**Rossmann Store Sales**

1. **Background**

Rossmann, founded in 1972, is the largest grocery store in Germany with more than 3000 stores in 7 European countries. From time to time, the stores organize short-term promotions as well as continuous promotions to increase sales. In addition, store sales are influenced by many factors, including promotions, competition, school and national holidays, seasonality and cyclicality.

1. **Data presentation**

### The data is based on 1115 Rossmann chain stores, with a total of 1017209 sales data (27 characteristics) recorded from January 1, 2013 to July 2015.

The data set covers a total of four files.

- train.csv: historical data with sales volume

- test.csv: historical data without sales

- sample\_submission.csv: a sample file submitted in the correct format

- store.csv: some additional information about each store

The data in train.csv contains a total of 9 columns of information.

- store: is the id number of the corresponding store

- DayOfWeek: represents the number of days per week that the store is open

- Data: is the date when the corresponding sales were generated

- Sales: is the historical data of sales

- Customers: is the number of customers who came into the store

- Open: indicates whether the store is open or not

- Promo: indicates whether the store has a promotion on that day

- StateHoliday: and SchoolHoliday indicate whether it is a national holiday or a school holiday, respectively.

1. train.csv

The data overview in the lower part of Kaggle's data page provides a general view of the distribution of each data and a sample of some of the data as follows.

图形用户界面, 应用程序

描述已自动生成

1. test.csv

The data columns in test.csv are almost the same as those in train.csv, but the columns Sales (i.e., sales data) and Customers (user traffic) are missing. Our ultimate goal is to predict the missing Sales data in test.csv using the additional information in test.csv and store.csv.

The data distribution of test.csv shows that Sales and Customer data, which are strongly associated with Sales, are missing compared to the above.

The data distribution and some example data are as follows.

图形用户界面, 应用程序

描述已自动生成

1. sample\_submission.csv

The results file sample\_submission.csv contains only the id and Sales columns, and this file is the standard format template for submitting our predicted answers to Kaggle's adjudicator.

In Python, we just need to open this file and put the predicted data into the Sales column in order, then use Dataframe.to\_csv('sample\_submission.csv') to save the sample\_submission.csv with the predicted data locally and prepare for subsequent upload. submission.csv with forecast data can be saved locally and prepared for subsequent upload.

1. store.csv

As you can see, there are store ids in train.csv and test.csv, and the details of these store ids are in store.csv, where some store location information and promotion information are recorded.

The data distribution of store.csv, you can notice that there are many discrete category labels.

The data distribution and some example data are as follows.

图形用户界面

低可信度描述已自动生成

Where:

- Store: corresponds to the number of the store.

- StoreType: the type of store, there are a, b, c, d four different types of stores. You can imagine it as a flash store, general business store, flagship store, or mini store, which is the type of our life.

- Assortment: a, b, and c are used to describe the level of combination of products sold in the store. For example, there must be a big difference between the products in the flagship store and the mini store.

- Competition Distance, Competition Open Since Year, Competition Open Since Month: indicate the distance of the nearest competitor's store, the opening time (in years), and the opening time (in months), respectively.

- Promo2: Describes whether the store has a long-term promotion.

- Promo2 Since Year in Promo2 Since Week: Indicates the year and calendar week, respectively, when the store started participating in the promotion.

- Promo Interval: Describes the continuous interval from promo2, named after the month in which the promotion restarted.

1. **Project Objectives**

After understanding the data we need to clarify the purpose of our project, in Rossmanns sales forecasting we need to use historical data, i.e. data in intrain.csv for supervised learning. The trained model is inferred (predicted) using the data in intrain.csv, and the predicted data is submitted to Kaggle as sample\_submission.csv

The predicted data is submitted to Kaggle for scoring in the format of sample\_submission.csv. This process can also be combined with the additional information in store.csv to enhance the ability of our model to obtain data.

1. **Evaluation Criteria**

The evaluation metric adopted for the model is the Root Mean Square Percentage Error (RMSPE) metric recommended by Kaggle in the competition.

手机屏幕截图

描述已自动生成

where：

  represents the real sales of the store on that day.

  represents the corresponding predicted sales.

  represents the number of samples.

If there is any day with zero sales, then it will be ignored. The smaller the RMSPE value is, the smaller the error is and the higher the score will be.

1. **Project Process**

**Step 1: Loading data**

The Rossmann scenario modeling data contains many information dimensions, such as the number of customers, holidays, and so on. It can also be judged as a typical regression-type modeling problem in supervised learning based on its task objective. We first do subsequent analysis of the loaded data before mining the modeling.

The DataFrame.info() operation allows you to view the basic information of the DataFrame data (value distribution, missing value situation, etc.)

**Step 2: EDA exploratory data analysis**

The scale of the data involved in this case is relatively large, and we cannot directly view the data characteristics by naked eyes, but the understanding of the data distribution characteristics can help us achieve better results in the subsequent mining and modeling. Here we will use Pandas, Matplotlib, Seaborn and other tools introduced before to analyze and visualize the data for understanding.

The IDE we use for this part is Jupyter Notebook, which is more convenient for interactive plotting to explore data characteristics.

- Hint: You may have line plots, univariate distribution plots, bivariate joint distribution plots, box line plots, heat maps

**Step 3: Data pre-processing (missing values)**

Some of the processing methods for missing values include:

* - Remove fields (remove columns containing missing values).
* - Fill in the missing values (fill in the mean, median, or fit fill, etc.).
* - Marking missing values by marking them as special values (e.g. -999) or adding a new column to mark whether a field is missing.

**Step 4: Feature Engineering**

* - Time features, extracting information such as year, month, day of the week
* - Character features are converted to numbers

**Step 5: Benchmark model and evaluation**

**Define the evaluation criterion function**

* Since continuous values need to be predicted, a regression model needs to be used. Since this project is a Kaggle competition, the test set is evaluated using Root Mean Square Percentage Error (RMSPE), so only RMSPE can be used here. The formula for calculating RMSPE is

手机屏幕截图

中度可信度描述已自动生成

where and  are the true and predicted values of the i-th sample label, respectively.

**Baseline model evaluation**

* We construct a regression tree model as the base model for modeling and evaluation. The regression tree we directly use SKLearn's DecisionTreeRegressor, with K-fold cross-validation and grid search for tuning, the main adjustment hyperparameter is the maximum depth max\_depth of the tree.
* We note that the evaluation criterion here is neg\_rmspe, which is the appropriate evaluation criterion for incoming model tuning. GridSearchCV defaults to finding the parameter with the largest scoring\_fnc, and directly uses the rmspe metric, the smaller the value, the better the model effect, so it should be taken as negative, thus the larger the value of neg\_rmspe, the better the model accuracy.

**Step 6: XGBoost Modeling**

### Model parameters

### XGBoost is a more powerful model with more adjustable parameters, we mainly adjust the following hyperparameters.

### - eta: learning rate.

### - max\_depth: the maximum depth of a single regression tree, smaller leads to underfitting, larger leads to overfitting.

### - subsample: between 0 and 1, which controls the proportion of random sampling for each tree. By decreasing the value of this parameter, the algorithm will be more conservative and avoid overfitting. However, if this value is set too small, it may lead to underfitting.

### - colsample\_bytree: between 0 and 1, used to control the proportion of randomly sampled features per tree.

### - num\_trees: the number of trees, i.e. the number of iteration steps.