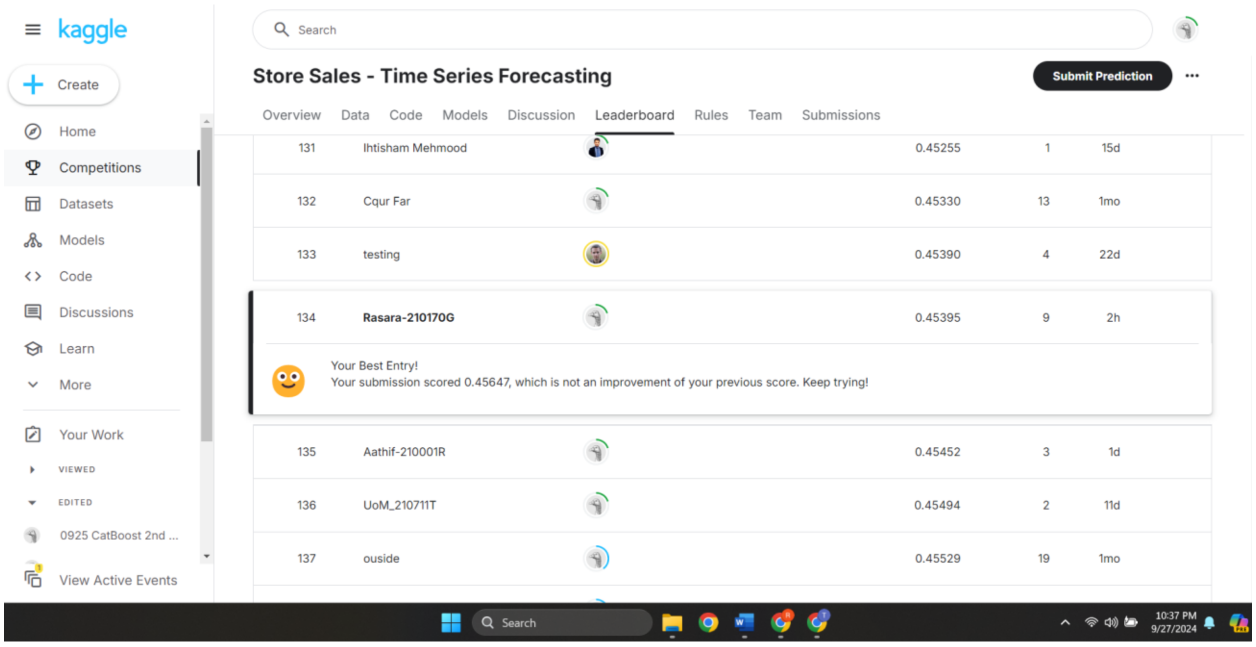
Lab 02 - Time Series Forecasting

Index : 210170G
Name : W.A.R.T. Fonseka



Store Sales - Time Series Forecasting

131	Ihtisham Mehmood	0.45255	1	15d
132	Cqur Far	0.45330	13	1mo
133	testing	0.45390	4	22d
134	**Rasara-210170G**	0.45395	9	2h
135	Aathif-210001R	0.45452	3	1d
136	UoM_210711T	0.45494	2	11d
137	ouside	0.45529	19	1mo

Your Best Entry!
Your submission scored 0.45647, which is not an improvement of your previous score. Keep trying!

Solution

In this solution, I employed the CatBoostRegressor model to forecast store sales, leveraging its ability to handle both numerical and categorical data effectively. After studying the dataset, I identified that the oil price on a specific day did not have a strong impact on sales. However, its average value over longer time periods consistently influenced sales. As a result, I calculated 7-day and 28-day moving averages for oil prices, which were used as features instead of the raw oil price.

Additionally, I decomposed date attributes into year, month, day, and weekday to capture the temporal effects of specific days on sales performance. This helped the model better understand seasonality and daily variations in sales patterns. I merged the relevant datasets and prepared them in the format required by CatBoost, using the Pool structure for efficient training and evaluation. Then I evaluated the model using various error metrics on the validation set, fine-tuning the hyperparameters to optimize performance.

Data cleaning and preprocessing steps

Steps taken:

1. Date Parsing: Converted date columns to datetime objects in all datasets.
2. Feature Engineering:
    a. Extracted useful temporal features from the date columns, such as:
       Year
       Month
       Day
       Weekday
    b. Created new features: 7-day moving average and 28-day moving average for oil price data to capture short- and long-term price trends.
3. Merging Datasets: Merged additional information from related files, including oil prices, holiday events, and store information, to enrich the dataset.
4. Handling Categorical Variables: Used Onehot encoding to encode categorical variables like store types and holiday event types.
5. Missing Data: For the merged oil dataset, missing values were filled using rolling mean values. This smooths out potential gaps without introducing bias.
6. Log Transformation: Applied log transformation on the target variable sales to stabilize variance and make the distribution more normal.
7. Removing Unnecessary Features: Dropped irrelevant features that do not contribute to the model's performance.
8. Creating Pool Structure: Prepared the dataset in a Pool structure suitable for CatBoost.

Additional Evaluation Metrices

Additionally, along with the Mean Root Logarithmic Error (used in the competition), I used the following error metrics to analyze the validation error and training performance:

- **Mean Squared Error (MSE):** I used MSE as the CatBoost training error metric because it imposes a larger penalty on larger errors, which helps the model focus on minimizing significant prediction mistakes.

- **Mean Absolute Error (MAE):** I used MAE to evaluate the error on the validation set because it provides an intuitive measure of error in the same unit as the target variable, making it easier to interpret.

- **Mean Percentage Error (MPE):** I also used MPE to analyze the validation set error. This metric is useful since the dataset contains both very high and very low sales values. It allows us to understand the error in relative terms, adjusting for the scale of the sales.

Issues in my solution

- **Takes More Time to Train**: This solution takes more time to train because our learning rate is set to 0.1. However, we cannot increase the learning rate as it may cause the model to overshoot the optimal point, leading to instability. A better solution would be to start with a higher learning rate and decrease it over time. This allows the model to take larger steps during the initial training phase and then fine-tune itself to reach the optimal point in later iterations.

- **Didn't Focus on Individual Time Series for Specific Shops and Items**: While we can predict individual time series for specific shops and items, I found that this approach degrades overall accuracy because it causes us to lose the broader effects from other time series. Instead, we could first predict individual time series and use those predictions as features, and then make predictions on the overall time series.

Alternate Approach

As alternative approach we can use of Long Short-Term Memory (LSTM) networks. LSTMs are specifically designed to capture temporal dependencies, making them a good fit for time series data.

In terms of data preparation, the process would be similar to what we implemented for the CatBoostRegressor model. Historical sales data would need to be organized in a suitable format, ensuring it includes relevant features such as store and item-specific data, promotions, and holidays. Additionally, LSTMs require the data to be structured as sequences, so we would generate input sequences by selecting a fixed window of historical sales data along with the associated features for each time step.

The model architecture for this alternative would include one or more LSTM layers, followed by fully connected layers for regression. The LSTM layers would focus on learning the temporal patterns in the data, such as trends and seasonality, while the fully connected layers would use these patterns to predict future sales.

For training and validation, the model would be trained on historical data, with a validation set reserved to assess performance. As in my CatBoostRegressor approach, we would evaluate the model using standard forecasting metrics like RMSE or MAPE to ensure accuracy.

Once trained, the LSTM model would generate sales forecasts dynamically, adapting to changing sales patterns. Although LSTM networks have the advantage of capturing time-based dependencies, I chose not to pursue this approach in my final solution due to its higher computational complexity and longer training times, which might not be suitable for our dataset size and forecasting requirements.