Adbot Ad Engagement Forecasting Challenge Documentation

Introduction

The Adbot Ad Engagement Forecasting Challenge is a project aimed at accurately predicting the number of clicks an advertisement will receive one and two weeks into the future. The goal is to provide small African businesses with actionable insights to refine their ad campaigns and bolster customer engagement. The project involves building interpretable models that can accurately forecast ad clicks while also identifying the variables that have the most significant impact on engagement.

Solution Approach

We use time series forecasting techniques to predict the number of clicks for each advertisement. The main forecasting methods used are:

Dynamic Forecast with Moving Average

Dynamic Forecast with Weighted Average

Dynamic Forecast with Weighted Average and ARIMA

Dynamic Forecast with Moving Average

This method calculates the moving average of the past window observations and uses it as the forecast for the next time step. The function dynamic_forecast_moving_average implements this approach.

Dynamic Forecast with Weighted Average

Instead of a simple moving average, this method assigns exponentially decaying weights to the past observations, giving more importance to recent data points. The function dynamic_forecast_weighted_average implements this approach, with a configurable weight_decay parameter.

Dynamic Forecast with Weighted Average and ARIMA

This method combines the weighted average approach with an ARIMA (Autoregressive Integrated Moving Average) model. The ARIMA model forecasts are used in conjunction with the weighted average of past observations to generate the final forecast. The function dynamic_forecast_weighted_average_arima implements this approach, with configurable weight_decay and ARIMA model order parameters.

The forecasting functions are applied to each group of data, identified by the ID column, using the add_dynamic_forecasts function. This function handles the time series indexing, frequency resampling, and forecast date generation.

Data Processing

We process to prepare the data for modeling and forecasting. Here are the key steps:

Extract

The data is in a tabular format. The extraction process involves reading the data into a pandas DataFrame.

Transform

The transformation steps include:

Grouping the data by the headline2_len column and applying a quantile-based cap on the clicks column to handle outliers.

Selecting the relevant columns (ID, date, clicks, cost, and impressions) and sorting the data by ID and date.

Grouping the data by date and ID, and summing the numerical columns.

Converting the date column to a datetime format.

Load

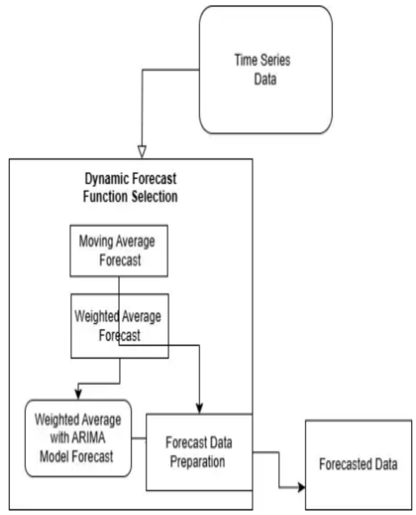
The transformed data is loaded into a pandas DataFrame, grouped by ID, and processed through the forecasting functions. The forecasted data is then concatenated with the original data, creating a combined DataFrame with actual and forecasted values.

Inference

The project includes code for generating forecasts for a specified forecast horizon (forecast_horizon) and window size (window_size). The forecast_method parameter allows selecting the desired forecasting technique ('moving_average', 'weighted_average', or 'arima_weight').

The forecasted data is stored in a pandas DataFrame, with an additional column is_forecast indicating whether the row represents actual or forecasted data.

Solution flowchat



Performance Metrics

private score on the leaderboard is 12,

Runtime

The code mentions that the runtime for generating forecasts is approximately 5 to 6 minutes.

Deployment:

- Uses containerization (Docker) and orchestration (Kubernetes, ECS)
- Containerized app with model deployed as scalable, fault-tolerant service
- CI/CD pipelines for automated testing and deployment

Inference:

- Secure data ingestion pipeline for preprocessing
- Model hosted on high-performance compute for low-latency inference
- Output stored in distributed data store/warehouse

Model Updates and Versioning:

- Versioning for model artifacts (weights, hyperparameters)
- Retrained models evaluated and deployed to staging, then production
- Seamless transition between model versions

Retraining Strategies:

- Scheduled retraining or triggered by events (performance, new data)
- Process: data collection, retraining, evaluation, deployment
- Monitoring and logging for performance tracking