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.

**Project Motivation & Problem Statement**

**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.

**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)

**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.

**Error Analysis**

To understand which model performed well given a particular training set, we visually analyzed 30+ model predictions.

- For organizations that have a good amount of signal in the training data, the SARIMA model captures the 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.

**Recommendations**

- For company-level predictions, averaging predictions from the Naive and SARIMA models gives 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.