# Time-Series Sales Forecasting using Store, Promotion, and Competitor Data

Ashwathy Mohan Menon (am5683), Gefei Zhang (gz2315), Shangzhi Liu (sl4973), Yash Jain (ycj2103), Yuzhao Liu (yl4897)

## Introduction:

Time series analysis comprises methods for analyzing time series data to get meaningful characteristics of the data and time series forecasting uses models to predict future values based on the characteristics we observed in the analysis. It has been widely used on economic forecasting, sales forecasting and budgetary analysis.
Rossmann operates over 3,000 drug stores in 7 European countries. With this high volume of sales and wide range of store locations, it's important to have a reasonable prediction of future sales and so managers can make appropriate decisions related to supply chains. In this project, we will explore and implement various machine learning techniques, analyze factors such as holidays, seasonality and competition, and predict the future sales for Rossmann based on historical sales records.

## Data Preprocessing and Exploration

**Dataset:** (Link)

**Data Preprocessing:**

- We sort the data by dates as required by a time series problem
- We apply one hot encoding to StateHoliday and Assortment.
- We have zero sales when stores are closed and these data points don't add value to our analysis hence we drop records of stores which are closed.
- CompetitionDistance is missing only for three rows and hence we replace the missing value with median.
- When Promo2 is not applicable all its dependent variables are naturally null. Hence, we replace such values with zero.

**Data Exploration:**

- We started by exploring total sales per day of the week and total customers per week for years 2013, 2014 and 2015 and noticed that sales are mostly low on Sunday because the number of stores opened on Sunday are very less. This trend has been constant for all the three years.
- Exploring total sales and total customers on School Holiday and State Holiday revealed that during school and state holidays sales and number of customers are lower.
- Plotting sales per month and sales per week of the year revealed that there are occasional spikes in sales during certain weeks like Christmas and Thanksgiving.
- Total Sales and Customers per store type was maximum for store type a.
- We checked the distribution of assortment against sales using box-plot.

# Time-Series Sales Forecasting using Store, Promotion, and Competitor Data

- To Check the variation of sales based on Day of the week and store type we opted for a factor plot.
- Barplot on the competition distance revealed that competition we are close proximity to the stores.

## Tree Based Models

### Motivation:

Decision Trees are one of the most useful and simple supervised Machine Learning Techniques that uses a tree based model for predictions. Time Series problem has been converted to supervised learning problem first. We implement two tree based models: Random Forests and XGBoost. Random forests do smarter bagging of trees where bootstrapped samples and random subset of features are used to train each tree. XGBoost ensures monotonicity and feature interaction constraints and is one of the most popular implementations of gradient boosting.

### Model Development:

We used Tree based models namely Random Forest and XGBoost for our time series problem. We sorted the data as per the dates as this is a time series model. After sorting, we divide the data in a way that our test set contains data for the last 30 days in a sorted manner and the train set contains the rest of the data in a sorted manner. We then built random forest and xgboost models using the sklearn library and employ a grid search to find the best parameters for each of the two models

### Model Evaluation:

After obtaining the best parameters for our models, we evaluate the performance of our models using the mean absolute error between the predicted value and the actual values. Additionally, we also evaluate the model by calculating the difference between the sum of the total predicted sales for the 30 days and the sum of the total actual sales for the 30 days.

### Results:

**Random forest:**

Actual Total Sales for the next 30 days: 145423
Predicted Total Sales for the next 30 days: 143643.8
Difference between actual total sales and predicted total sales: 1779.2

# Time-Series Sales Forecasting using Store, Promotion, and Competitor Data

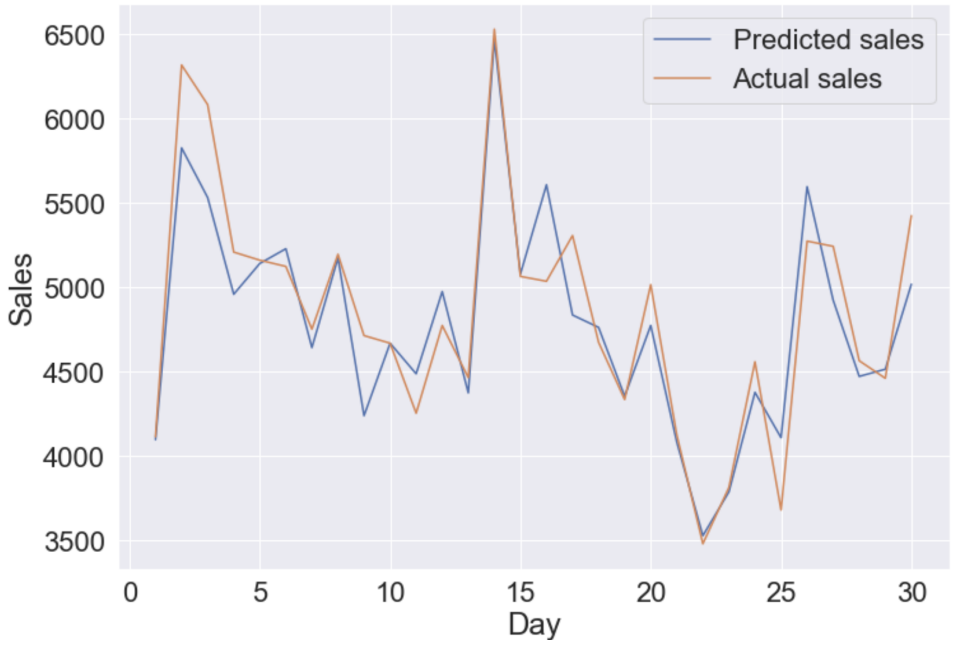


Fig 1. Predicted and Actual 30 day sales (Random Forest)

**XGBoost**

Actual Total Sales for the next 30 days: 145423
Predicted Total Sales for the next 30 days: 146194.93
Difference between actual total sales and predicted total sales: -771.92

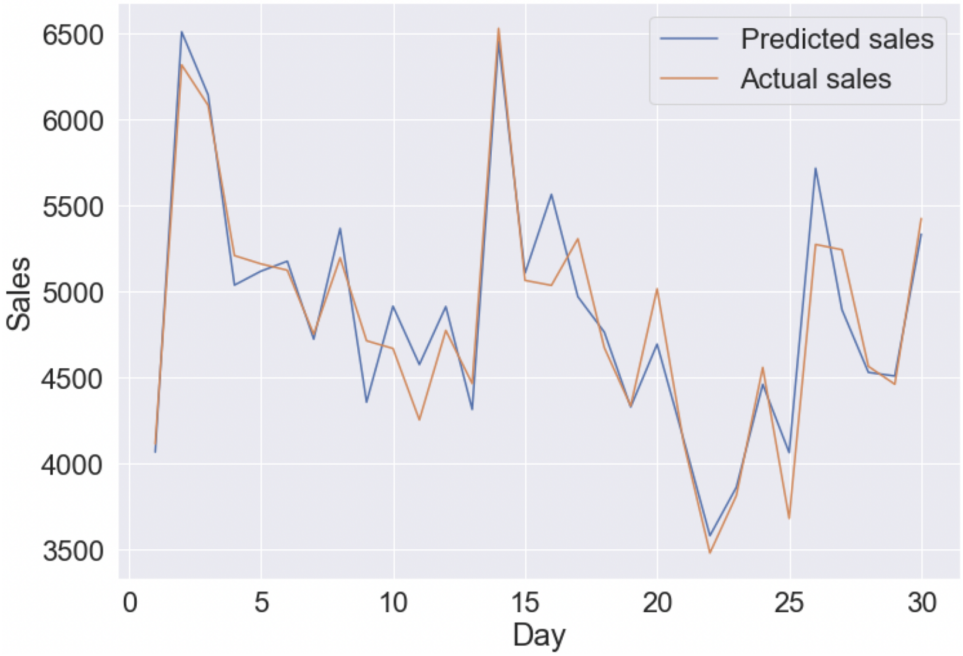


Fig 1. Predicted and Actual 30 day sales (XGBoost)

# Time-Series Sales Forecasting using Store, Promotion, and Competitor Data

## ARIMA

### Motivation:

ARIMA is widely used in time series analysis and forecasting. It predicts future values based on past values.The AR part indicates that the data is regressed on its own lagged values. The I part is used to stationarize the data. The MA part indicates that the value of the current time point is equal to the regression of the prediction error of the past several time points. This model is easy to use since it only requires the prior data of a time series to generalize the forecast, but it also has some limitations, for example, only limited factors are being considered in the analysis and the choosing of parameters is a bit subjective.

### Model Development:

Due to the requirement of stationary time series in I part of ARIMA model, we analyzed the plots which show the original sales of some store, the first order difference of sales, and the second order difference of sales to determine the parameter d and find the first order difference of sales is stationary enough, so we choose to set d = 1 for each ARIMA model applied to each store.
Then we divided all stores into 12 groups based on their type and competition distance. For each group, we used the average of sales for each day and drew the plot of autocorrelation function (ACF) and partial autocorrelation function (PACF) to determine the parameters p and q for the ARIMA model applied to all stores in this group.

### Model Evaluation:

After getting our ARIMA models for each store, due to the time limit, we choose one store in each category to evaluate our model by calculating the difference between the sum of the total predicted sales for the last 30 days and the sum of the total actual sales for the last 30 days. What's more, we also plot the curves of predicted sales and actual sales for the last 30 days in the same graph.

**Output**: (This is the output for one of the stores we chose.)

# Time-Series Sales Forecasting using Store, Promotion, and Competitor Data

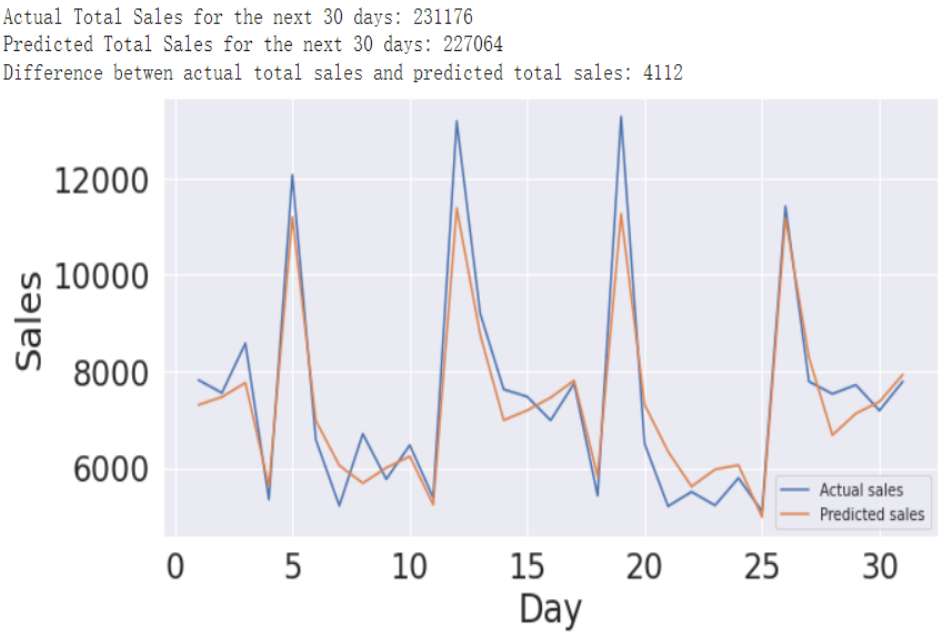


Fig 3. Predicted and Actual 30 day sales (Arima)

## Prophet

### Motivation:

Time-series data has been more popular than ever as numerous business companies are analyzing past data to make predictions for the future. The seasonality behind the time-series data projects the real characteristic of the modern business models. Understanding time-based patterns is now critical for any business. Facebook, as a cutting-edge tech giant, published an open-source library Prophet to help entrepreneurs analyze their business data without detailed implementation of the complicated models. The Prophet model is based on decomposable models that include trend, seasonality, and holidays. It gives us the ability to make time-series predictions without setting up complex parameters while maintaining a good prediction accuracy. As most of the business profit are sales/orders, which involve a huge amount of seasonality factors, Prophet becomes tremendously popular compared to complicated models such as LSTM.
Since our goal is to predict the future sales of stores, applying Prophet on our data would be an ideal fit for our situation.

### Model Development:

Unlike tree-based models or ARIMA models, Prophet is a highly integrated library that does most of the parameter tuning and back-end operation in a black box. Due to this reason, we re-processed the data based on the EDA analysis and designed the processed data to fit Prophet's requirements. Prophet takes the date object as a whole to fit into its backend

calculation, so, we reverse the One-hot encoded features back to its original form. The previously analyzed holiday factors will be input as a separate data object which later be put into the model fitting action. Thus, we derived the holiday features from the previous overall data processing steps to accommodate this requirement. All other factors stayed as they were in the previous processing steps.

**Output:**

With slight modification and manipulation on the data, we split and assigned the last 30 days of data to be as the test dataset while keeping the rest as training. We made this decision based on the analysis of the classic business model as monthly cycle profit would make sense for most grocery stores. Like previous tree models and ARIMA models, we first calculated the error rate for the test dataset, then visualized the difference between the prediction data and the actual sales record.

In the realistic business model, most of the future features will stay unknown since they haven't happened yet. Although tree models may have relatively high accuracy on each individual data point (date), they do not have the ability to predict the ones that are missing future feature values. In contrast, Prophet allows users to train the model on various features or factors that may have an impact on the prediction, but only takes date as future input (does not accept or ignore features other than date).


Fig 4. Predicted and Actual 30 day sales (FB Prophet)

Specifically here, we focus on the overall revenue over a specific time period to make more actionable decision-making. The Prophet model achieved an error rate around 2.5% on the

summed-up sales data. Although the graph shows quite different between each individual prediction and the actual sales record, the sum of the record maintained a relatively high accuracy and proved that Prophet is suitable for predicting the unknown-feature dataset.

## Conclusion:

Based on the previous EDA and model testing, we can conclude that all of the three models (Tree-based, ARIMA, Prophet) are able to detect, capture and predict the seasonality. All models eventually have strong forecast ability and good prediction accuracy. The tree-based models can accurately predict sales for each individual day if the features values are known. The tree based models take into account multiple features to make the prediction. The Prophet and ARIMA models on the other hand, due to their nature of prediction rely only on the past data and the future date (test dataset can only have forecast date as input), can predict the dataset with unknown future feature values well as we tested through the process. ARIMA models assume some sort of causal relationship between past values and past errors and future values of the time series. Facebook Prophet doesn't look for any such causal relationships between past and future. Instead, it simply tries to find the best curve to fit to the data, using a linear or logistic curve, and Fourier coefficients for the seasonal components. Hence, it does not perform as well as ARIMA even when both the models have the same set of input features. All three models are therefore rectified and proved to have the ability to forecast the time-series dataset accurately.