Technical Documentation

ROSSMANN SALES PREDICTION

****

# Introduction

A major challenge for large retailers is to address the needs of the consumers more effectively on a local level while maintaining the efficiencies of central distribution. As the demand for mass customization by consumers grows, methods focused on store-level optimization increase in value. Prediction of sales is an important application of machine learning in the retail space. Given accurate predictions, retailers can manage dynamic pricing, staff rostering, and inventory so as to maximize profit and improve the customer experience.

An accurate forecast of sales allows retail outlets to answer questions such as:

* **Can dynamic pricing prove to be a better way to maximize profit?**
* **Do we have enough stock to satisfy demand without being overstocked?**
* **What are the most important factors that affect sales, and how can we optimize them?**

**Problem Statement**

Data provided by Rossmann gives various information about 3,000 drug stores in 7 European countries. Currently, Rossmann store managers were tasked with predicting their daily sales 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.

We are provided with historical sales data for 1,115 Rossmann stores. Our task is to forecast the "Sales" column for the test set with the help of given datasets.

**Data Description**

The dataset contains information about the stores and their sales. Two datasets are provided.

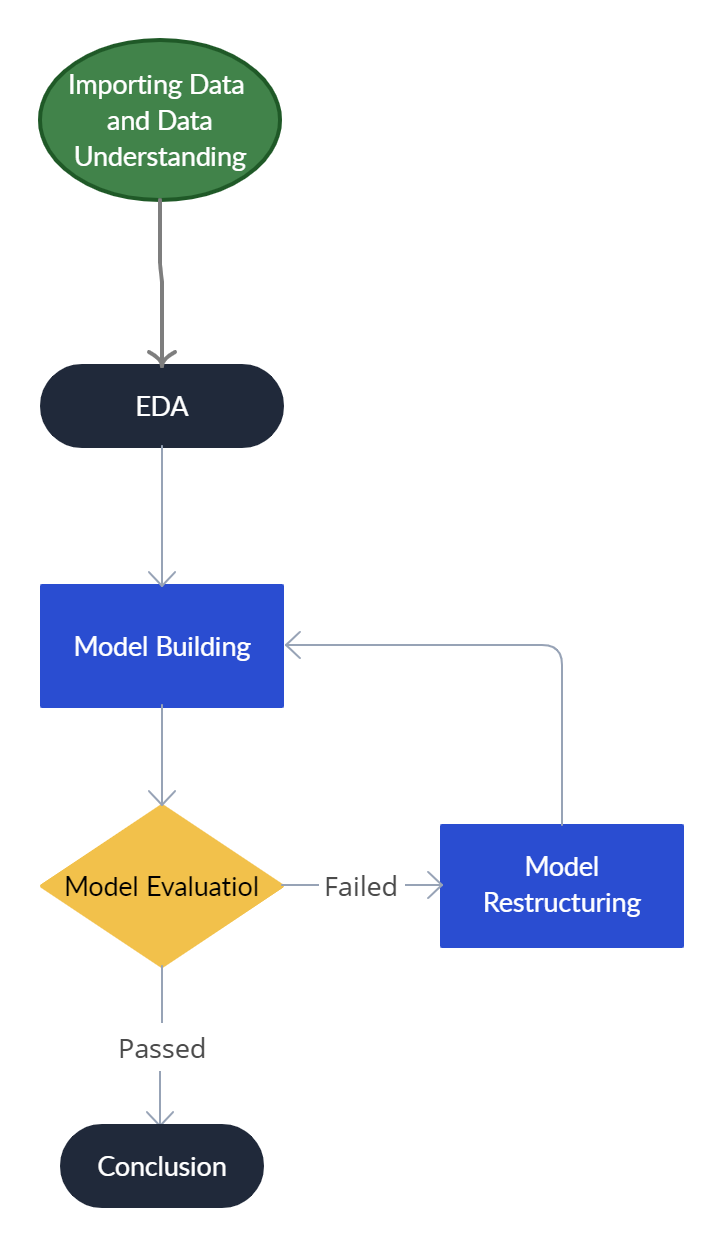
* **Rossman\_stores\_data.csv**

historical data including Sales

* **store.csv**

information about the stores'

**Project Roadmap**



**Libraries used:**

**Pandas:** Pandas is a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries. It has an extremely active community of contributors.

Pandas is built on top of two core Python libraries Matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, *pandas' .plot()* combines multiple Matplotlib methods into a single method, enabling you to plot a chart in a few lines.

**NumPy:** NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

**Matplotlib** - Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

## SEABORN - Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of the Matplotlib library and is also closely integrated into the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

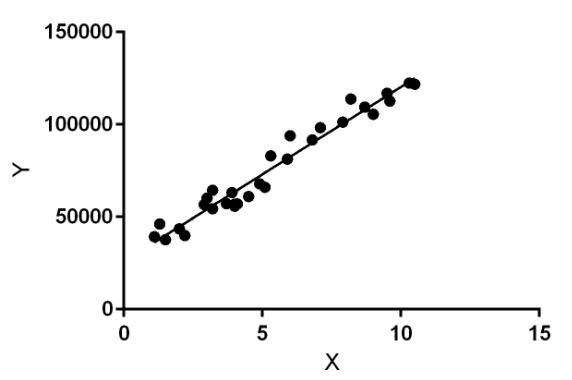
## **BINARYENCODER:** Binary Encoding is a special case of One Hot Encoding in which binary digits are used for encoding i.e. 0 or 1. For example for 7 binary code is 111. This technique is preferable when there are more number categories. Suppose you have 100 or more different categories then One hot encoding will create 100 or more different columns, but binary encoding only need 7 columns to represent it.

**Approach**

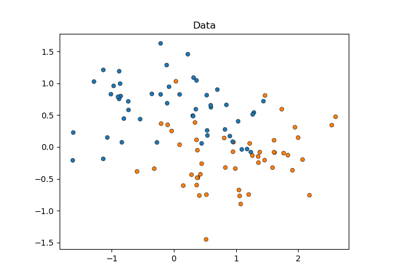
* Loading our dataset and importing all the useful libraries
* Understanding Data
* Merging both Ross and Sales DataFrame and working on them
* Understanding Merged Dataset
* Exploratory Data Analysis
* Feature Engineering
* Outlier Detection and Removal
* Pre-Processing Data
* Creating Models
* Standardization of features
* Machine Learning Data Modeling (for our Prediction)

**Algorithms Used**

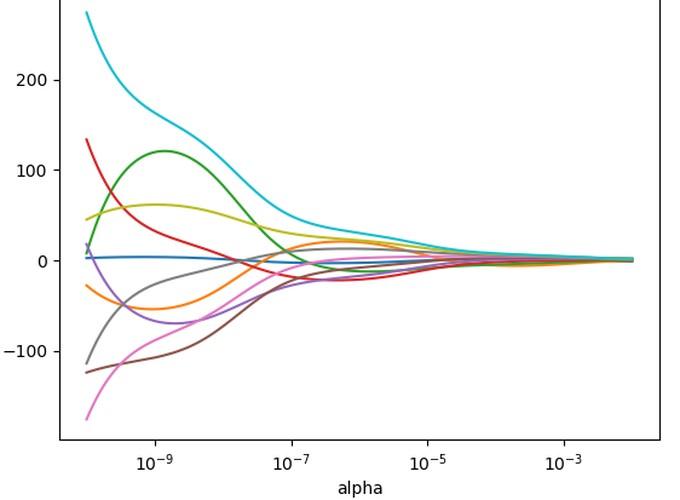
1. **Linear Regression:** Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. A linear regression line has an equation of the form ***Y = a + bX***, where ***X*** is the explanatory variable and ***Y*** is the dependent variable. The slope of the line is ***b***, and ***a*** is the intercept (the value of ***y*** when ***x*** = 0).

****

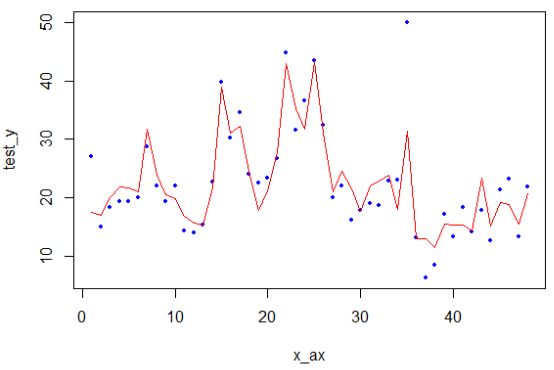
1. **Linear Regression With Grid Search CV:** Grid Search CV is a library function that is a member of sklearn’s model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

****

1. **Lasso Regression:** Lasso Regression is used for the selection of the subset of variables. It provides greater prediction accuracy as compared to other regression models. Lasso Regularization helps to increase model interpretation. The less important features of a dataset are penalized by the lasso regression. The coefficients of this dataset are made zero leading to their elimination. The dataset with high dimensions and correlation is well suited for lasso regression.

****

1. **Ridge Regression:** Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.
2. **Gradient Boosting Regression:** Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error.

****

**Conclusion**

* Some features were extremely important for the Prediction like Customer, Promo, Open, etc
* As the Dataset was quite large the model generalized well, and chances of overfitting and underfitting were reduced.
* In order to earn more focus company should focus more on Store d which generates the highest sales for the firm.
* Gradient boosting regressor seems to be relatively very efficient with approximately 96% of r2 score
* ordinary least square regression performed well with approximately 89% accuracy in both the train and test set, thus model wasn't looking highly biased or at high variance.
* lasso and ridge didn't show any effective results in comparison to OLS
* The company earns the most from Store type d thus it must maintain this and should focus more on low-performing stores like store b.
* On Sunday the market might be closed, resulting in Extremely fewer sales.
* The Data could also be used for Time Series Analytics but as the data is from before 2015, it is less likely for the firm to use this in future predictions.
* For around 0.65 million stores the competitor is present within a range of 5KM which shows that good competition is present in most of the cases but most of the competitors are established after the year 2000.