Research and Development on Time Series Analysis for Sales Forecasting (Task - 1)

Name:- Ayush Manoj Bisoyi
Registration Number:- 20MIM10118

Introduction

In this project, I set out to create a time series forecasting model capable of predicting the number of units sold for each item ID based on sales data from Amazon. The task entailed delving deeply into the data, developing relevant features, choosing an appropriate model, tuning its parameters, and assessing its performance. Here is a full description of my study and development process.

Exploratory Data Analysis (EDA)

Objective: Understand the data structure, identify patterns, and detect anomalies.

1. **Time Series Plotting**: I began by plotting the sales data for individual items. Visualizing the data helped me identify trends, seasonality, and any irregular patterns. It was fascinating to see how sales fluctuated over time for different items. For instance, some items showed clear weekly patterns, while others had more irregular sales.
2. **Distribution Analysis**: Next, I examined the distribution of key variables such as units sold, ad spend, and unit price. This step was crucial to detect skewness, outliers, and the overall spread of the data. The distributions of units sold and ad spend were particularly interesting, revealing insights into sales performance and advertising strategies.

Feature Engineering

Objective: Create additional features that could help the model understand the data better.

1. **Date-related Features**: I extracted features like the day of the week, day of the month, month, and week of the year from the date. These features are essential for capturing seasonality and cyclic patterns in the data. For example, sales might spike on weekends or during specific months, and these patterns can significantly improve the model's accuracy.
2. **Handling Missing Values**: I ensured the integrity of the dataset by handling missing values. For ad spend, missing values were filled with zeros, assuming no spend when data was missing. This assumption seemed reasonable given the nature of the data.

Model Selection

Objective: Choose an appropriate model that can capture the dynamics of the sales data.

1. **AutoRegressive Integrated Moving Average (ARIMA)**: After some research, I decided to use the ARIMA model. It's a well-known statistical method for time series forecasting that combines:
    - **AutoRegression (AR)**: Uses the dependent relationship between an observation and some number of lagged observations.
    - **Integrated (I)**: Differencing of raw observations to make the time series stationary.
    - **Moving Average (MA)**: Uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.
2. The `pmdarima` library's `auto_arima` function was a lifesaver here. It automates the process of finding the best set of parameters for the ARIMA model using a stepwise search. This saved me a lot of time and effort.
3. **Exogenous Variables**: I incorporated external factors like ad spend and unit price as exogenous variables in the ARIMA model. This decision was based on the intuition that these factors can significantly influence sales.

Hyperparameter Tuning

Objective: Optimize the model parameters to improve its performance.

1. **Grid Search**: I used the `auto_arima` function's stepwise search to explore different combinations of p, d, and q parameters. The function selects the best model based on the Akaike Information Criterion (AIC), ensuring a good balance between model fit and complexity.
2. **Validation**: I split the training data into training and validation sets to tune the hyperparameters and avoid overfitting. This step was crucial to ensure that the model generalizes well to unseen data.

Evaluation

Objective: Assess the model's performance using appropriate metrics.

1. **Mean Squared Error (MSE)**: I chose MSE as the primary evaluation metric. It measures the average squared difference between the predicted and actual values, with lower values indicating better performance. MSE is sensitive to large errors, which is beneficial for our use case as we want to penalize significant deviations from the actual sales figures.
2. **Train/Test Split**: I used the train dataset to build the model and the test dataset to evaluate its performance. This approach ensures that the model is evaluated on data it hasn't seen during training, providing a realistic measure of its predictive power.

Implementation

Here's a summary of the implementation steps:

1. **Data Loading and Preprocessing**:
    - Load the training and test datasets.
    - Convert date columns to datetime format.
    - Handle missing values and filter out negative values.
2. **Feature Engineering**:
    - Create date-related features.
    - Prepare training and test data for each item.
3. **Model Training and Forecasting**:
    - Use `auto_arima` to select the best ARIMA model for each item.
    - Forecast sales for a limited period (e.g., 15 days) to ensure valid prediction intervals.
4. **Result Compilation**:
    - Compile the forecasts into a DataFrame and save to a CSV file for submission.

Conclusion

By combining EDA, feature engineering, ARIMA modeling with exogenous variables, and hyperparameter tuning, I developed a robust framework for forecasting sales on Amazon. This approach leverages both historical sales data and external factors like ad spend and unit price to improve forecast accuracy. The final model is evaluated based on MSE, ensuring that the predictions are as close to the actual values as possible. This project has been a great learning experience, and I'm confident that the model will provide valuable insights for sales forecasting.