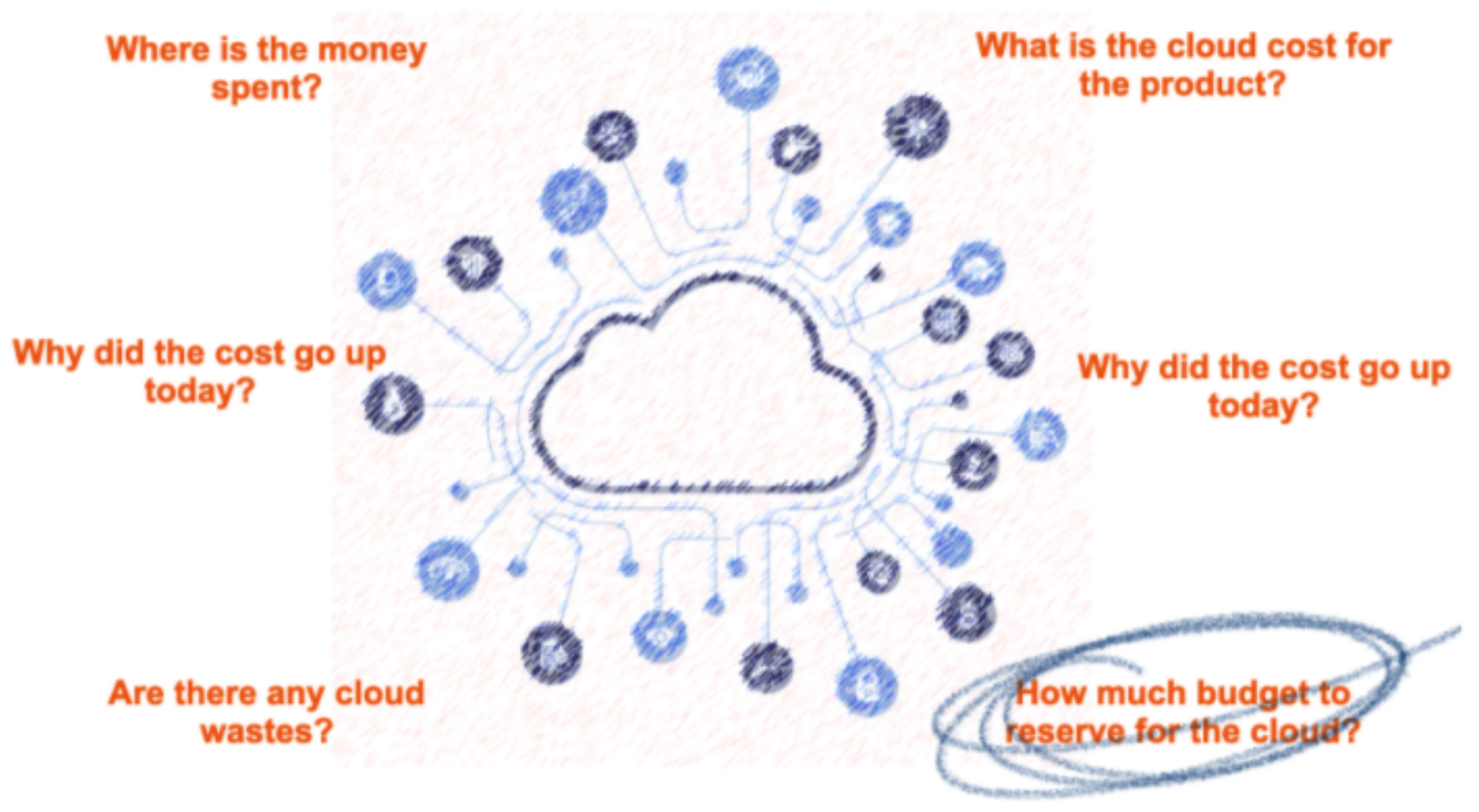
CloudZero: Cloud Cost Prediction

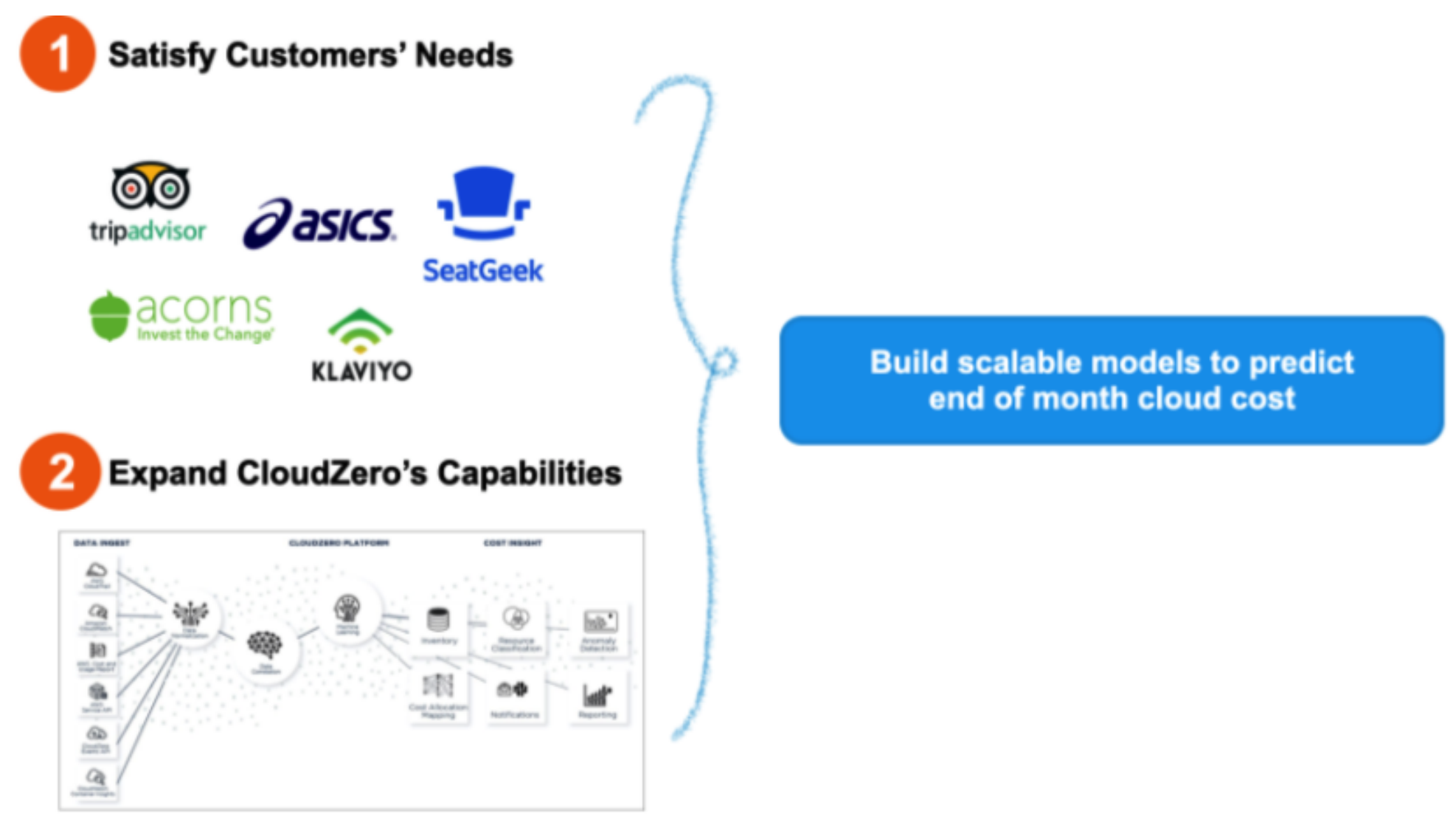
Yurui (Rui) Tong, Aarushi Bagga | Advisor: Prof. Daniel Freund

Introduction

More companies are migrating to the cloud, but they are also finding it challenging to manage and understand their cloud spend. CloudZero is a Cloud Cost Intelligence Platform that helps clients control their cloud spend and optimize cloud infrastructure.


Where is the money spent?
What is the cloud cost for the product?
Why did the cost go up today?
Why did the cost go up today?
Are there any cloud wastes?
How much budget to reserve for the cloud?

Project Motivation & Problem Statement


1 Satisfy Customers' Needs
tripadvisor
asics
SeatGeek
acorns Invest the Change
KLAVIYO
2 Expand CloudZero's Capabilities
Build scalable models to predict end of month cloud cost

Project Goals

1. **End of Month Cloud Cost Estimate for Each Paying Customer (company-level)**
   Build modeling pipelines that could be adopted by each CloudZero's paying customers over different horizons.
2. **End of Month Cloud Cost Estimate for 2 Paying Customers (product/feature-levels)**
   The outcome here is the same as Goal 1, but the prediction target is the cloud cost for a client's products and (aggregated) features.
3. **(Stretch Goal) Rolling Cloud Cost Estimates**
   Build modeling pipelines to forecast cost on a rolling basis (e.g. the next 3 days, 7 days, 14 days) instead of the end of month spend.

Bucketing Clients via Clustering

There are a variety of cloud cost dynamics observed among CloudZero's customer base. While some exhibit regular patterns, others are erratic. We clustered the companies using time-series clustering - a clustering method based on Dynamic Time Warping that groups time series with similar shapes together.

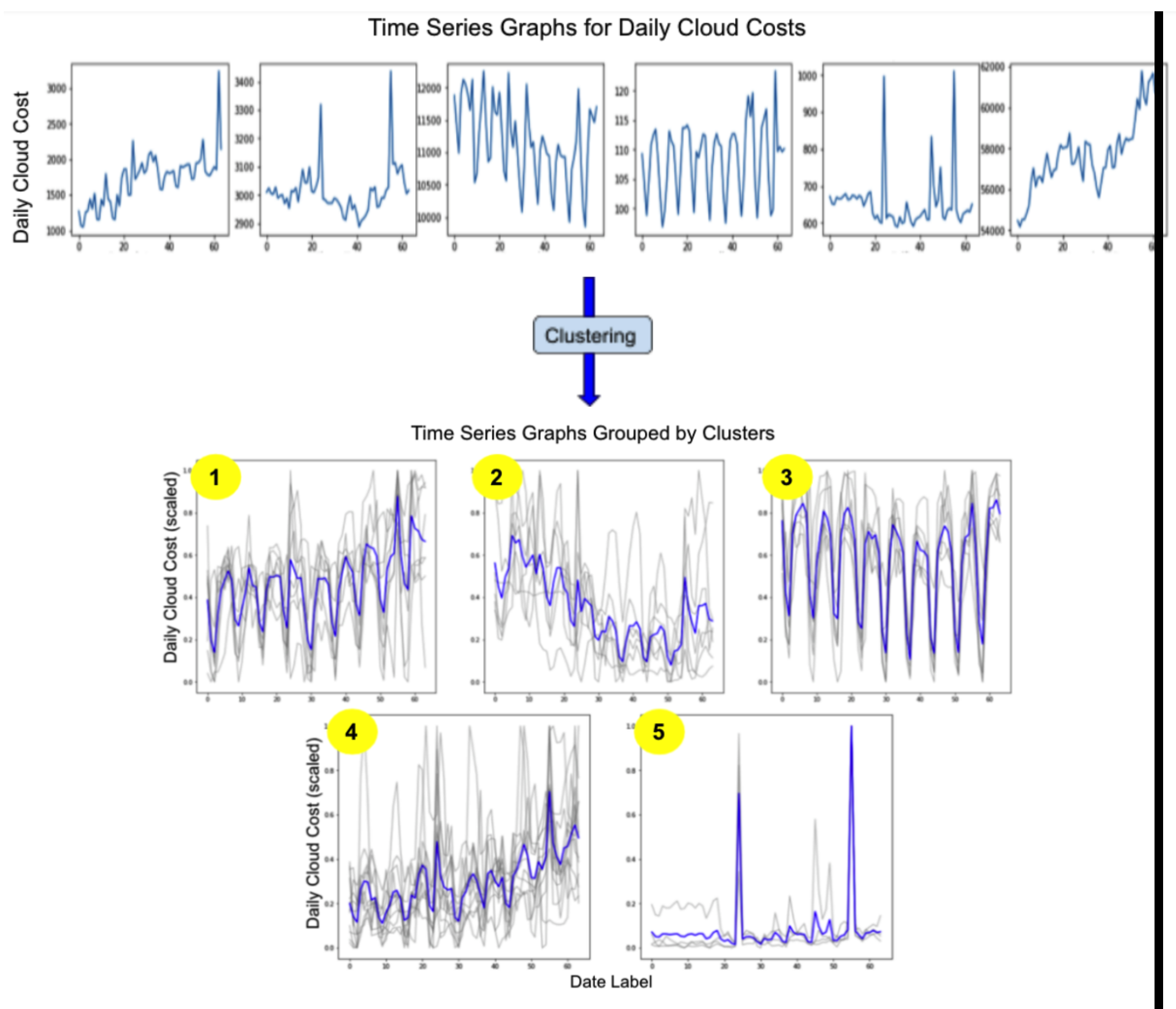
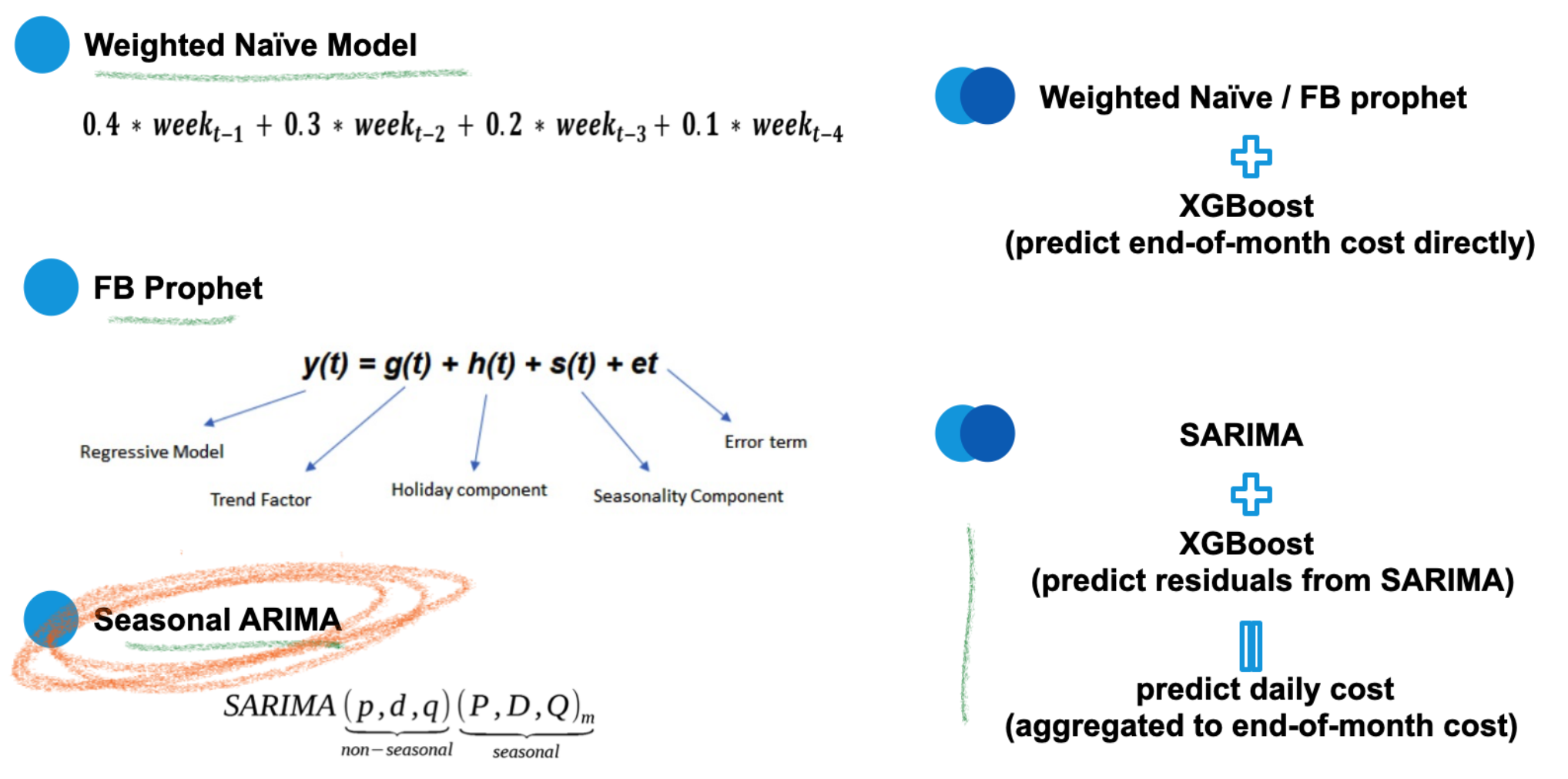


Figure 1: Various cloud cost dynamics & results after Clustering

Forecasting Methodologies

Given that we were using only billing data to build generalizable models for all customers, we had limited methods we could leverage. We tested the below mentioned univariate forecasting methods on companies from each cluster.

- Auto-regressive Methods (SARIMA/SARIMAX)
- Exponential Smoothing (TBATS)
- FB Prophet
- Naive Weighted Average
- Neural Networks (Neural Prophet and LSTM)
- Tree based models (XGBoost)


Weighted Naïve Model

$0.4 * week_{t-1} + 0.3 * week_{t-2} + 0.2 * week_{t-3} + 0.1 * week_{t-4}$

FB Prophet

$y(t) = g(t) + h(t) + s(t) + et$

Seasonal ARIMA

$SARIMA\underbrace{(p,d,q)}_{non-seasonal}\underbrace{(P,D,Q)_m}_{seasonal}$

Weighted Naïve / FB prophet
+
XGBoost
(predict end-of-month cost directly)

SARIMA
+
XGBoost
(predict residuals from SARIMA)
‖
predict daily cost
(aggregated to end-of-month cost)

Model Pipelines

To test different methodologies on multiple clients and time horizons, we built scalable model pipelines. This helped us experiment quickly and also fuse different methodologies together efficiently.

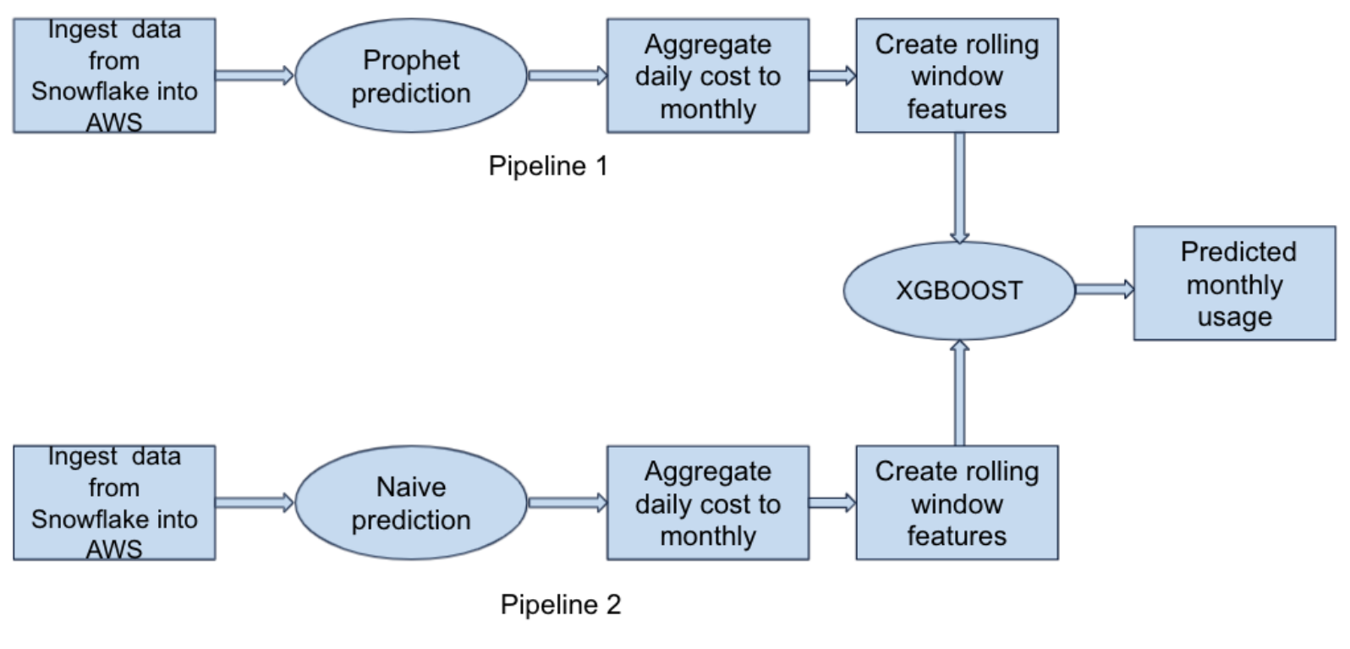

Ingest data from Snowflake into AWS
Prophet prediction
Aggregate daily cost to monthly
Create rolling window features
Pipeline 1
XGBOOST
Predicted monthly usage
Ingest data from Snowflake into AWS
Naive prediction
Aggregate daily cost to monthly
Create rolling window features
Pipeline 2

Figure 2: Using XGBoost to directly correct 30 day prediction

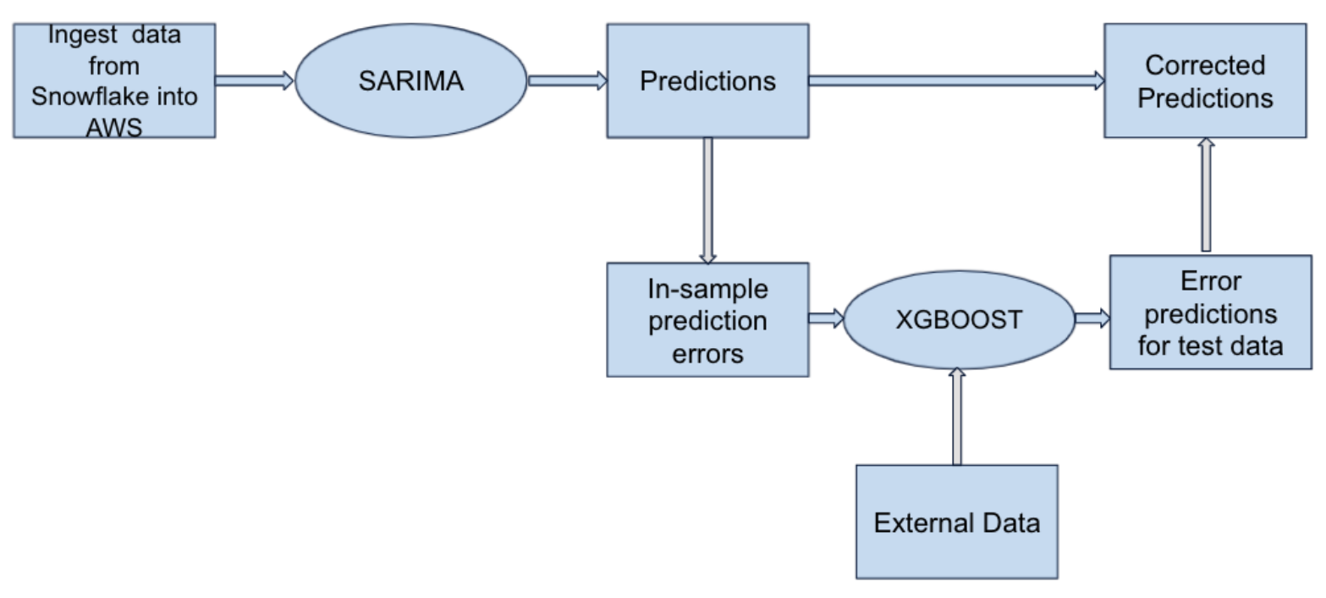

Ingest data from Snowflake into AWS
SARIMA
Predictions
Corrected Predictions
In-sample prediction errors
XGBOOST
Error predictions for test data
External Data

Figure 3: Using XGBoost to correct daily predictions

Error Analysis

To understand which model performed well given a particular training set, we visually analyzed 50+ model predictions.

- For organizations that have a good amount of signal in the training data, the SARIMA model captures shape of the series.
- Organizations for which most of the series does not exhibit a strong pattern in the signal or shows sudden jumps/lows right before the prediction date, SARIMA model overfits/underfits.

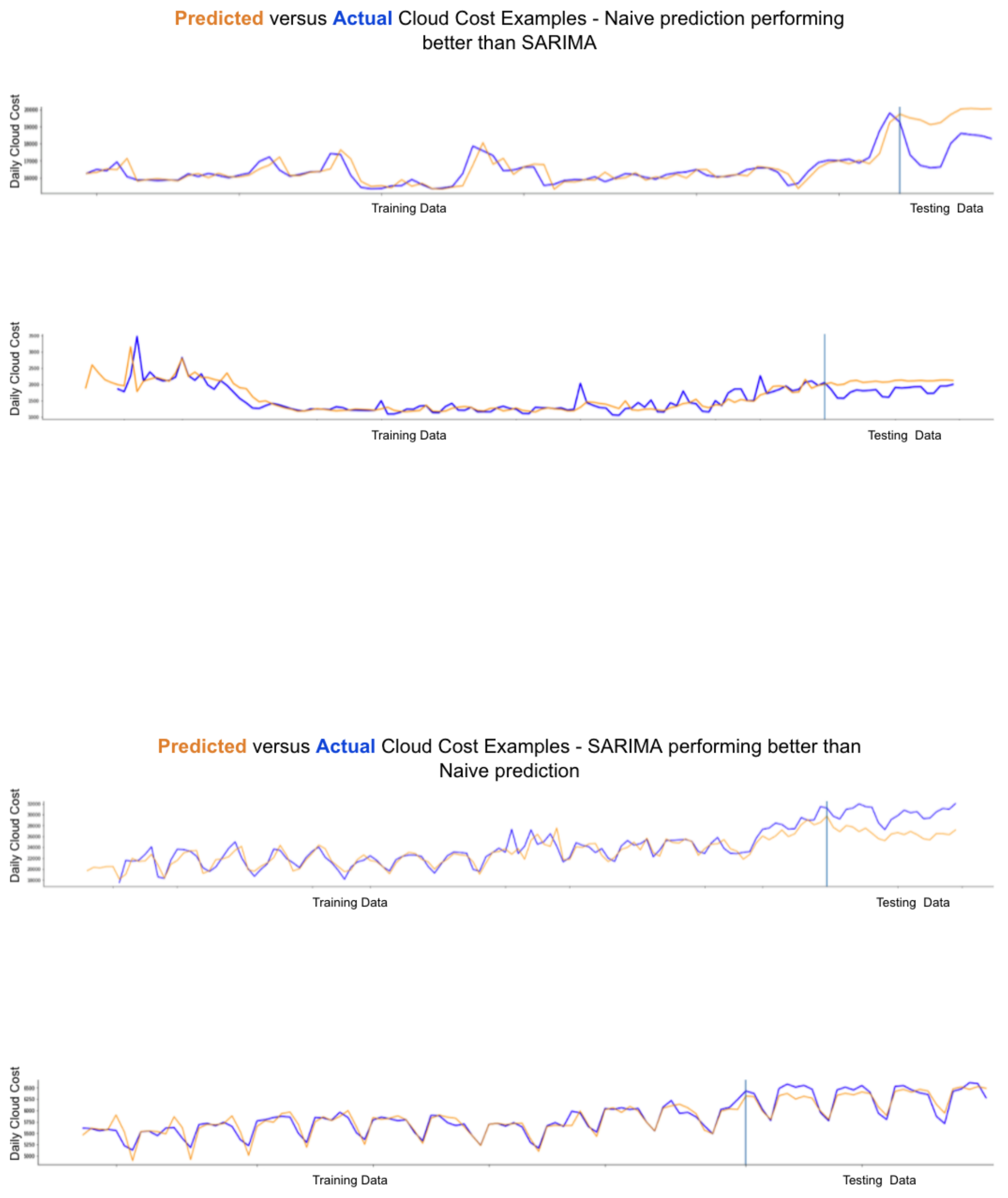


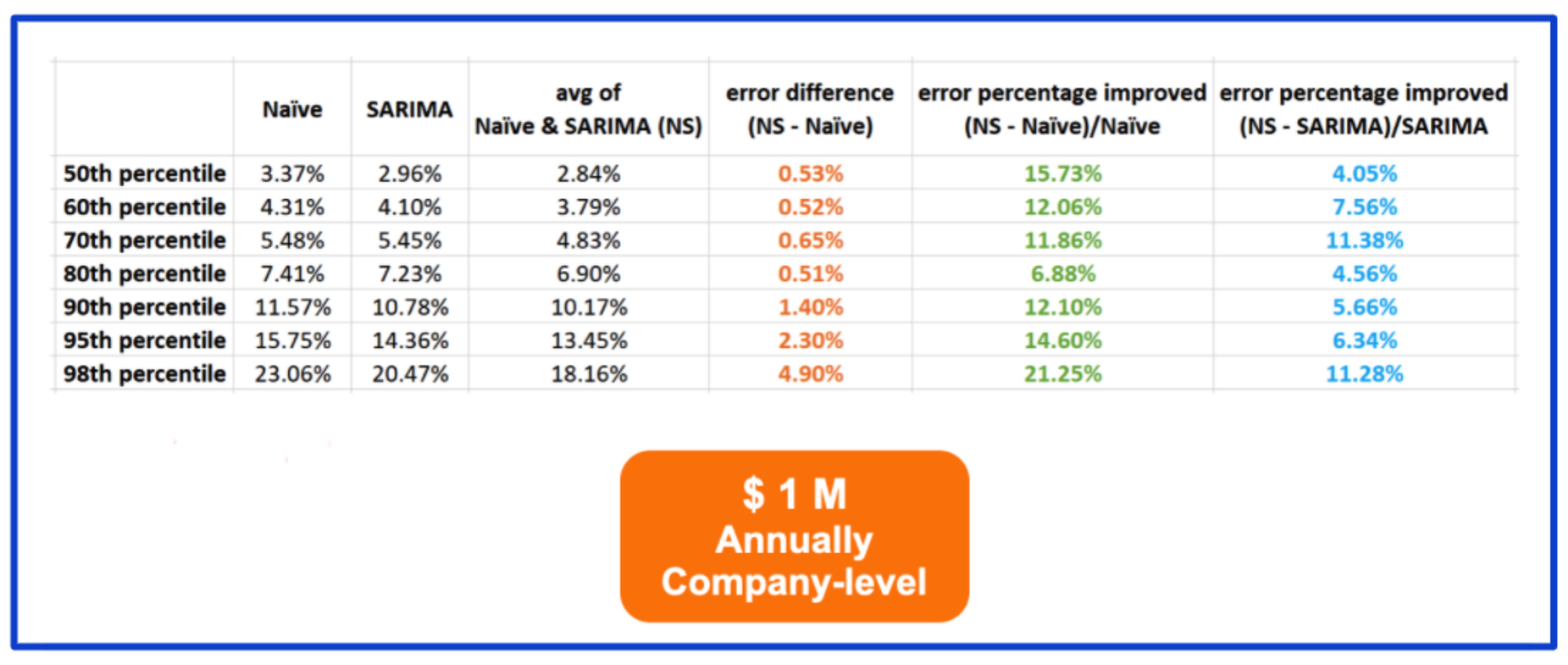
Figure 4: Deep dive into model comparisons

Recommendations

- For company-level predictions, averaging predictions from the Naive and SARIMA models gives the produces the most accurate results
- For product/feature-level predictions, using SARIMA predictions directly yielded the most accurate results
- Incorporating additional data like deployment frequency, changes in cloud infrastructure will be inevitable for further reducing forecasting errors

Results

Our modeling strategy improved naive predictions by reducing large errors.The actual dollar amount improvement was about $1 million annually.



	Naïve	SARIMA	avg of Naïve & SARIMA (NS)	error difference (NS - Naïve)	error percentage improved (NS - Naïve)/Naïve	error percentage improved (NS - SARIMA)/SARIMA
50th percentile	3.37%	2.96%	2.84%	0.53%	15.73%	4.05%
60th percentile	4.31%	4.10%	3.79%	0.52%	12.06%	7.56%
70th percentile	5.48%	5.45%	4.83%	0.65%	11.86%	11.38%
80th percentile	7.41%	7.23%	6.90%	0.51%	6.88%	4.56%
90th percentile	11.57%	10.78%	10.17%	1.40%	12.10%	5.66%
95th percentile	15.75%	14.36%	13.45%	2.30%	14.60%	6.34%
98th percentile	23.06%	20.47%	18.16%	4.90%	21.25%	11.28%

$ 1 M
Annually
Company-level