**Rossmann Sales Forecasting**

**Abstract**

Rossmann Stores Sales Forecasting is originated from one of Kaggle’s Machine Learning competitions. The task is to predict 6 weeks of daily sales for 1,115 Rossmann stores that are located across whole Germany. In this paper, we will describe how to use existing data set and external dataset in training a forecasting machine learning model for this.

**Project Overview**

Sales forecasting or future prediction base on time series [1] dataset is common in many business cases. The better predictions can make business more efficient in early planning like stock preparation and human power relocation etc. This project is focusing on predicting a chain medicine beauty store called Rossmann daily sales for up to 6 weeks in advanced. Rossmann stores locate across multiple states in Germany. In this task, it proves 1,115 stores properties data and daily sales record from 2013 to 2015. It turns out that predicting numeric continuous data is a regression [2] task in machine learning area, and also this is a supervised learning [3]. Regression Analysis is to focus on the relationship between dependent variable and one or more independent variables. In machine learning area, there are many regression algorithms that we can use for this purpose, linear regression[4], SVM regression[5] and Decision Tree[6], all of which have their pros and cons.

**Project Statement**

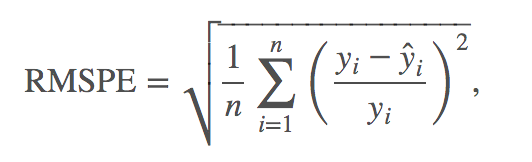
In this project, predicting daily sales for up to 6 weeks in advanced is a supervised regression problem. By using machine learning models to predict the sales numbers from a set of vary dependent features. The targets that need to be predicted is in test.csv dataset. Ultimately, the task is about data analysis and data prediction.

A regression problem need to be justified in performance, we need some metric methods for evaluating our models performance. For general regression problems, we have several evaluation metrics for those. Like Mean Absolute Error [7], Mean Squared Error [8] and R2 squared [9] etc. In this Kaggle competition, we use Root Mean Square Percentage Error (RMSPE) [10]. In this project our goal is to minimize the RMSPE score as lower as possible. Bench mark score will be top 300 in this Kaggle competition.

I will follow data analysis process to apply in this project, which contains data exploratory, data cleaning, feature engineering, Algorithm selection, and model fine tuning. In data exploratory stage, I will load the existing data into Python Pandas data structure , and apply some data exploratory function to take a look at the data first, and then I will clean the data as needed, since good predicting model comes from good data. After all these, I will try to use find out the relationship and coefficient relationship among the dataset, and apply feature engineering on this, also the most important step in this project. Then I will apply the selected features to candidate algorithm and compare their performance by metrics. The expectation for this project is the performance metrics as low as 0.117, of course the lower the better. Meaning your prediction is closer to the true value.

**Metric**

In this project, we will use Root Mean Square Percentage Error (RMSPE) for evaluating the model performance. RMSPE is suitable for evaluating regression problem. Basically, it will calculate how close the perdition value to actual value by putting punishment (scale) on large error. Our model will use SGD [11] method to decrease the value of RMSPE, the perfect situation would be 0 as no errors at all. Here with the formula for RMSPE:

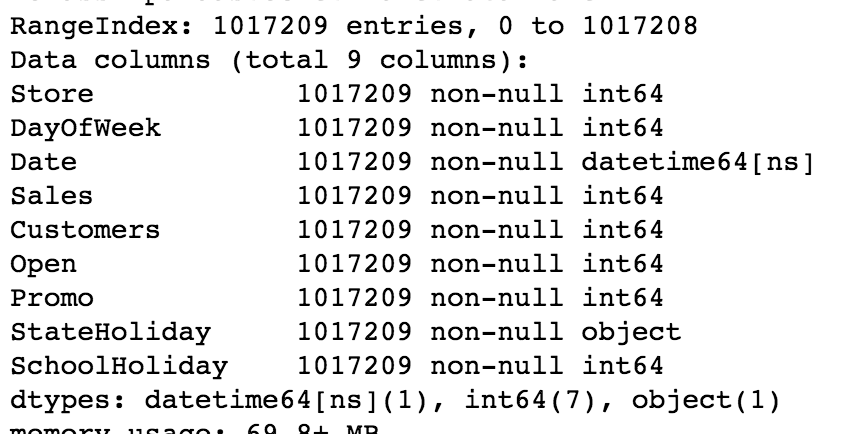
****

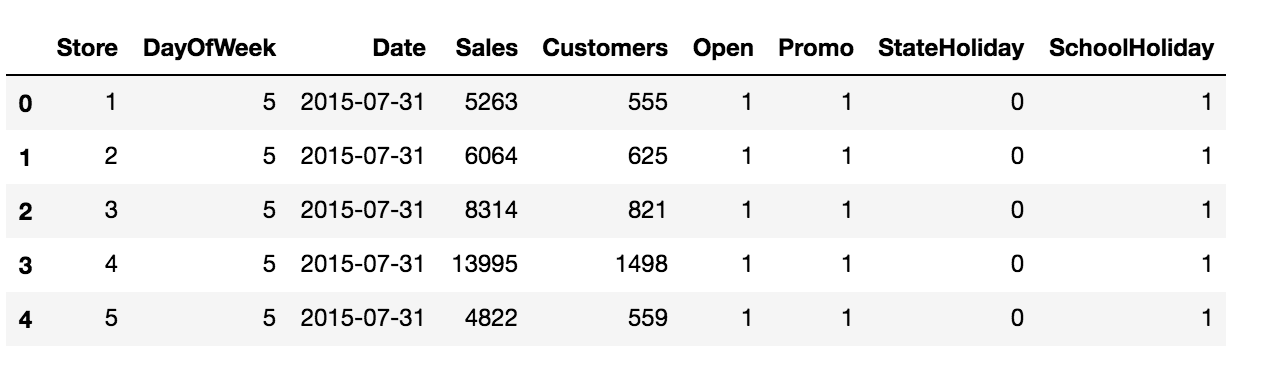
Where y\_i is the actual target value for a record, in this project, it will be the sales of a single store on a single day and y\_hat is the predict target value.

**Analysis**

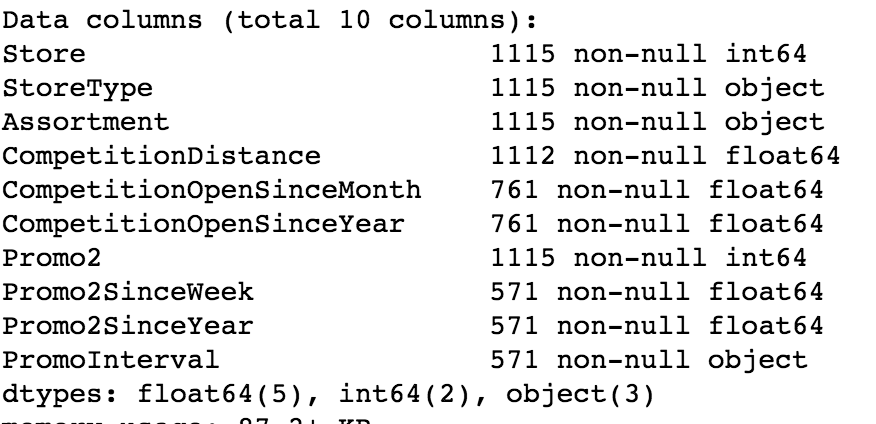
**Data Exploration**

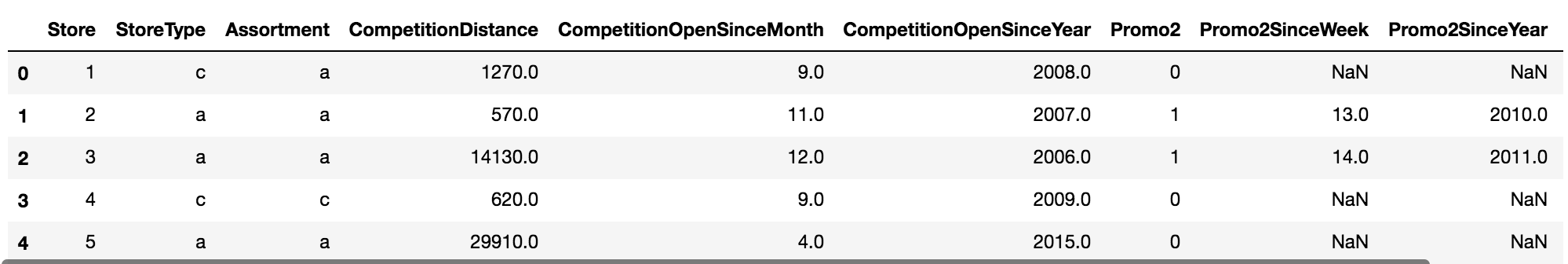
In this project, it provides several datasets for competition purpose, and you are also allow to use external dataset as you can post the source where it comes from. I will first try to use the available dataset. There are three available datasets in the project: train.csv, store.csv and test.csv. Train.csv and store.csv are the training data for building the model to predict the targets in test.csv. Train.csv records 1,017,209 rows of different stores daily sales records from 2013/01/01 to 2015/07/31 including numbers of customers, store status and days property as state holidays or school holiday. We find that the data in train dataset is complete without missing data. Looking into the dataset, we find 9 features, amount them, Store, DayOfWeek, Open, Promo, StateHoliday and SchoolHoliday are category data, the rest I will treat them as numeric data. Sales is the one we are going to predict, so I will label the Sales as target for training in the later process.





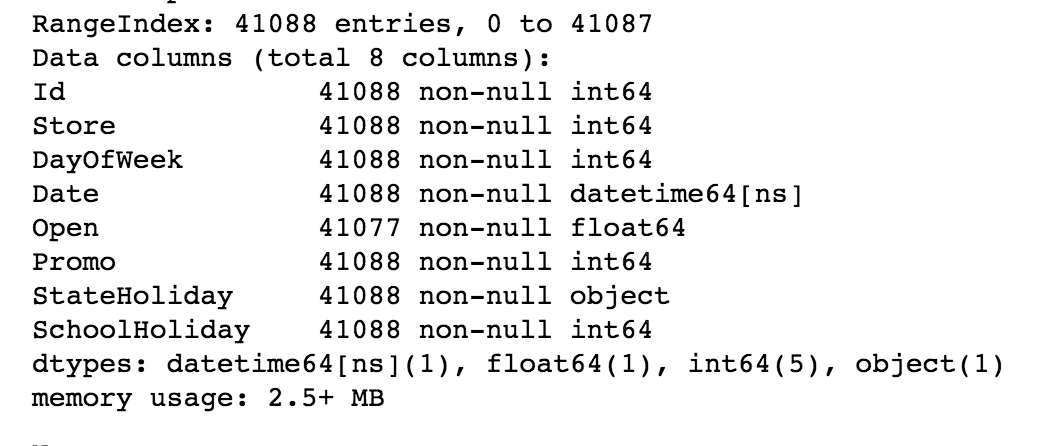
In store.csv, it records 1,115 stores properties value which contains store ID, store type, assortment type, competition information from distance to time beginning, promotion activities information. We can find that there are some missing records in this dataset, for some of reasons like absent or non-applicable, which we will handle this later. Amount them, Store, StoreType, Assortment, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2 Promo2SinceWeek, Promo2SinceYear, PromoInterval, are category date, other are numeric data.

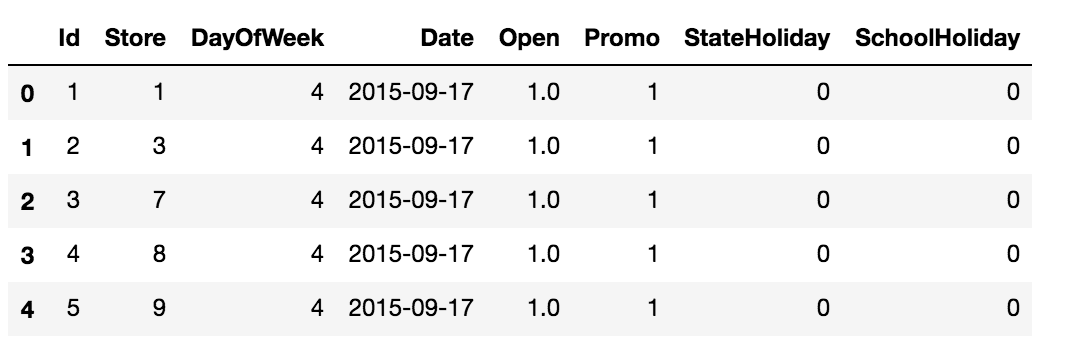
****



For StoreType, Assorment and PromoInterval, they are represented by character. Value in character is difficult for machine learning model to recognize or even know the importance. So we need to do some preprocessing at later step.

In test.csv, it has 41,088 records that about 1,115 store daily record except sales and customers. There are 11 missing data, we will also need to fix this at later step.





**Exploratory Visualization**

**Reference**

1. Time series <https://en.wikipedia.org/wiki/Time_series>
2. Regression Analysis <https://en.wikipedia.org/wiki/Regression_analysis>
3. Supervised Learning <https://en.wikipedia.org/wiki/Regression_analysis>
4. Linear regression http://scikit-learn.org/stable/modules/linear\_model.html
5. SVM <http://scikit-learn.org/stable/modules/svm.html>
6. Decision Tree <http://scikit-learn.org/stable/modules/tree.html>
7. Mean absolute error <https://en.wikipedia.org/wiki/Mean_absolute_error>
8. Mean squared error <https://en.wikipedia.org/wiki/Mean_squared_error>
9. Coefficient of determination <https://en.wikipedia.org/wiki/Coefficient_of_determination>
10. Root Mean Square Percentage Error <https://www.kaggle.com/c/rossmann-store-sales#evaluation>
11. Stochastic gradient descent https://en.wikipedia.org/wiki/Stochastic\_gradient\_descent