access restricted systems

  Services [Option+S]






 **You need permissions**
You do not have the permission required to perform this operation. Ask your administrator to add permissions.
You don't have permissions to access this resource.


8.2 Connected to AWS through AWS CLI

8.3 Creating an S3 Bucket to store data in chunks

General purpose buckets | Directory buckets

General purpose buckets (5) [Info](#)

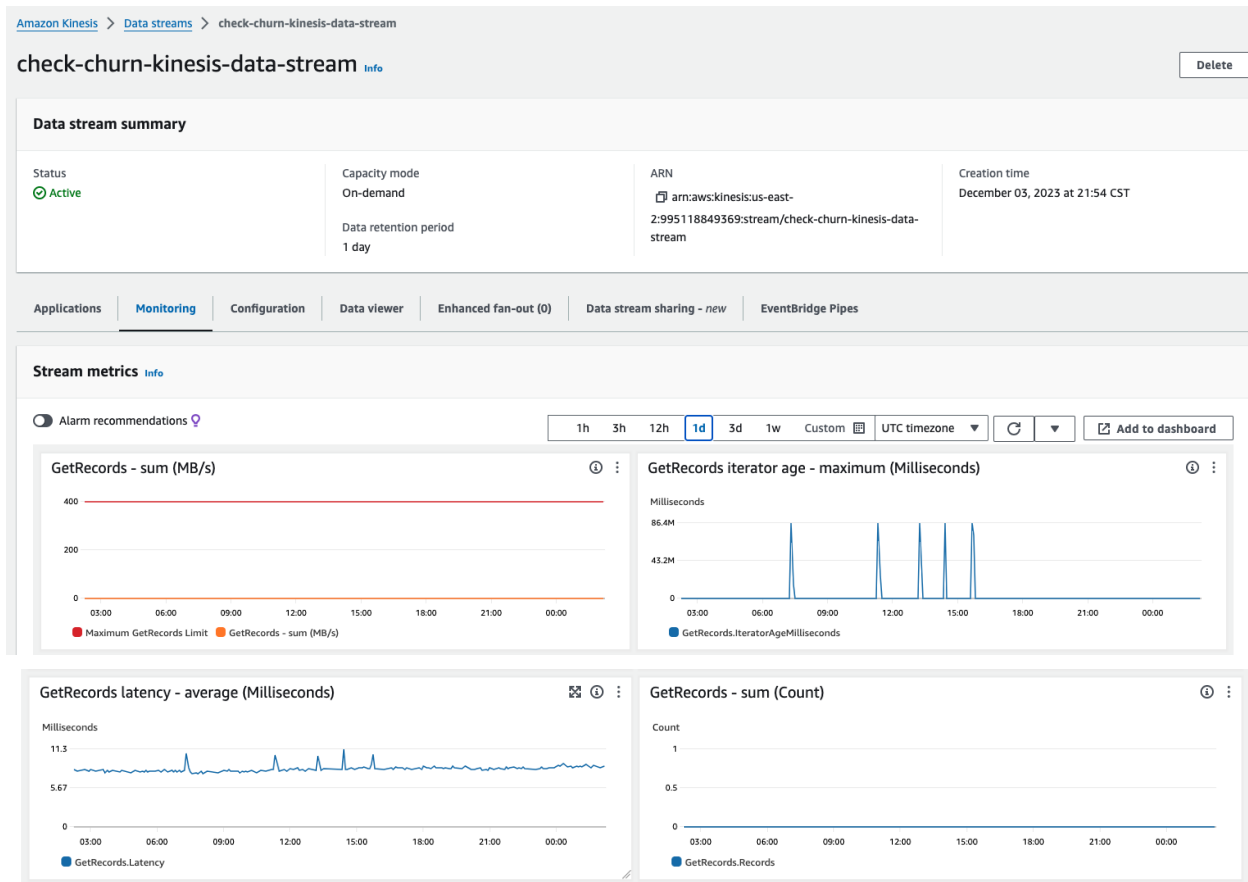
  Copy ARN  Empty  Delete  Create bucket

< 1 > 

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	aws-logs-995118849369-us-east-2	US East (Ohio) us-east-2	Bucket and objects not public	December 3, 2023, 23:07:17 (UTC-06:00)
<input type="radio"/>	churn-check-s3	US East (Ohio) us-east-2	Bucket and objects not public	December 3, 2023, 20:33:16 (UTC-06:00)
<input type="radio"/>	sagemaker-studio-995118849369-1qh1ewg78ani	US East (Ohio) us-east-2	Bucket and objects not public	December 3, 2023, 19:42:33 (UTC-06:00)
<input type="radio"/>	sagemaker-studio-995118849369-bixwzkrbcl	US East (Ohio) us-east-2	Bucket and objects not public	December 3, 2023, 19:42:32 (UTC-06:00)
<input type="radio"/>	sagemaker-us-east-2-995118849369	US East (Ohio) us-east-2	Bucket and objects not public	December 3, 2023, 19:42:35 (UTC-06:00)

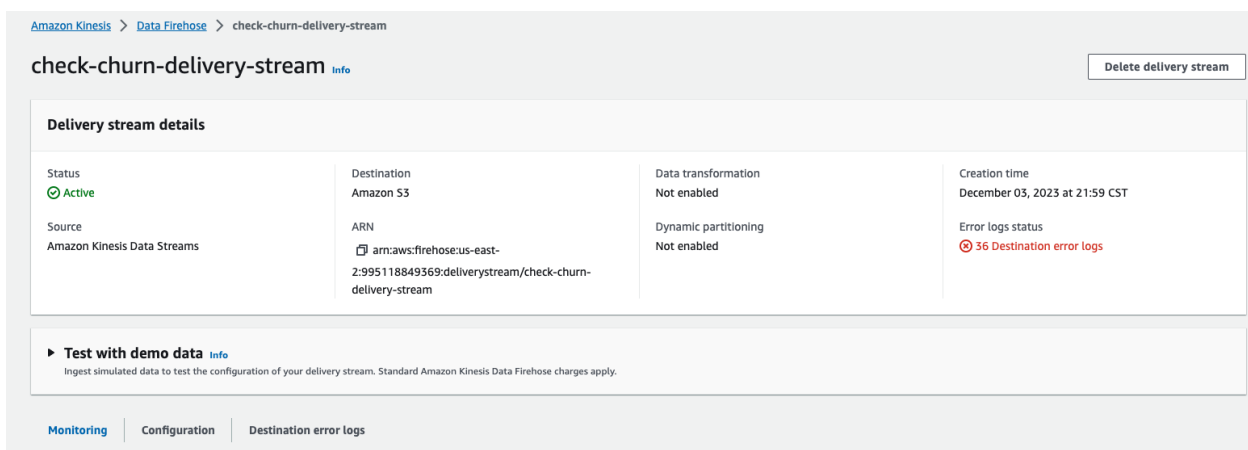
8.3 AWS Kinesis Data Streams & Firehose

8.3.1 Creating Data Stream

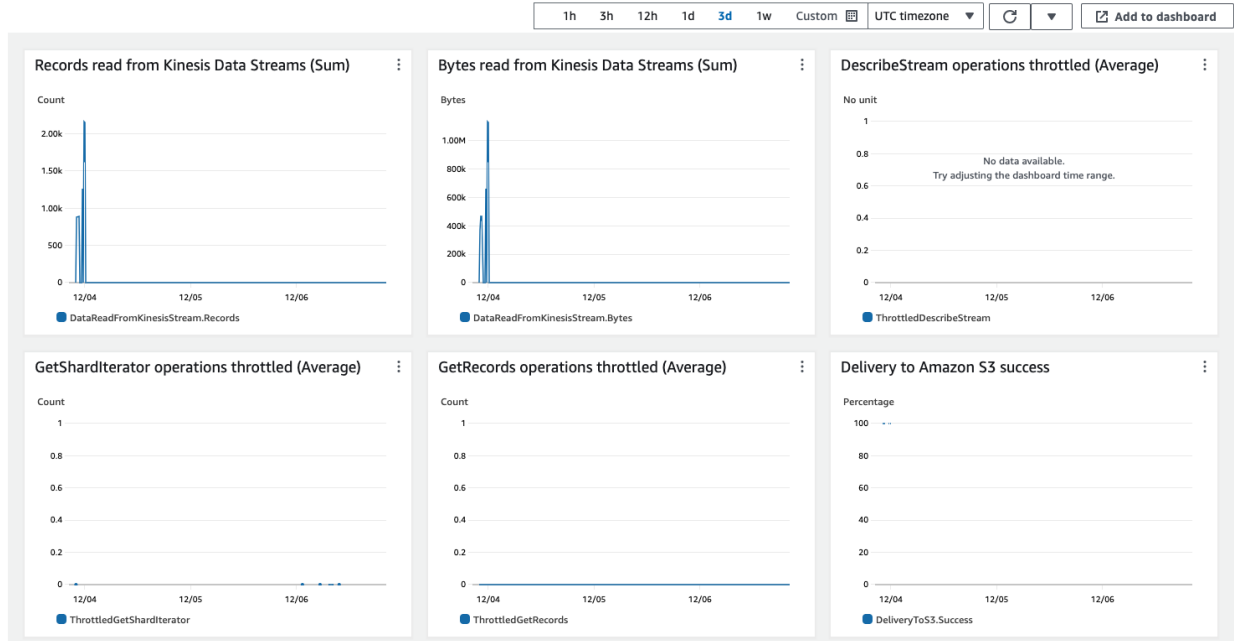


8.3.2 Sent Stream Data to AWS from local machine

8.3.3 Firehose Directs Data to S3



Delivery stream metrics [Info](#)



8.3.4 S3 stores data in chunks with date prefix

[Amazon S3](#) > [Buckets](#) > [churn-check-s3](#) > [churn-check-data2023/](#) > [12/](#) > [04/](#) > [05/](#)

05/ [Copy](#)

Objects | **Properties**

Objects (12) [Info](#)

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](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	check-churn-delivery-stream-4-2023-12-04-05-49-12-8c581ff6-94d9-40a7-8157-8e2e38e4c266	-	December 3, 2023, 23:54:13 (UTC-06:00)	280.7 KB	Standard
<input type="checkbox"/>	check-churn-delivery-stream-4-2023-12-04-05-49-12-c70d5dbf-6282-4396-a927-0318da19a5d1	-	December 3, 2023, 23:54:14 (UTC-06:00)	256.3 KB	Standard
<input type="checkbox"/>	check-churn-delivery-stream-4-2023-12-04-05-49-12-d82b524d-dce4-414b-8f7b-2bdc58467fbf	-	December 3, 2023, 23:54:14 (UTC-06:00)	284.9 KB	Standard
<input type="checkbox"/>	check-churn-delivery-stream-4-2023-12-04-05-49-12-eca525c9-f6d4-44f6-9071-7f193f022f87	-	December 3, 2023, 23:54:15 (UTC-06:00)	272.2 KB	Standard

8.4 Data Profiling

8.4.1 Data pre-processing with Spark EMR Cluster (compiled all files in chunks into a single csv file)

The image shows two screenshots from the AWS Management Console. The top screenshot displays the 'Amazon EMR > EMR on EC2: Clusters' page. It shows a table of clusters with columns for Cluster ID, Cluster name, Status, Creation time, Elapsed time, and Normalized instance hours. Two clusters are listed: 'Churn-check-Spark' and 'Churn-Check-EMR', both with a status of 'Terminated'. The bottom screenshot shows the 'Amazon S3 > Buckets > churn-check-s3 > compiled_data/' page. It displays a table of objects with columns for Name, Type, Last modified, Size, and Storage class. One object, 'compiled_data.csv', is listed with a size of 954.6 KB and a storage class of 'Standard'.

Cluster ID	Cluster name	Status	Creation time (UTC-06:00)	Elapsed time	Normalized instance hours
j-152HFIKAXD8PQ	Churn-check-Spark	Terminated User request	December 05, 2023, 19:53	4 hours, 51 minutes	80
j-28CKEFDYWSDH	Churn-Check-EMR	Terminated User request	December 03, 2023, 23:07	1 hour, 3 minutes	16

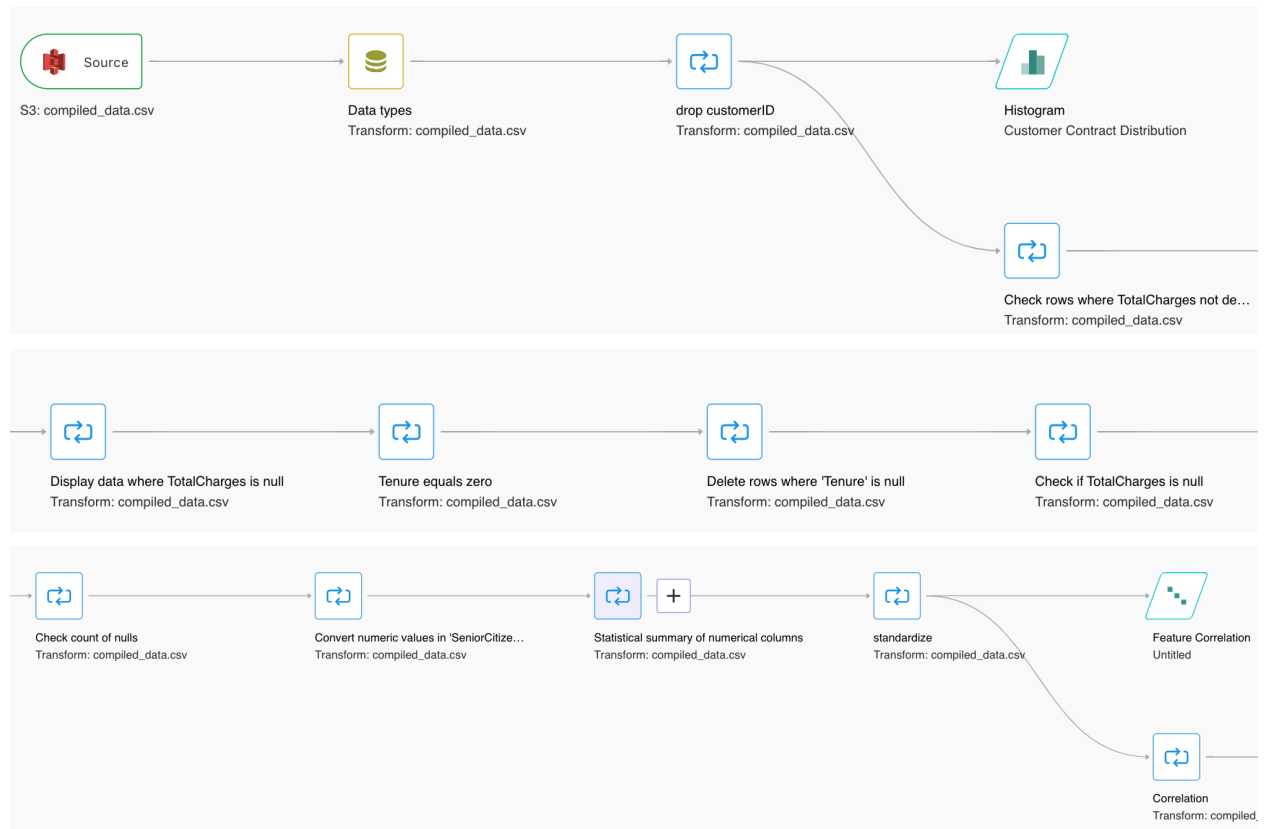
Name	Type	Last modified	Size	Storage class
compiled_data.csv	csv	December 6, 2023, 00:25:48 (UTC-06:00)	954.6 KB	Standard

8.4.2 Switched to using Data Wrangler to easily use PySpark scripts on the dataset from Data Flow

8.4.2.1 SageMaker Studio main page

The image shows the SageMaker Studio main page. The left sidebar contains navigation links for Applications (5), Home, Running instances, Data, Auto ML, Experiments, Jobs, Pipelines, Models, JumpStart, and Deployments. The main content area has a 'Home' section with a 'Launch workflows, manage your applications and spaces, and view getting started materials.' message. Below this is an 'Overview' section with a 'Start a new ML workflow or jump back into your workflow' message. The page features two large cards: 'JupyterLab' and 'Code Editor'. The 'JupyterLab' card describes it as a tool to 'Create, manage, and run durable instances of JupyterLab using spaces.' The 'Code Editor' card describes it as a tool to 'Create, manage, and run durable instances of Code Editor using spaces.' Below these cards is a 'Prebuilt and automated solutions' section with two cards: 'JumpStart' and 'AutoML'. The 'JumpStart' card describes it as a tool to 'Quickly deploy, fine-tune, and evaluate pre-trained models.' The 'AutoML' card describes it as a tool to 'Automatically build, train, and tune models.'

8.4.2.2 SageMaker Data Wrangler - Data Flow



8.4.2.3 Visualizing data and processing with PySpark scripts

Data Wrangler: Data flow > churn-check.flow > compiled_data.csv

[Data](#)[Analyses](#)

Step 5. Display data where TotalCharges is null

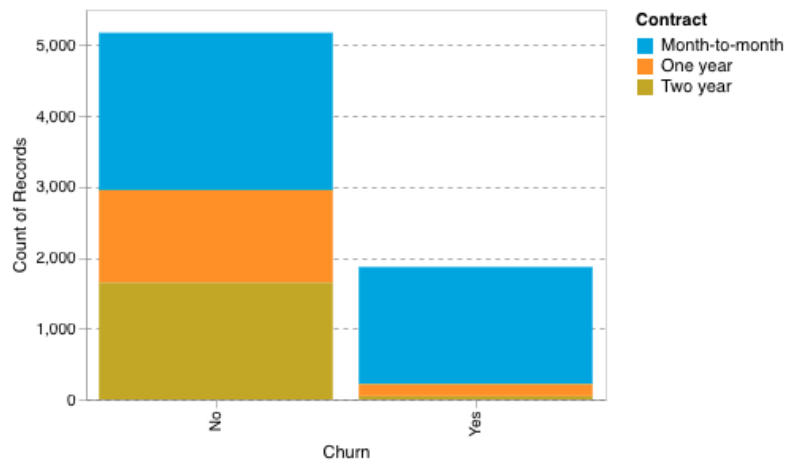
Chat for data prep

Show steps

Create model

Export data

gender (string)	SeniorCitizen (long)	Partner (string)	Dependents (string)	tenure (long)	PhoneService (string)
2 Categories	2 Categories	2 Categories	2 Categories	0 - 72 123	2 Categories
Female	0	Yes	No	1	No
Male	0	No	No	34	Yes
Female	0	No	No	8	Yes
Male	0	No	No	25	Yes
Male	0	No	Yes	71	Yes



8.4.2.4 Find rows with null values and drop them

```

1 # Table is available as variable `df`
2 from pyspark.sql.functions import col
3
4 # Filter the DataFrame for rows where 'TotalCharges' is null
5 df_filtered = df.filter(col("TotalCharges").isNull())
6
7 # Display the filtered rows
8 df_filtered.show()

```

[Clear](#)

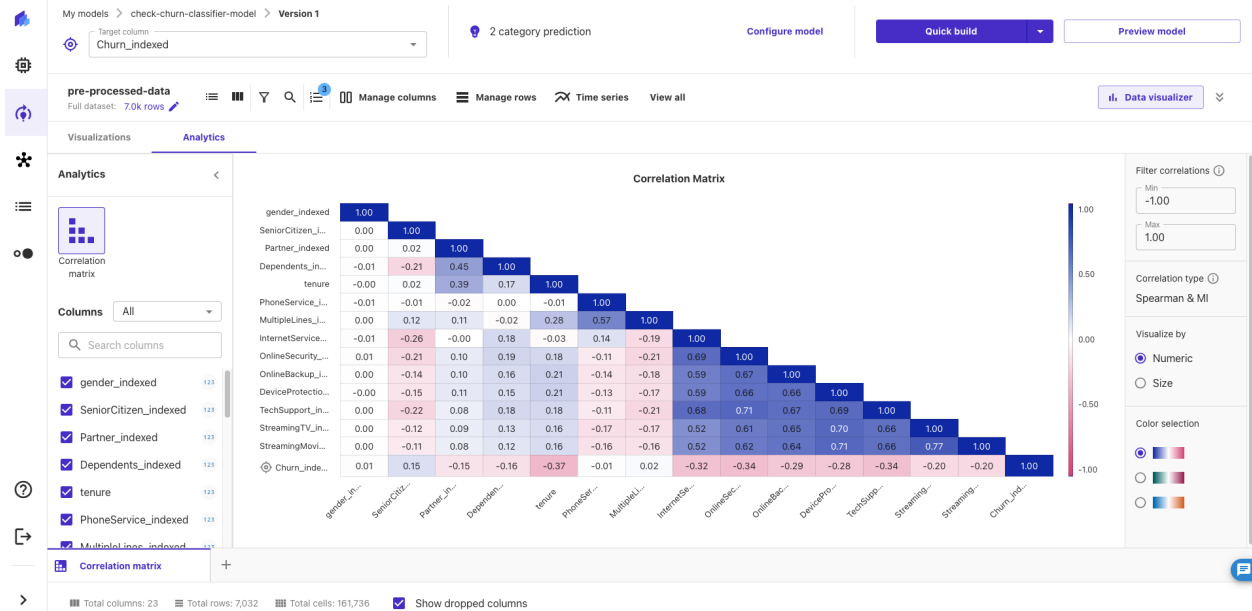
[Preview](#)

[Update](#)

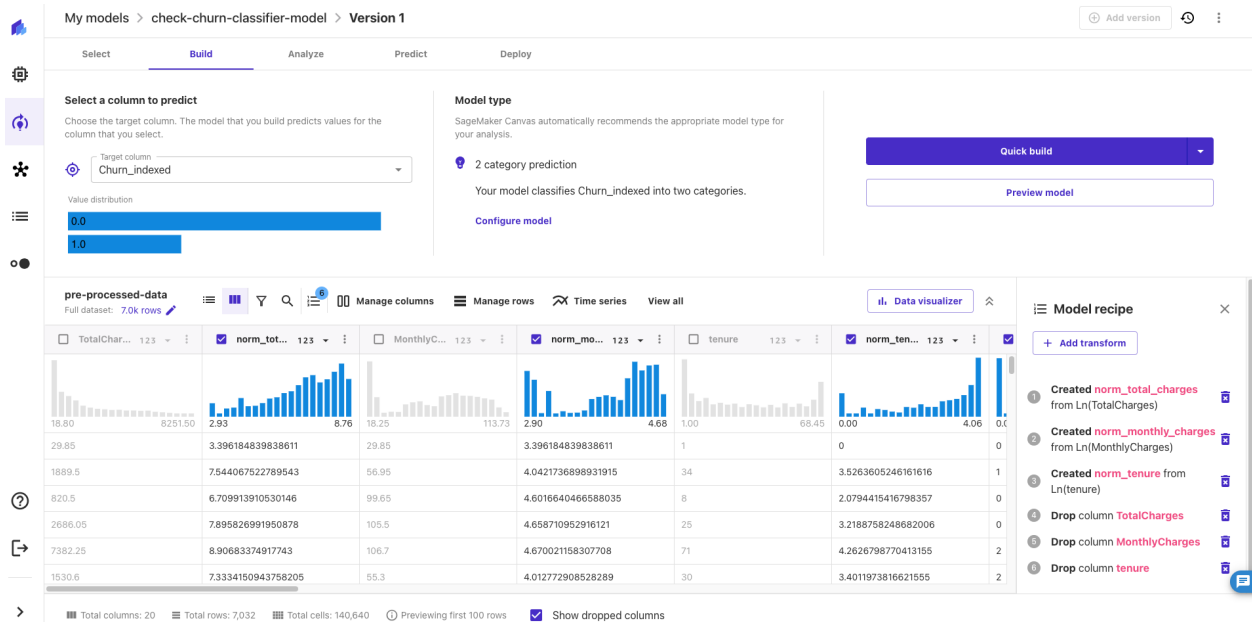
Output

	hSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
4	No	Yes	Yes	Two year	No	Mailed check	80.85	null	No
5	service	No internet service	No internet service	Two year	No	Mailed check	25.35	null	No
6	service	No internet service	No internet service	Two year	No	Mailed check	19.85	null	No
7	service	No internet service	No internet service	Two year	No	Mailed check	20.25	null	No
8	Yes	Yes	No	Two year	Yes	Bank transfer (au...	52.55	null	No
9	service	No internet service	No internet service	Two year	No	Mailed check	25.75	null	No
10	Yes	Yes	No	Two year	No	Credit card (auto...	56.05	null	No
11	service	No internet service	No internet service	Two year	No	Mailed check	20.0	null	No
12	service	No internet service	No internet service	One year	Yes	Mailed check	19.7	null	No
13	Yes	Yes	No	Two year	No	Mailed check	73.35	null	No
14	Yes	No	No	Two year	Yes	Bank transfer (au...	61.9	null	No

8.4.2.5 Correlation Matrix to find relevant columns



8.4.2.6 Normalize columns with scalar values (ex: total charges)



8.5 Building a Machine Learning model

My models > check-churn-classifier-model > Version 1

Add version

Select

Build

Analyze

Predict

Deploy

Select a column to predict

Choose the target column. The model that you build predicts values for the column that you select.

Target column

Churn_indexed

Value distribution

0.0

1.0

Model type

SageMaker Canvas automatically recommends the appropriate model type for your analysis.

2 category prediction

Your model classifies Churn_indexed into two categories.

Configure model

View Analyze tab to view your build details.

Quick build

Preview model

pre-processed-data

Full dataset: 7.0k rows

Data visualizer

Column name	Data type	Feature type	Missing	Mismatched	Unique	Mode	Correlation to target
TotalCharges	Numeric	-	0.00% (0)	0.00% (0)	6,530	20.2	-0.199
tenure	Numeric	-	0.00% (0)	0.00% (0)	72	1	-0.354
TechSupport_indexed	Numeric	-	0.00% (0)	0.00% (0)	3	0	-0.329
StreamingTV_indexed	Numeric	-	0.00% (0)	0.00% (0)	3	0	-0.205
StreamingMovies_indexed	Numeric	-	0.00% (0)	0.00% (0)	3	0	-0.207
SeniorCitizen_indexed	Numeric	Binary	0.00% (0)	0.00% (0)	2	0	0.151

Expected build time

2–15 minutes

Build type

Quick build

Detailed progress

Generating column impact

8.6 Analyzing the model

Model status

Accuracy

F1

Optimization metric

76.262%

0.642

Predict

Deploy

The model predicts the correct Churn_indexed 76.262% of the time.

Overview

Scoring

Advanced metrics

Model leaderboard

Column impact

Search columns...

1

Contract_indexed

51.131%

2

norm_monthly_charges

12.738%

3

OnlineSecurity_indexed

9.467%

4

norm_tenure

7%

5

InternetService_indexed

4.997%

6

norm_total_charges

2.173%

7

PaymentMethod_indexed

2.08%

Impact of Contract_indexed on prediction of Churn_indexed

0.0

0.44

-0.52

-1.07

-1.62

Impact on prediction

Contract_indexed