Stanza Data Challenge

Background

At Stanza, we track the data that comes from our StanzaCal unit. StanzaCal is a timeline of future & past events that is live on hundreds of publishers' sites. StanzaCal makes it possible for fans to discover content and events, get additional information, access relevant links, buy event tickets, and subscribe to receive live updates in their devices. We monetize StanzaCal by placing several ad units within it. Tracking and analyzing the data revolving around these ad units is incredibly important.

The two key metrics in play in advertising are **impressions** and **CPM**. An impression occurs when an ad is shown on the page. CPM (cost per mille) is the amount of revenue we receive per 1000 impressions of an ad. If an ad for the Warriors had a CPM of $1000 (completely unrealistic), we will see $1 of revenue every time we make an impression of that ad.

Dataset

You are given a CSV representing the daily revenue and impressions data (named `daily_revenue.csv`). This CSV contains the amount of revenue each site brought in every day, and how many impressions of ads they made. *You can use these two numbers to calculate the CPM.* There will be quite a number of sites. Not all sites will have information every day; assume the revenue and impressions are zero if the datapoint is not present.

Task

1. Using Python, construct a notification system that detects when the CPM for any site becomes abnormally low or high. If the CPMs drop abnormally, it is likely that something has gone wrong with the ads on that site. Knowing about this quickly can help us prevent lost ad revenue.
   - Keep in mind the fact that CPM tends to fluctuate seasonally. For example, the CPMs tend to drop at the start of each fiscal quarter, i.e. at the beginning of January, April, July, and October. Feel free to consider other time-based factors such as weekend vs. weekday, etc.
   - Train/Inform your notification system using a subset of the data, and test with another subset. For example, you can use the data prior to 2017/04/01 to train, and the data after 2017/04/01 to test.
2. Create a short presentation/writeup of what you did, what decisions you made and why, and how your model performed when you tested it. Please also send us your code and include in your write how we can use it!
3. **Extra Credit:** If you finished the tasks above very quickly, also include in your presentation/writeup an analysis of which sites are underperforming or overperforming when it comes to CPMs.

Your solution does not have to be perfect! This is purposefully an open-ended project that you can approach with many different perspectives. Keep in mind that sometimes a simpler algorithm is sufficient. If you have any trouble with the dataset or need clarification, do not hesitate to ask us. Good luck!