

Rossmann Store Sales Forecasting

Problem Description :

Rossmann is a well-known drugstore chain with thousands of stores across different regions. The company aims to improve its sales and inventory management by accurately forecasting future sales. Accurate sales forecasting enables the company to optimize inventory, plan promotions effectively, and make informed business decisions.

The objective of this project is to build a sales forecasting model that predicts future sales for each Rossmann store. By leveraging historical sales data, store-specific information, and external factors like holidays, the model aims to provide reliable sales predictions for upcoming periods. This will help Rossmann allocate resources efficiently, optimize store operations, and maintain customer satisfaction by ensuring the availability of products.

Dataset Information:

The dataset for this project consists of two main CSV files:

- 1. data.csv:** This file contains historical sales data for various Rossmann stores. It includes information such as the store ID, date, sales, customers, whether the store was open or closed, whether a promotion was running, and various other store-related attributes.
- 2. store.csv:** This file provides additional information about each Rossmann store. It includes details like the store ID, store type (a, b, c, d), assortment type (a, b, c), competition distance, whether there is a competitor nearby, whether the store is under promotion (Promo2), and related details.

Background Information:

Rossmann aims to gain a competitive edge by enhancing sales forecasting accuracy. Accurate predictions help Rossmann to optimize inventory levels, ensure sufficient stock availability during high-demand periods, and reduce excess inventory costs. Furthermore, the accurate forecast enables the company to plan promotions strategically, resulting in increased footfall and higher sales.

To achieve this, the project will explore the historical sales data of various Rossmann stores, identify patterns, trends, and seasonality, and leverage this information to build a reliable sales forecasting model. Additionally, external factors such as holidays and competitor activity will be considered to account for their impact on sales.

By implementing an accurate sales forecasting model, Rossmann can make data-driven decisions, improve resource allocation, and optimize store performance to achieve higher profitability and customer satisfaction.

Possible Framework :

1. Data Loading and Exploration:

- Import necessary libraries (Pandas, NumPy, Seaborn, Matplotlib, DateTime).
- Load the sales training data and store information data from CSV files.
- Explore the shape and structure of the data to understand the number of rows, columns, and data types.
- Display the first few rows of the sales training data to get an overview of the data.

2. Data Preprocessing:

- Handle missing values in both sales training data and store information data.
- Replace missing values in certain columns with appropriate defaults (e.g., 0 for competition open since year/month).
- Fill missing values for competition distance with the mean distance of other stores.
- Drop rows where the store is closed (Open = 0) as they won't contribute to sales forecasting.

3. Exploratory Data Analysis (EDA):

- Visualize the distribution of sales and other numerical features using histograms.
- Explore the unique values of categorical features like DayOfWeek, Promo, StateHoliday, SchoolHoliday, etc.
- Observe correlations between different features and the target variable (Sales).
- Plot average sales, customers, and other metrics against time (day, month, day of week) to identify trends and seasonality.

4. Data Preparation for Time Series Forecasting:

- Convert the 'Date' column to a datetime data type in both sales training data and store information data.
- Create additional time-related features such as 'Year', 'Month', and 'Day' to capture seasonal patterns.
- Group the data by 'Month', 'Day', 'DayOfWeek', 'StoreType', etc., and compute the average sales and customer count for each group.
- Prepare a separate dataframe containing school and state holidays to be used in the forecasting model.

5. Sales Forecasting with Prophet:

- Define a function for sales prediction using Facebook Prophet time series forecasting library.

- In the function, pass a store ID, sales training data, holidays dataframe, and the number of periods to forecast.
- Subset the sales data for the given store and convert it into the required format for Prophet.
- Create a Prophet model and fit it to the sales data for the store.
- Generate future dates and make predictions using the trained model.
- Plot the forecasted sales along with the historical sales to visualize the predictions.
- Display the components of the forecast (trend, seasonality) using Prophet's built-in visualization.

6. Sales Forecasting for Multiple Stores:

- Loop through each store and use the sales prediction function to forecast sales for each store.
- Store the predictions for all stores in a suitable data structure (e.g., dictionary or dataframe).

7. Evaluation and Model Performance:

- Compare the forecasted sales with actual sales for some stores to evaluate the model performance.
- Calculate evaluation metrics such as Mean Absolute Error (MAE) to assess the accuracy of the predictions.

8. Results Visualization:

- Visualize the forecasted sales for some stores using line plots and highlight the forecasted periods.
- Display the trends and seasonality components of the forecast using line plots.
- Plot the comparison between actual sales and predicted sales for some stores to observe model performance.

9. Conclusion and Recommendations:

- Summarize the findings of the sales forecasting project.
- Provide recommendations based on the forecasted sales to optimize inventory, promotions, and resource allocation.
- Discuss any limitations of the model and suggest potential areas of improvement.

10. Future Work and Extensions:

- Discuss potential enhancements and extensions to the sales forecasting model.
- Suggest incorporating additional external factors or data sources for improved accuracy.

- Propose using other advanced forecasting models or machine learning algorithms for comparison.

11.Final Remarks:

- Conclude the project and highlight the significance of accurate sales forecasting for Rossmann.
- Mention the key takeaways from the project and how it can benefit the company in achieving its goals.

12.Code Implementation:

- Implement the code for each step in the outlined process, explaining the logic and functionality of each code block.
- Include necessary comments and explanations to make the code easily understandable.

Code Explanation :

*If this section is empty, the explanation is provided in the .ipynb file itself.

1. Data Loading and Exploration:

- The code starts by importing necessary libraries such as Pandas, NumPy, Seaborn, Matplotlib, and DateTime. These libraries are used for data manipulation, visualization, and handling datetime objects.
- Next, the code loads the sales training data from a CSV file into a Pandas DataFrame named `sales_train_df`. This DataFrame contains information about sales, store details, and other relevant attributes.
- The code then explores the shape of the DataFrame to understand the number of rows and columns in the data.
- It displays the first few rows of the sales training data using the `head()` function to get a quick overview of the data.

2. Data Preprocessing:

- In this step, the code handles missing values in the sales training data and the store information data using the `isnull()` and `fillna()` functions.
- The code replaces missing values in certain columns with appropriate defaults. For example, it sets missing values in columns like 'Promo2SinceWeek', 'Promo2SinceYear', 'PromoInterval', 'CompetitionOpenSinceYear', and 'CompetitionOpenSinceMonth' to 0, as it indicates that certain promotions or competitions were not present during those times.
- The code also fills missing values for the 'CompetitionDistance' column with the mean distance of other stores, ensuring that no essential information is lost.
- Additionally, the code removes rows where the store is closed (`Open = 0`), as these rows won't contribute to sales forecasting.

3. Exploratory Data Analysis (EDA):

- EDA involves visualizing and summarizing the data to gain insights and identify patterns or trends.
- The code uses Matplotlib and Seaborn to create histograms to visualize the distribution of sales and other numerical features in the data.
- It explores unique values in categorical features like 'DayOfWeek', 'Promo', 'StateHoliday', and 'SchoolHoliday', giving an understanding of the different categories present in the data.

- Correlation heatmaps are generated to observe the correlations between different features and the target variable 'Sales'. This helps to identify which attributes may be more influential in predicting sales.

4. Data Preparation for Time Series Forecasting:

- Time series forecasting requires data to be in a specific format, so the code prepares the data accordingly.
- The 'Date' column is converted to a datetime data type using Pandas' `to_datetime()` function.
- Additional time-related features like 'Year', 'Month', and 'Day' are created using Pandas' `DatetimeIndex()` function to capture seasonal patterns in sales.
- The code groups the data by 'Month', 'Day', 'DayOfWeek', 'StoreType', etc., and computes the average sales and customer count for each group. This will be useful for understanding trends in sales over time.

5. Sales Forecasting with Prophet:

- In this step, the code uses the Facebook Prophet library to forecast sales.
- A function is defined to handle sales prediction for a specific store. The function takes the store ID, sales training data, holidays dataframe, and the number of periods to forecast as inputs.
- Inside the function, the sales data for the given store is filtered and converted into the required format for Prophet.
- A Prophet model is created and fitted to the sales data for the store using the `fit()` function.
- The model is then used to make future predictions using the `make_future_dataframe()` function. These predictions are stored in the 'forecast' variable.
- Finally, the code plots the forecasted sales along with historical sales to visualize the predictions. It also displays the components of the forecast (trend, seasonality) using Prophet's built-in visualization functions.

6. Sales Forecasting for Multiple Stores:

- The code loops through each store and uses the sales prediction function defined earlier to forecast sales for each store.
- The forecasted sales for all stores are stored in a suitable data structure, such as a dictionary or dataframe, to be further analyzed and evaluated.

7. Evaluation and Model Performance:

- The code compares the forecasted sales with actual sales for some stores to evaluate the model's performance.
- It may calculate evaluation metrics such as Mean Absolute Error (MAE) to assess the accuracy of the predictions.

8. Results Visualization:

- The code visualizes the forecasted sales for some stores using line plots and highlights the forecasted periods.
- It also displays the trends and seasonality components of the forecast using line plots.

9. Conclusion and Recommendations:

- The code summarizes the findings of the sales forecasting project.
- It provides recommendations based on the forecasted sales to optimize inventory, promotions, and resource allocation.
- The code may also discuss any limitations of the model and suggest potential areas of improvement.

10. Future Work and Extensions:

- The code may suggest potential enhancements and extensions to the sales forecasting model.
- It could propose incorporating additional external factors or data sources for improved accuracy.
- Advanced forecasting models or machine learning algorithms may be suggested for comparison.

11. Final Remarks:

- The code concludes the project and highlights the significance of accurate sales forecasting for Rossmann.
- It may mention the key takeaways from the project and how it can benefit the company in achieving its goals.

12. Code Implementation:

- The code implements each step mentioned above and explains the logic and functionality of each code block.
- It includes necessary comments and explanations to make the code easily understandable.

This outline gives a clear idea of the workflow of the code, starting from data loading and exploration, preprocessing, and EDA, to sales forecasting using Prophet. The code

efficiently analyzes the data and provides useful insights into sales patterns, allowing the user to make informed decisions for the business.

Future Work :

Sales forecasting is a crucial aspect of retail business planning, and the Rossmann Store Sales Forecasting project provides valuable insights into predicting future sales for individual stores. To further enhance the accuracy and usefulness of the sales forecasting model, we can explore several avenues for future work. Below is a step-by-step guide on how to implement the future work:

****1. Incorporate External Factors:**

- One way to improve the sales forecasting model is to include relevant external factors that might influence sales. These factors could include economic indicators, weather data, holidays, special events, and competitor information.
- Collect and preprocess the additional data sources to ensure they are in a suitable format for analysis.

****2. Feature Engineering:**

- Engage in feature engineering to create new features from the existing data that could capture more intricate patterns in sales behavior. For instance, you could create lagged variables, rolling averages, or other time-based features to consider sales trends and seasonality.

****3. Advanced Forecasting Models:**

- Experiment with more advanced forecasting models and machine learning algorithms, such as ARIMA, SARIMA, XGBoost, or LSTM, to compare their performance with Prophet.
- Implement these models and tune their hyperparameters to optimize forecasting accuracy.

****4. Ensemble Methods:**

- Consider using ensemble methods that combine predictions from multiple models to create a more robust and accurate forecasting system.
- Explore methods like model averaging, stacking, or blending to leverage the strengths of different models.

****5. Cross-Validation and Evaluation:**

- Implement robust cross-validation techniques, such as time series cross-validation (e.g., TimeSeriesSplit), to avoid data leakage and better evaluate model performance.
- Evaluate model performance using various metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE).

****6. Hyperparameter Tuning:**

- Optimize the hyperparameters of the forecasting models using techniques like Grid Search or Random Search.
- Use cross-validation to find the best combination of hyperparameters that yields the lowest forecasting error.

****7. Automate Forecasting:**

- Develop an automated system for sales forecasting that updates predictions regularly based on new data.
- Implement a scheduling mechanism to trigger the forecasting process at regular intervals (e.g., daily or weekly).

****8. Visualizations and Dashboards:**

- Create interactive visualizations and dashboards to present the forecasted sales and trends in an easy-to-understand format.
- Use libraries like Plotly or Dash to build dynamic and interactive charts.

****9. Uncertainty Estimation:**

- Consider estimating the uncertainty of the forecasts using probabilistic forecasting methods. This will provide a range of potential outcomes rather than a single point estimate.

****10. Business Scenario Analysis:**

- Conduct scenario analysis to assess the impact of different business strategies on sales forecasting.
- For example, simulate the effects of promotional campaigns, price changes, or store openings/closures on future sales.

Step-by-Step Guide to Implement Future Work:

1. Gather and preprocess additional data sources, such as economic indicators, weather data, holidays, and competitor information.
2. Perform feature engineering to create new features that could capture more intricate patterns in sales behavior.
3. Explore and implement more advanced forecasting models and machine learning algorithms, such as ARIMA, SARIMA, XGBoost, or LSTM.
4. Experiment with ensemble methods to combine predictions from multiple models for more robust forecasting.
5. Use time series cross-validation techniques to evaluate model performance and select the best model.
6. Optimize hyperparameters of the forecasting models using techniques like Grid Search or Random Search.
7. Develop an automated system for sales forecasting that updates predictions regularly based on new data.
8. Create interactive visualizations and dashboards to present the forecasted sales and trends.
9. Estimate uncertainty in the forecasts using probabilistic forecasting methods.
10. Conduct scenario analysis to assess the impact of different business strategies on sales forecasting.

By implementing these future work steps, the Rossmann Store Sales Forecasting project can deliver more accurate and actionable insights, helping the company make informed decisions to optimize inventory management, resource allocation, and overall business strategies.

Concept Explanation :

Hey there, curious souls! Today, we're going to introduce you to a fascinating algorithm called "Prophet." Now, before you start picturing a magical being with a crystal ball, let's clarify that Prophet is a forecasting tool, not a fortune teller! But hey, it works wonders in predicting future sales for our beloved Rossmann stores.

What's Prophet's Superpower?

Prophet is a forecasting algorithm developed by the fine folks at Facebook (Yep, they are more than just a place to share cat memes!). This amazing algorithm is designed to handle time series data, like our historical sales data for Rossmann stores. It's like having a math genius on your team who can spot patterns in the data and predict the future (without needing a time machine, of course!).

Understanding Time Series Data:

Time series data is like a diary that records information over time. In our case, it's like a record of how many chocolate bars, toothpaste tubes, and rubber duckies Rossmann stores sold each day. Time series data typically has a time stamp (date) and a corresponding value (sales in our case). Picture it like a timeline of sales events!

How Prophet Works Its Magic:

- 1. Preparing the Potion (Data Preprocessing):** Before we let Prophet work its magic, we need to get our data ready. This means cleaning up any missing values, removing irrelevant columns, and making sure our data is in a format that Prophet can understand. Think of this step as cleaning up your room before the wizard arrives!
- 2. Summoning the Prophet:** Once our data is all tidy, we call upon the Prophet algorithm. "Prophet, we need your forecasting wisdom!" we shout (okay, maybe not literally, but you get the idea).
- 3. Unveiling the Future (Making Predictions):** Now, Prophet takes a good look at our time series data and spots patterns and trends. It's like having a crystal ball that shows us what the future sales might look like!
- 4. Interpreting the Prophecy (Visualization):** Prophet is not just about predicting the future; it also helps us understand the past. It creates cool graphs and charts

to show us how sales have been behaving over time. Think of it as a magical artist creating beautiful paintings.

- 5. Evaluating the Crystal Ball (Model Performance):** But hold on! We need to make sure the crystal ball is accurate. Prophet uses some nifty tricks to evaluate how well it predicted past sales. If it's doing a great job, we can trust its future predictions too!
- 6. Continuous Learning:** Like any good wizard, Prophet is always learning and getting better at its predictions. We can update it with new data regularly, so it's always up-to-date with the latest sales events.

Example of Prophet's Magic:

Imagine we have sales data for a Rossmann store. We can use Prophet to predict how many unicorn plushies the store might sell in the next 30 days. Prophet looks at the historical sales data, considers factors like weekdays, holidays, and promotions, and then confidently predicts the future sales of those adorable unicorns.

In Conclusion:

So there you have it! Prophet is like a magical sales forecasting wizard that analyzes time series data, uncovers patterns, and predicts future sales. With Prophet on our side, we can make better decisions for inventory management, sales strategies, and ensuring our customers always get their favorite products.

Just remember, as amazing as Prophet is, it's not perfect. It can't predict unexpected events like a dragon invasion causing store closures (we wish it could!). But for everything else, it's a pretty impressive tool in our data-driven magical arsenal!

Exercise Questions :

1. What is the purpose of the Rossmann Store Sales Forecasting project?

Answer: The purpose of the Rossmann Store Sales Forecasting project is to predict future sales for Rossmann stores based on historical sales data. This will help the stores in better inventory management, sales planning, and making informed business decisions.

2. Why is time series data important in this project?

Answer: Time series data is crucial in this project because it captures the sales events over time. It helps in understanding sales patterns, trends, and seasonality, which are essential for accurate sales forecasting.

3. How does the Prophet algorithm work for sales forecasting?

Answer: Prophet is a forecasting algorithm developed by Facebook. It works by fitting a time series model to the historical sales data and capturing underlying patterns, such as seasonality and trends. It then uses this model to make predictions for future sales based on the available data.

4. What are the steps involved in implementing the Rossmann Store Sales Forecasting project using Prophet?

Answer: The steps include data preprocessing, merging store information, training the Prophet model, making sales predictions, and evaluating the model's performance.

5. How can missing data in the dataset be handled before using Prophet?

Answer: Missing data can be handled by either filling the missing values with appropriate methods (e.g., zero for certain columns, mean or median for numerical columns) or by removing the rows or columns with missing data.

6. How does Prophet handle holidays and special events in sales forecasting?

Answer: Prophet allows the incorporation of holidays and special events as additional features in the forecasting model. These events can influence sales, and Prophet uses them to make more accurate predictions.

7. What are some of the metrics to evaluate the performance of the sales forecasting model?

Answer: Common evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics measure the accuracy of the model's predictions compared to the actual sales data.

8. Can Prophet be used to forecast sales for individual stores?

Answer: Yes, Prophet can be used to forecast sales for individual stores by filtering the data for a specific store before training the model.

9. How can the Prophet model be updated with new data for continuous forecasting?

Answer: The Prophet model can be updated with new data by retraining the model with the historical data along with the new observations. This way, the model stays up-to-date with the latest sales events.

10. What are the potential challenges in using Prophet for sales forecasting?

Answer: Some challenges include handling outliers and unexpected events, as Prophet may not account for such anomalies in its predictions. Additionally, the accuracy of the model depends on the quality of the historical data and the availability of relevant features for forecasting.