ROSSMANN SALES PREDICTION PROJECT

Mujtaba Ali & Prateek Sachdeva
Data science trainees,
AlmaBetter, Bangalore

Abstract:

Dirk Rossmann GmbH, commonly referred to as Rossmann, is one of the largest drug store chains in Europe with around 56,200 employees and more than 4000** stores. In 2019 Rossmann had more than €10 billion turnover in Germany, Poland, Hungary, the Czech Republic, Turkey, Albania, Kosovo and Spain. The company was founded in **1972 by Dirk Rossmann with its headquarters in Burgwedel near Hanover in Germany. The Rossmann family owns 60% of the company. The Hong Kong-based A.S. Watson Group owns 40%, which was taken over from the Dutch Kruidvat in 2004.

1. Problem Statement

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their

Unique circumstances, the accuracy of results can be quite varied.

- **Id -** an Id that represents a (Store, Date) duple within the test set.
- **Store -** a unique Id for each store.
- **Sales -** the turnover for any given day (this is what you are predicting).
- **Customers -** the number of customers on a given day.
- **Open -** an indicator for whether the store was open: 0 = closed,1 = open.
- **StateHoliday -** indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = none.
- **SchoolHoliday -** indicates if the (Store, Date) was affected by the closure of public schools.
- **StoreType -** differentiates between 4 different store models: a, b, c, d
- **Assortment -** describes an assortment level: a = basic, b = extra, c = extended.
- **CompetitionDistance -** distance in meters to the nearest competitor store.
- **CompetitionOpenSince[Month/Year]** - gives the approximate year and month of the time the nearest competitor was opened

- **Promo -** indicates whether a store is running a promo on that day.
- **Promo2 -** Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating.
- **Promo2Since [Year/Week] -** describes the year and calendar week when the store started participating in Promo2.
- **PromoInterval -** describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb, May, Aug, Nov" means each round starts in February, May, August, November of any given year for that store

2. Introduction

By using this project, we can predict the sales of Rossmann store based on the provided features. This method can be very useful for those who wish to grow their sales.

3. Steps involved:

- **Exploratory Data Analysis:**
  After loading the dataset we performed this method by comparing our target variable that is Sales with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

- **Null values Treatment:**
  Our dataset contains a large number of null values which might tend to disturb our accuracy hence we imputed them at the beginning of our project inorder to get a better result.

- **Encoding of categorical columns:**
  We used One Hot Encoding to handle nominal categorical variables. For ordinal variable, we used ordinal encoding.

- **Feature Construction:**
  We made three new features i.e. day, month, and year based on the existing date feature.

- **Feature Selection:**
  We checked multicollinearity using variance inflation factor, and dropped the features having high VIF coefficient.

- **Standardization of features:**
  We used Standard-scaler to scale the distribution, which was earlier right skewed to the normal. For

Outliers treatment we used robust-scaler.

- **Fitting different models:**
  For modeling we tried various regression algorithms like:

1. **Linear Regression**
2. **Lasso & Ridge(Regularization)**
3. **Decision Tree Regressor**
4. **Random forest Regressor**
5. **XGBoost Regressor**

- **Tuning the hyperparameters for better accuracy:**
  Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models
  Like Random Forest Regressor and XGBoost regressor.

- **SHAP Values for features**
  We have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn't put much weight on the performance of our model.

4. Algorithms:

- **Linear Regression:**

  By using linear regression, we were able to obtain 89% accuracy.

- **Ridge and Lasso(Regularization):**
  We are able to obtain 89% accuracy by using Regularization. We observed that accuracy did not improve much.

- **Decision Tree Regressor:**
  By using decision tree regressor, we got 97% accuracy.

- **Random forest:**
  We were able to obtained 98% accuracy using random forest.

- **XGBoost Regressor:**
  We were able to obtained 93% accuracy using xgboost regressor.

5. Model performance:

- **R2_Score:**

  R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent

variable on a convenient 0 – 100% scale.

After fitting a linear regression model, you need to determine how well the model fits the data. Does it do a good job of explaining changes in the dependent variable? There are several key goodness-of-fit statistics for regression analysis. In this post, we'll examine R-squared (R2), highlight some of its limitations, and discover some surprises. For instance, small R-squared values are not always a problem, and high R-squared values are not necessarily good!

6. Hyper parameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

- **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

7. Conclusion:

That's it! We reached the end of our exercise.
Starting with loading the data so far we have done EDA, null values treatment, encoding of categorical columns, feature selection and then model building.
In all of these models our accuracy revolves in the range of 89 to 98%.
And there is no such improvement in accuracy score even after hyperparameter tuning.
So the accuracy of our best model is 98% which can be said to be good for this large dataset.