Rossmann 销售预测

Machine Learning Engineer Nanodegree

Capstone Proposal

GuanLi Liu
April 20th 2018

Domain Background

Forecast Rossmann Store Sales Project is a Kaggle competition which forecasts sales using store, promotion, and competitor data[1]. In statistics, prediction is a part of statistical inference. One particular approach to such inference is known as predictive inference, but the prediction can be undertaken within any of the several approaches to statistical inference. When information is transferred across time, often to specific points in time, the process is known as forecasting.

It is obviously that this is a time series related problem. A time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data[2]. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. It is essential in our daily life like the temperature forecast is quite convenient for our life. The forecast of stock price is helpful for some business man to earn more money. Also the selling forecast can help the merchants to cut cost.Time series forecasting is the use of a model to predict future values based on previously observed values. In this problem we should analysis the dataset first and then forecast the result by our model.

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[2]. The goal of the Project is to predict 6 weeks of daily Sales in 1115 stores located in different parts of Germany based on 2.5 years of historical daily sales.

Datasets and Inputs

There are four data Files proviede by Kaggle:

> - train.csv - historical data including Sales
> - test.csv - historical data excluding Sales
> - sample_submission.csv - a sample submission file in the correct format
> - store.csv - supplemental information about the stores

Most of the fields are self-explanatory like DayOfWeek and Date. While the following are descriptions for those that aren't.

> - 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

All of the datasets should be used and we can choose some essential information from them. Customers and Open is of vital importance. Because they are positively correlated with sales. Besides, the information about promotion is also useful because people tend to buy more goods. The type, Competition Distance, holiday also have some influence on the sale. So, in the following time, I will try those given information to predict the sale.

However, there are some flaws in the dataset. For example, as I watched in the forum that Store 622 has 11 missing values in the Open columns and most people assume the store is open[3]. If it's closed then sales will be 0 and it won't count toward the score. But if it's open and you predict 0 then you're in for some heavy penalty. Additionally, there is a 6 month gap where we have a smaller number of stores reporting sales revenue[4]. We also have to impute the missing values to have a stable result. Finally, we have to deal with the value of some fields which are not suitable for the train model. Like StoreType and Assortment. The values of them are letters we should change letter to values like 1-4 for StoreType and 1-3 for Assortment respectively.

Solution Statement

I suppose the problem can be solved by using Random Forest. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples: For $b = 1, \ldots, B$: Sample, with replacement, n training examples from X, Y; call these X_b, Y_b. Train a classification or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x' : $\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$. The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable, these features will be selected in many of the B trees, causing them to become correlated[5].

Benchmark Model

The Benchmark Model predicts the geometric mean of past sales grouped by Store, DayOfWeek, Promo[6]. It is a big improvement over the original benchmark, median Sales grouped by Store, DayOfWeek. The Benchmark Model was found in the Kernels part of Kaggle which had a approving result. The Private Score is 0.14825 and the Public Score is 0.13952. The model was quite easy to be implemented that only including 10 lines of code.

Evaluation Metrics

Since it's a prediction related question, the evaluation metric can not be accuracy or Precision-Recall. There are some evaluation metrics like MAE(Mean Absolute Error), MSE(Mean Square Error), RMSE(Root Mean Square Error)[7]. As for RMSE, it uses the average error, and the average error of abnormal point is sensitive, if return to implement for the return of a certain point value is not very reasonable, then its error is relatively large, so as to have a great effect on the RMSE value, the average is not robust.

I think submissions should be evaluated on the Root Mean Square Percentage Error (RMSPE) which is introduced in the Kaggle official website[1]. The RMSPE is calculated as $RMSPE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{y_i - \hat{y_i}}{y_i} \right)^2}$. where y_i denotes the sales of a single store on a single day and $\hat{y_i}$ denotes the corresponding prediction. Any day and store with 0 sales is ignored in scoring.

Project Design

The workflow includes four parts. Data analyzing, Model training, benchmark running, result comparing and visualizing.

Firstly, as I mentioned before, this is a Time Series Prediction question. So we should analyze the data before we decide the input of the model[8]. For example, we need to identifyi the important factors which are the most likely to affect the outcome of the prediction. Besides, the analysis of per store type and correlational analysis of stores activity, the seasonal decomposition, trends, autocorrelation are also important. Besides, there are some flaws in the datasets that we should solve before training the model. Like the Store 622 has 11 missing values in the Open columns and most people assume the store is open. The value 0 means it's closed that it won't count toward the score, while it's open and you predict 0 then you're in for some heavy penalty. And there is a 6 month gap where we have a smaller number of stores reporting sales revenue. We also have to impute the missing values to have a stable result. Additionally, we have to deal with the value of some fields which are not suitable for the train model. For example, StoreType and Assortment. The values of them are letters we should change letter to valus like 1-4 for StoreType and 1-3 for Assortment respectively.

Secondly, after dealing with the datasets we must have neater datasets which are more suitable and sensible for the training of model. We may use PCA to choose the principle components first. Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components[9]. After picking some fields from the datasets we will use *sklearn.ensemble.RandomForestRegressor*[10] to train a model. There are some important parameters like *max_depth*, *min_samples_split* and so on that we have to train to generalize the dataset better. In order to have a better model. We have to try different parameters of the fields and also the configuration of the random forest. *sklearn.model_selection.GridSearchCV*[11] is a helpful tool for us to find a better model. After that, we will generate the experimental results for the result visualization.

Thirdly, we have to use the benchmark model to train the model and evaluate the result with the same circumstance of our method. The Benchmark Model predicts the geometric mean of past sales grouped by Store, DayOfWeek, Promo. which improves the original benchmark, median Sales grouped by Store, DayOfWeek.

Finally, visualize the result of the two algorithms so that we can know the quality of the result clearly. The visualization includes some parts: the line chart of the whole trend and what we predicted at the next to the given value. Also the predicted from our model and the benchmark model respectively. Besides, the score of the model with different parameters should be compared and checked in the diagram.

To sum up, it is my workflow and what I will do in the following time.

Reference

1. https://www.kaggle.com/c/rossmann-store-sales
2. https://en.wikipedia.org/wiki/Time_series
3. https://www.kaggle.com/c/rossmann-store-sales/discussion/17229
4. https://www.kaggle.com/nsecord/filling-gaps-in-the-training-set
5. https://en.wikipedia.org/wiki/Random_forest
6. https://www.kaggle.com/shearerp/store-dayofweek-promo-0-13952/code
7. https://en.wikipedia.org/wiki/Root-mean-square_deviation
8. https://www.kaggle.com/elenapetrova/time-series-analysis-and-forecasts-with-prophet
9. https://en.wikipedia.org/wiki/Principal_component_analysis
10. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor
11. http://scikit-learn.org/stable/modules/grid_search.html#exhaustive-grid-search