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 data[1] 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 CLI[2], 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 Streams[3]. 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 Firehose[4]. 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 Sp[6]ark 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 SageMaker[7] Data Wrangler[8] 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)*:

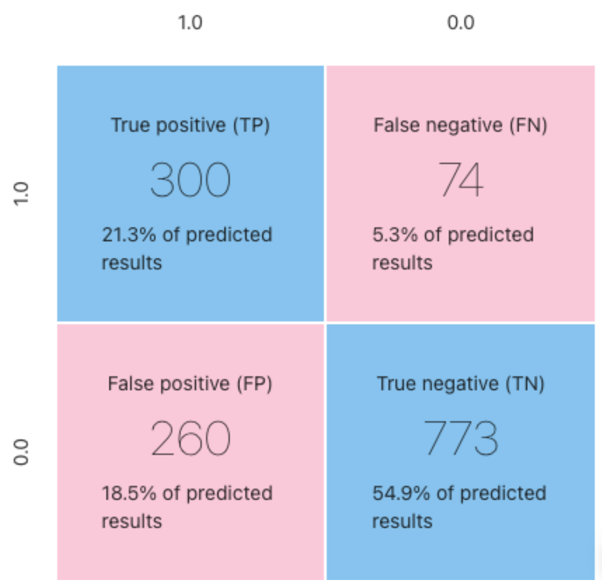


Figure X: Confusion matrix, 1: Churn-yes, 0: Churn-no

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.

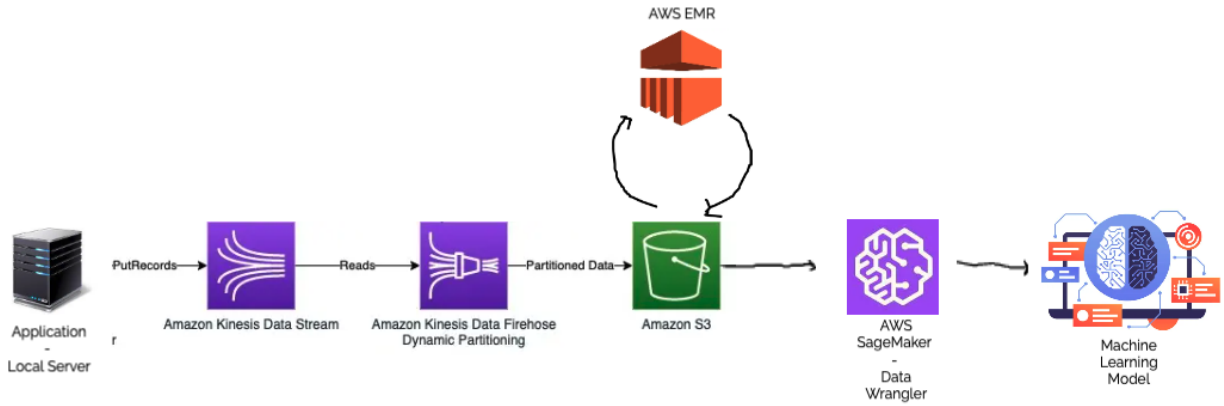
7. REFERENCES

[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

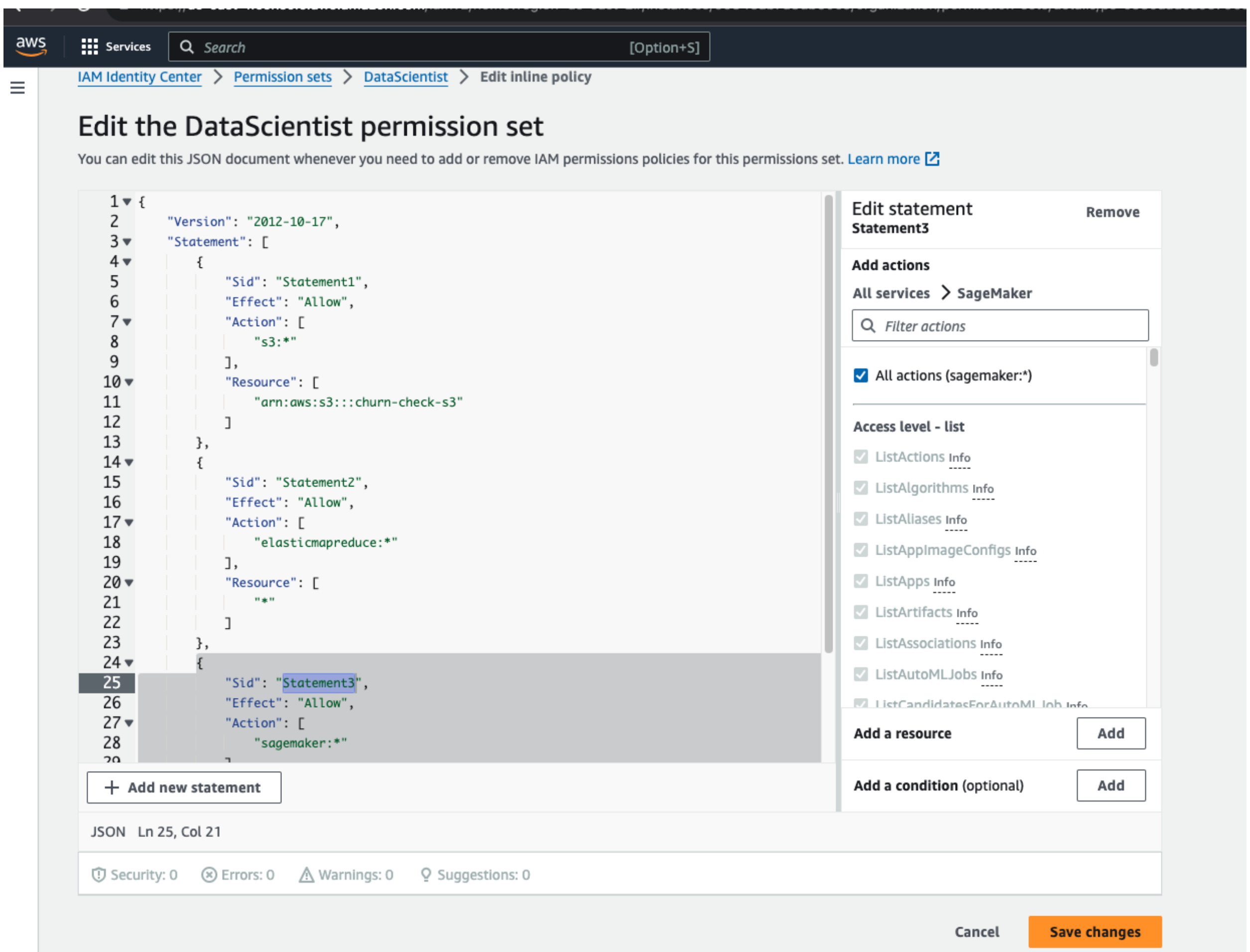
Starts from the next page.

8.0 AWS - Utilized Systems Architecture

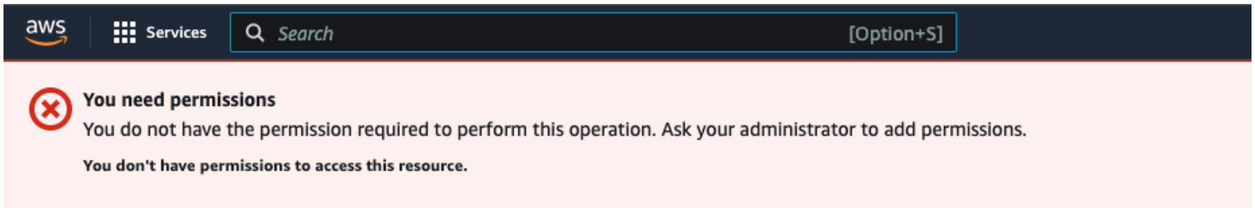


8.1 Organizational Roles

8.1.1 Adding a "Data Scientist" role to the organization

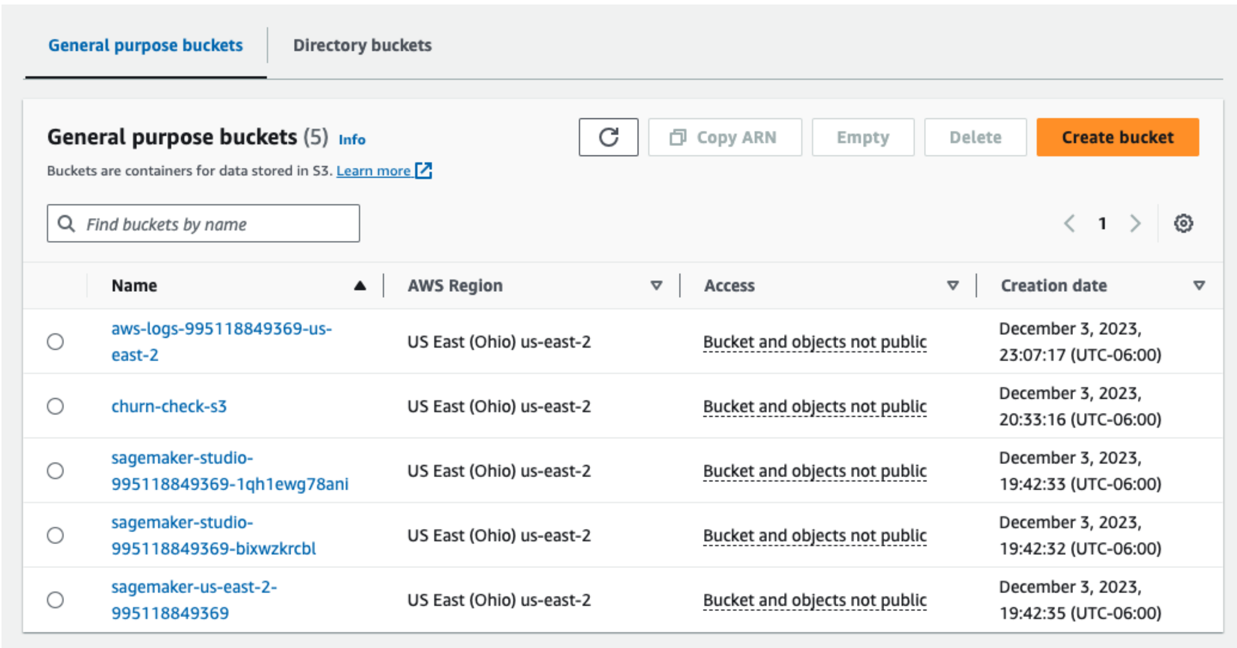


8.1.2 “Data Scientist” role cannot access restricted systems



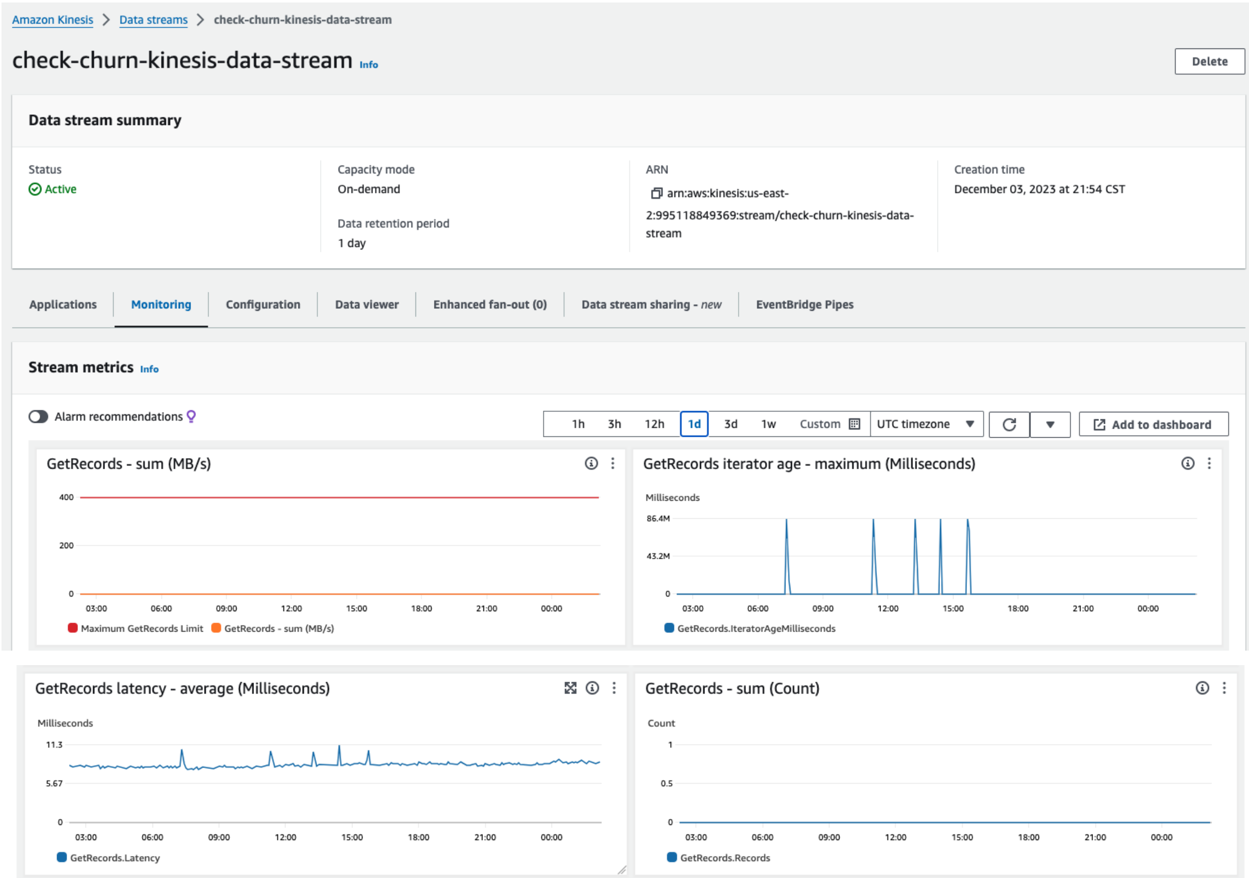
8.2 Connected to AWS through AWS CLI

8.3 Creating an S3 Bucket to store data in chunks



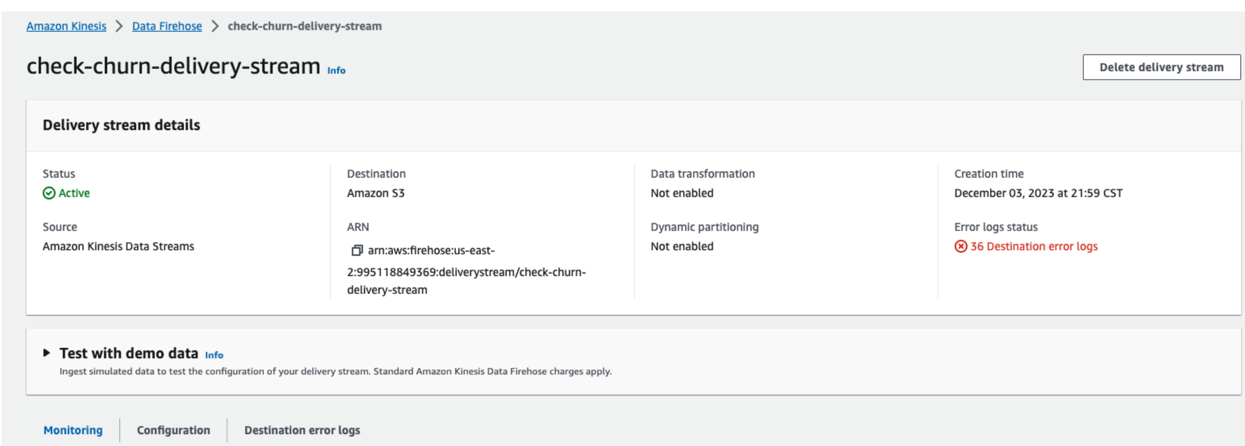
8.3 AWS Kinesis Data Streams & Firehose

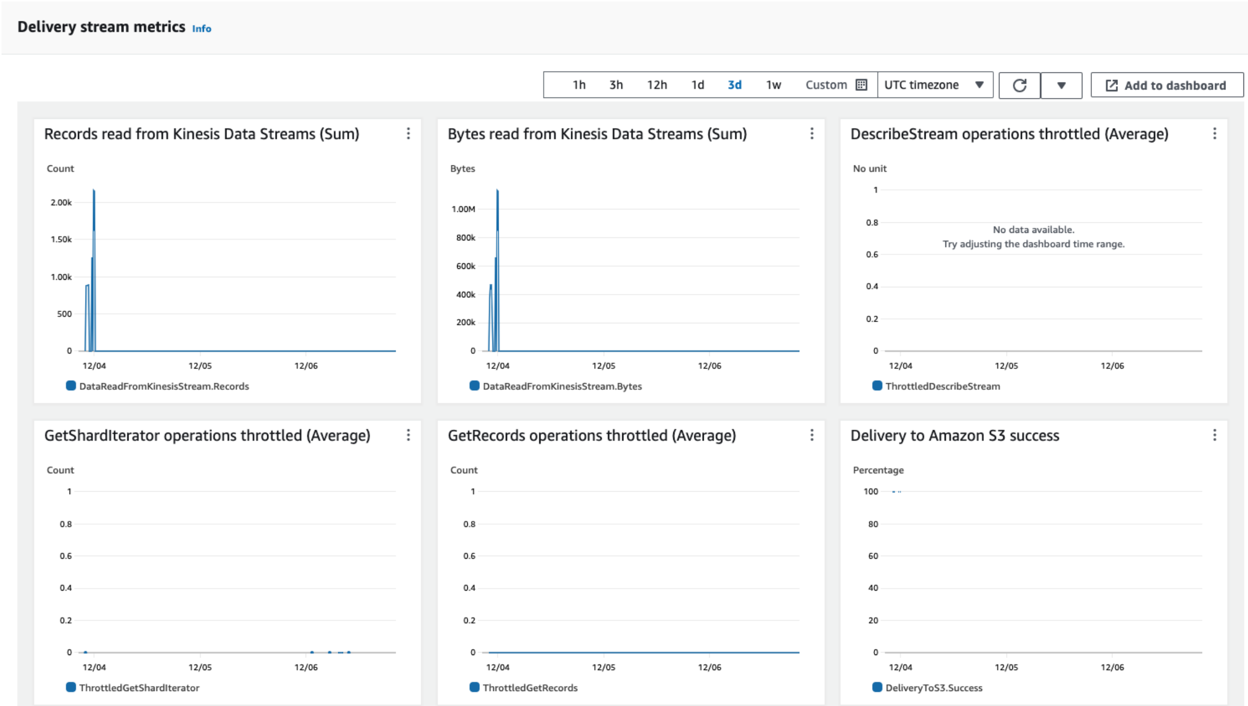
8.3.1 Creating Data Stream



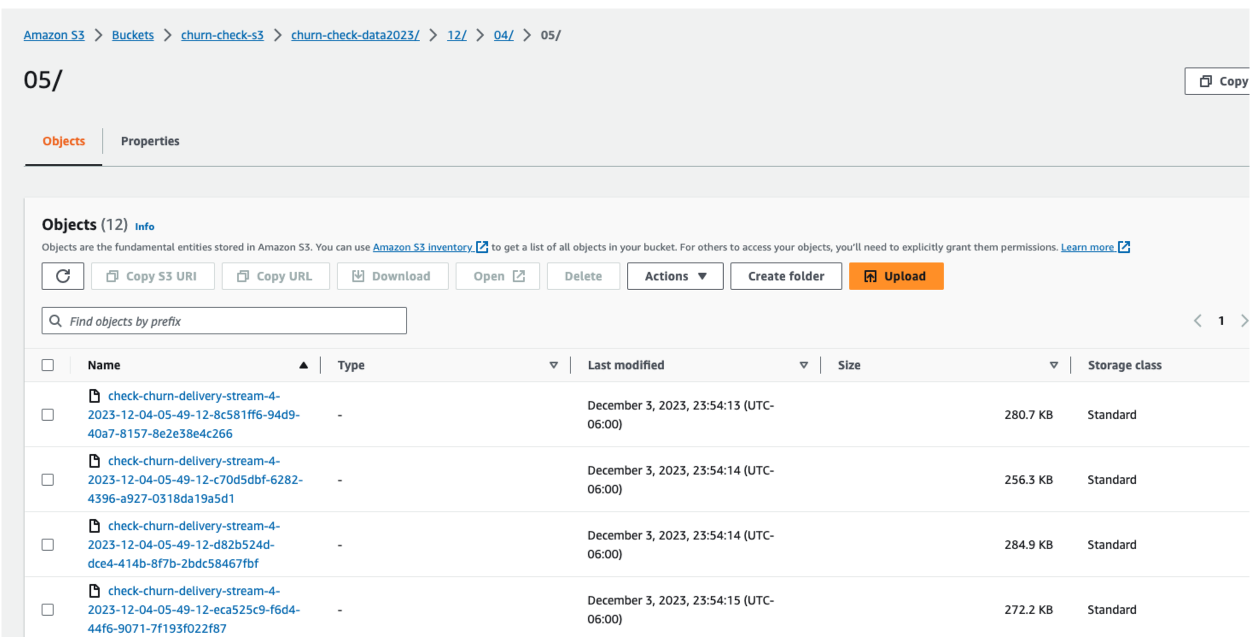
8.3.2 Sent Stream Data to AWS from local machine

8.3.3 Firehose Directs Data to S3



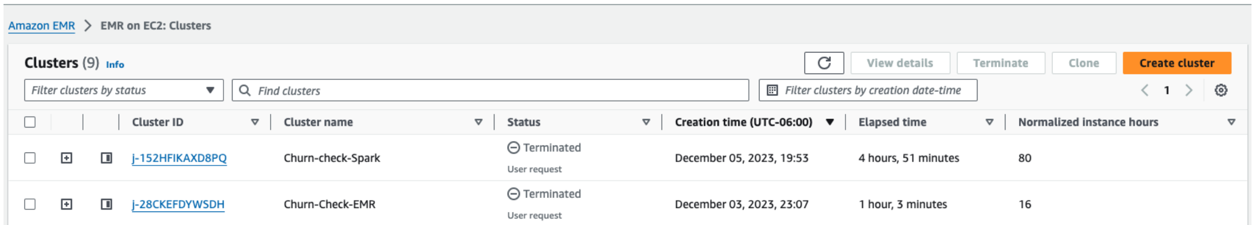


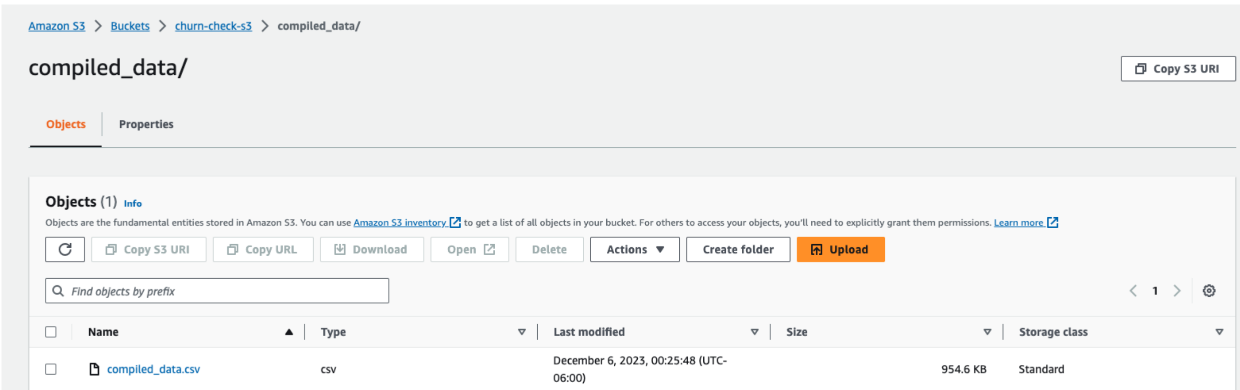
8.3.4 S3 stores data in chunks with date prefix



8.4 Data Profiling

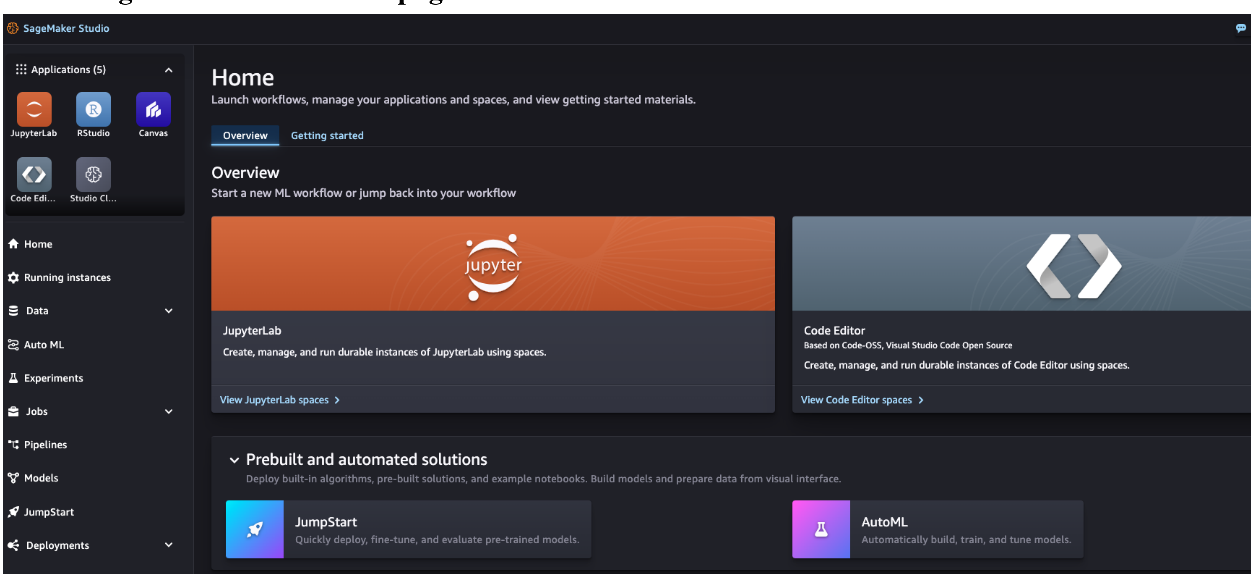
8.4.1 Data pre-processing with Spark EMR Cluster (compiled all files in chunks into a single csv file)



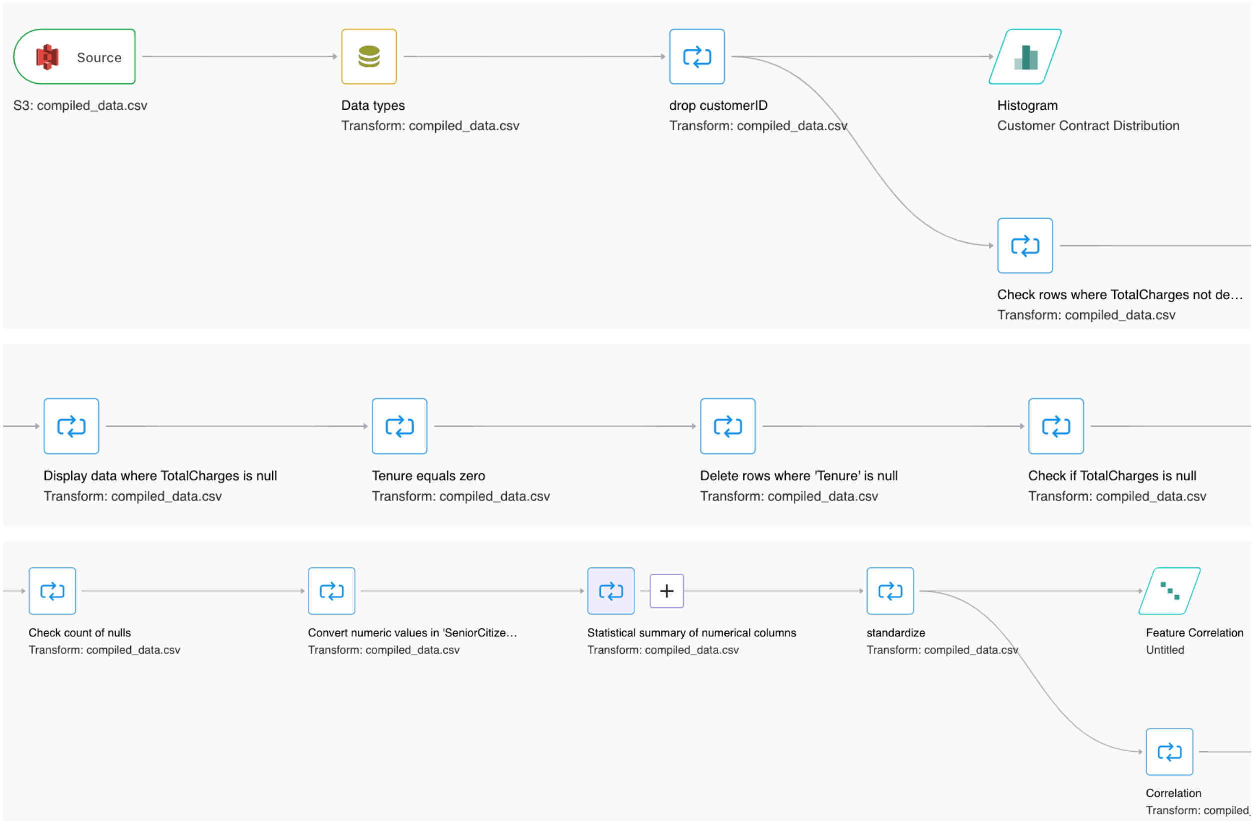


8.4.2 Switched to using Data Wrangler to easily use PySpark scripts on the dataset from Data Flow

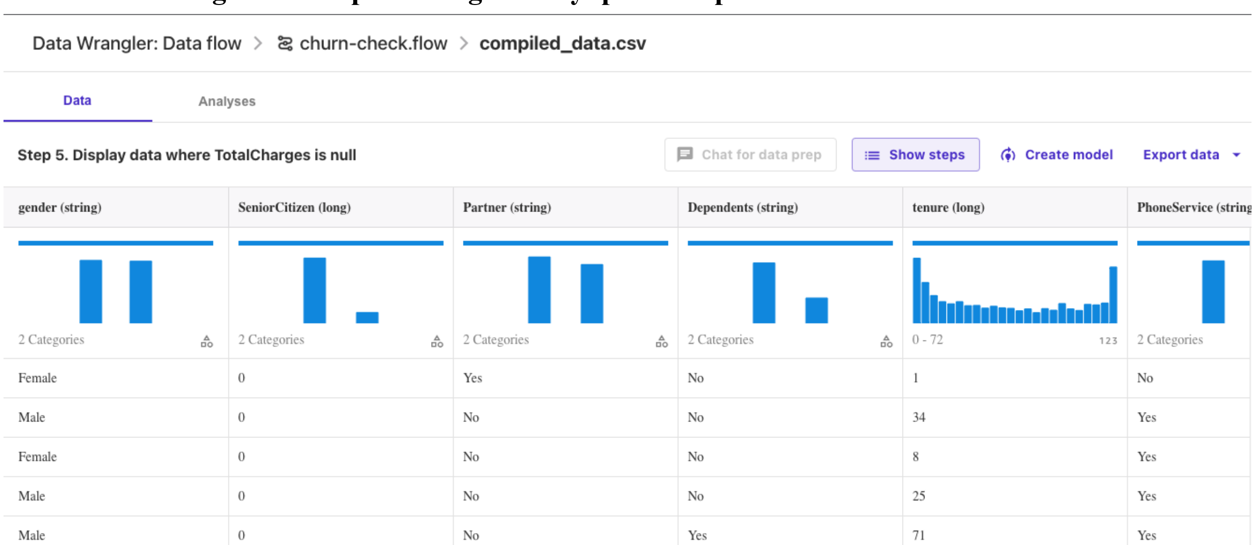
8.4.2.1 SageMaker Studio main page



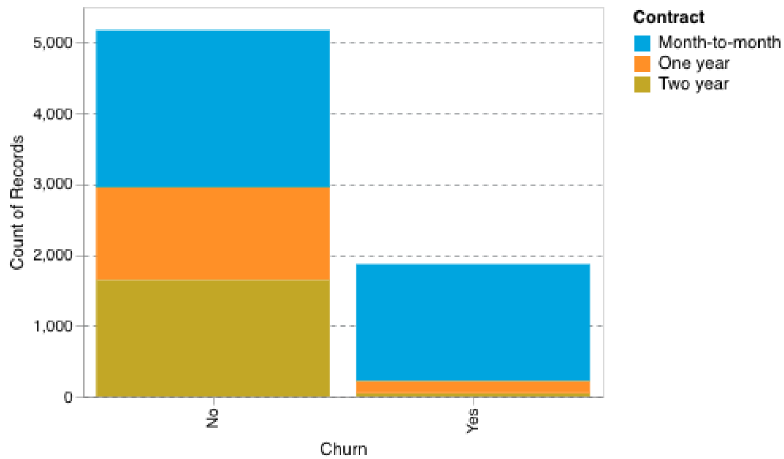
8.4.2.2 SageMaker Data Wrangler - Data Flow



8.4.2.3 Visualizing data and processing with PySpark scripts



gender (string)	SeniorCitizen (long)	Partner (string)	Dependents (string)	tenure (long)	PhoneService (string
2 Categories	2 Categories	2 Categories	2 Categories	0 - 72	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

```
# Table is available as variable `df`
from pyspark.sql.functions import col

# Filter the DataFrame for rows where 'TotalCharges' is null
df_filtered = df.filter(col("TotalCharges").isNull())

# Display the filtered rows
df_filtered.show()
```

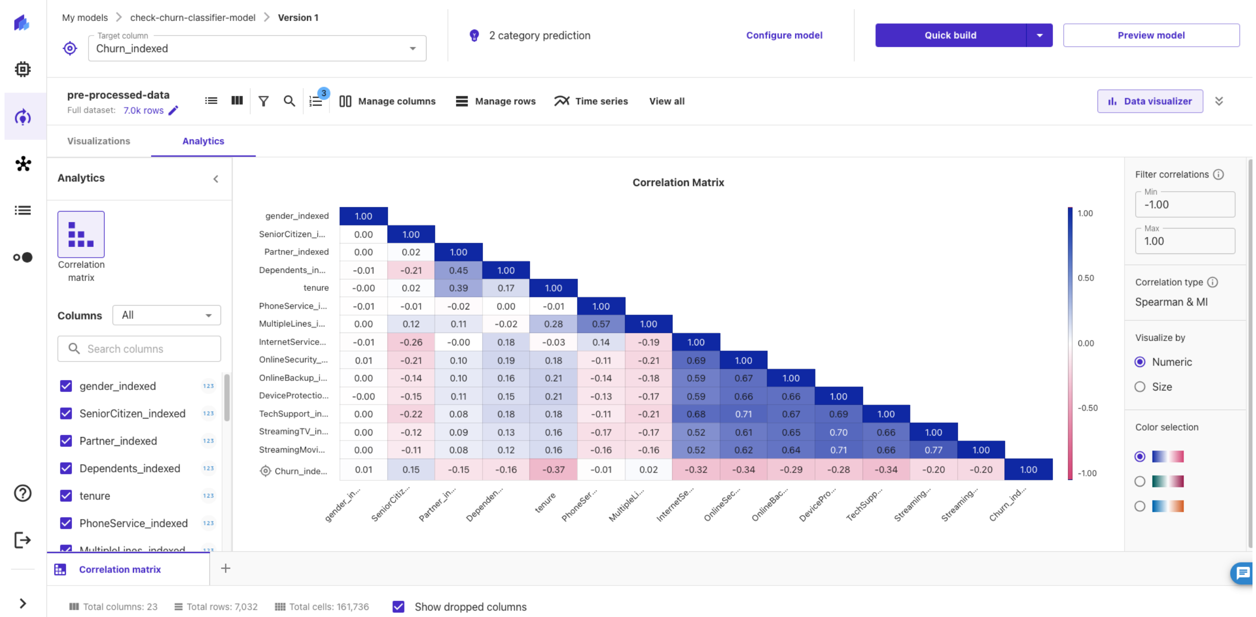
Clear

Preview Update

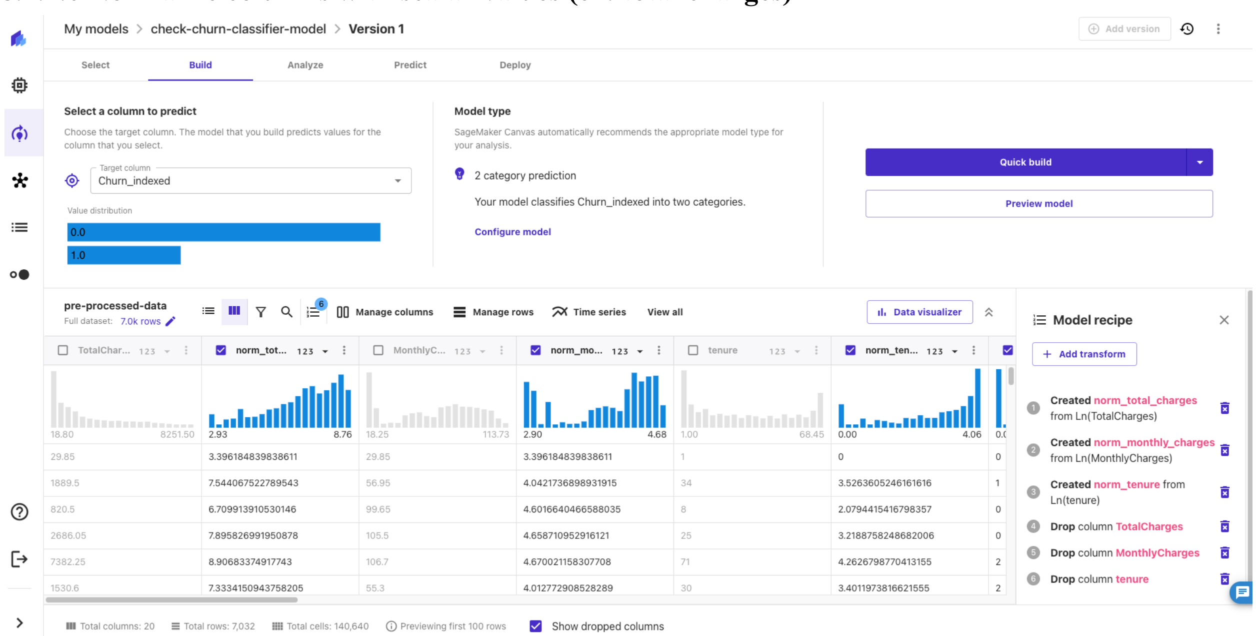
Output

```
--------+--------------------+-------------------+--------+----------------+--------------------+--------------+------------+-----+
hSupport|         StreamingTV|    StreamingMovies|Contract|PaperlessBilling|       PaymentMethod|MonthlyCharges|TotalCharges|Churn|
--------+--------------------+-------------------+--------+----------------+--------------------+--------------+------------+-----+
      No|                 Yes|                Yes|Two year|              No|        Mailed check|         80.85|        null|   No|
 service|No internet service|No internet service|Two year|              No|        Mailed check|         25.35|        null|   No|
 service|No internet service|No internet service|Two year|              No|        Mailed check|         19.85|        null|   No|
 service|No internet service|No internet service|Two year|              No|        Mailed check|         20.25|        null|   No|
     Yes|                 Yes|                 No|Two year|             Yes|Bank transfer (au...|         52.55|        null|   No|
 service|No internet service|No internet service|Two year|              No|        Mailed check|         25.75|        null|   No|
     Yes|                 Yes|                 No|Two year|              No|Credit card (auto...|         56.05|        null|   No|
 service|No internet service|No internet service|Two year|              No|        Mailed check|          20.0|        null|   No|
 service|No internet service|No internet service|One year|             Yes|        Mailed check|          19.7|        null|   No|
     Yes|                 Yes|                 No|Two year|              No|        Mailed check|         73.35|        null|   No|
     Yes|                  No|                 No|Two year|             Yes|Bank transfer (au...|          61.9|        null|   No|
--------+--------------------+-------------------+--------+----------------+--------------------+--------------+------------+-----+
```

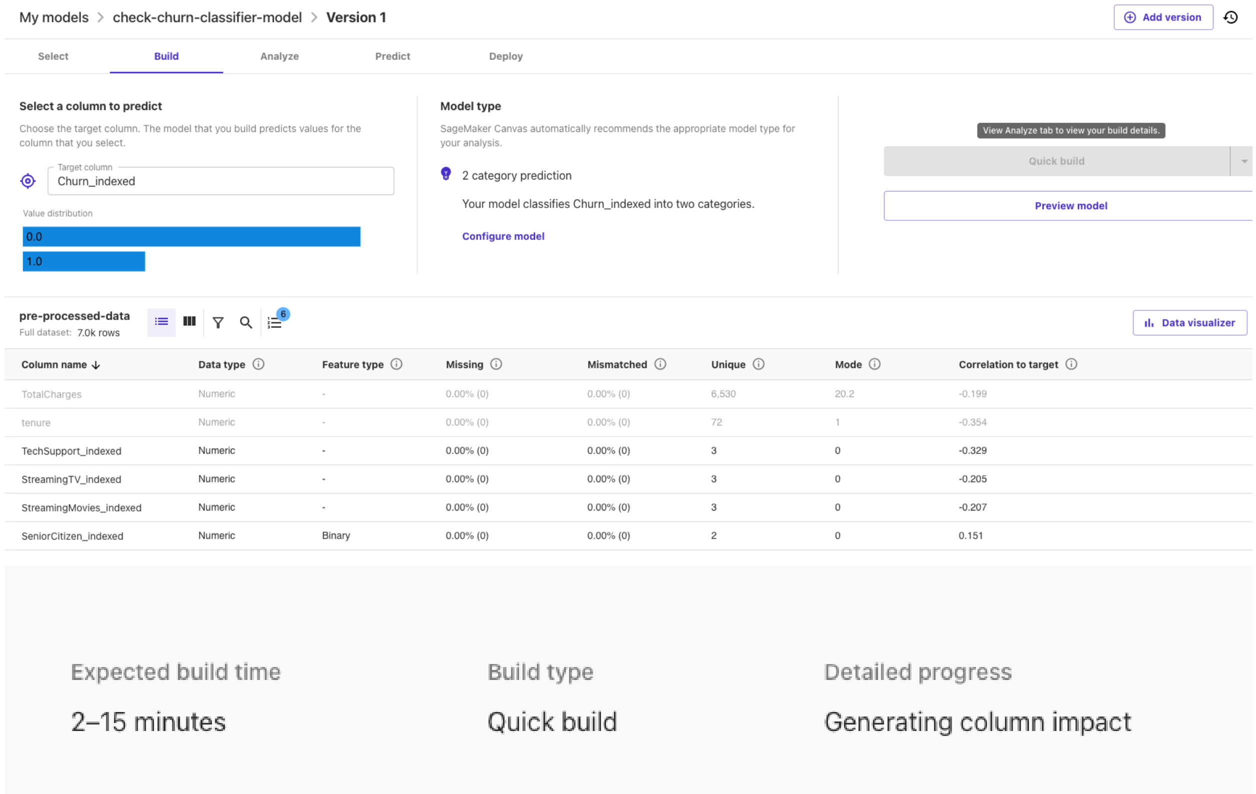
8.4.2.5 Correlation Matrix to find relevant columns



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



8.5 Building a Machine Learning model



8.6 Analyzing the model