Kaggle Challenge: Predicting Rossmann Store Sales

Mingu Jo
University of California, Berkeley

Wonjohn Choi
University of California, Berkeley

1 Abstract

The objective of this project is to predict 6 weeks of daily sales for 1115 Rossmann stores located across Germany using data about stores' past daily sales, competitor, etc (the list of available data will be introduced more in depth later in this paper). The data was provided by Rossmann through Kaggle. The motivation of this project is the following: by reliably predicting daily sales, store managers can create effective staff schedules (by assigning more staffs during busy days). In order to predict daily sales, we first preprocessed the data and applied several methods (multivariate linear regression, random forest, generalized linear regression, etc). Then, we compared the methods' predictive power by computing Root Mean Square Percentage Error (RMSPE). We found that TODO:X performed the best with a RMSPE score of TODO:XXX.

2 Introduction

(TODO: literature review) Is the literature review in the Introduction informative and organized? text.

More fascinating text. Features[1] galore, plethora of promises.

2.1 Dataset and Preprocessing

Training and test dataset are given by Rossmann through Kaggle. The training set is composed of two different parts: one is 1̃0 million observed daily sales of different stores from 2013 to 2015, and the other is supplementary information for each store. We merged the two different parts by stores to have the training set. The training set had originally 15 features, but we extended to 19. We separated out year, month of the year, and day of the month from the feature d̈ate.̈

Also, we computed average sales for each store. In addition, we noticed the feature C̈ompetitionDistanceïs highly skewed to right. For the sake of linear regression, we decided to take log transformation on the feature C̈ompetitionDistanceẗo make sure that the relationship between explanatory variables and dependent variable is approximately linear.

Because of several missing values in the training set, we filled them based on our logic. - There are some stores on some date which do not have any competitors. In this case, the competitor information such as C̈ompetitionDistancef̈or this specific observation are missing. Therefore, we assigned some big number as 1000000 for C̈ompetitionDistanceẗo weaken its impact. In the similar way, we filled missing values for C̈ompetitionOpenSinceYearänd CompetitionOpenSinceMonth - There are some stores on some date which are not in the continuing promotion. In this case, the feature P̈romo2ïs 0, and Promo2Since

$$Year/Week$$

are missing. We filled them with the lowest possible value to weaken their impact.

2.2 Exploratory Data Analysis

To check any distinctive relationships between variables, we drew a pearson correlation coefficient plot using corrplot library. There are several interesting facts we captured. - D̈ayOfWeekänd S̈alesäre negatively correlated. This implies that, as D̈ayOfWeekäpproaches to Saturday and Sunday, sales would decrease because most drugstores would close on these days. - L̈ogCompetitionDistanceänd S̈alesäre negatively correlated. In other words, lower distance to the competitor implies slightly higher sales. We assume this could occur because stores with a close distance to the competitor are mostly located in crowded regions such as a downtown city with higher sales overall.

3 Methods

We started off by setting a benchmark to obtain a baseline for the prediction. Then, we took 3 different approaches to outperform the benchmark and former model.

3.1 5-fold Cross Validation

To generalize our predictions to limit problems of overfitting on the training set, we decided to ramdomly partition the original training observations into 5 equal sized subsamples. The training set is comprised of 4 equal sized subsamples and the rest as the validation set. For the purpose of reproducibility and comparison of the predictive powers of different models, we set the seed. Our general process would be as the following: 1. Repeat cross-validation process 5 times with each of the 5 subsamples used exactly once as the validation set. 2. Compute the average of 5 different validation error rate. 3. Predict sales on the test set using the model. 4. Submit on Kaggle website to get the test error rate. 5. Compare the average validation error rate and the test error rate among differnent models.

3.2 Error Metric

As the metrics that Kaggle uses to compute test error is the root-mean-squared-percentage-error (RMSPE), we used the same metrics to compute our validation error. It's the following:

where y_i is a store's sales on a single day, $yhat_i$ is the prediction, and n is the number of days.

3.3 Benchmark

Our benchmark is simply the average sales of each store. This gives us the validation error rate of : and the test error rate of :

3.4 Linear Regression

3.4.1 Feature Selection

3.4.2 Lasso

3.5 Random Forest

Some embedded literal typset code might look like the following :

```
int wrap_fact(ClientData clientData,
              Tcl_Interp *interp,
              int argc, char *argv[]) {
    int result;
    int arg0;
    if (argc != 2) {
        interp->result = "wrong # args";
        return TCL_ERROR;
    }
    arg0 = atoi(argv[1]);
    result = fact(arg0);
    sprintf(interp->result,"%d",result);
    return TCL_OK;
}
```

Figure 1: Wonderful Flowchart

Now we're going to cite somebody. Watch for the cite tag. Here it comes [**?**, **?**]. The tilde character (˜) in the source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

4 Supplementary Methods

4.1 First SubSection

Here's a typical figure reference. The figure is centered at the top of the column. It's scaled. It's explicitly placed. You'll have to tweak the numbers to get what you want.

This text came after the figure, so we'll casually refer to Figure 1 as we go on our merry way.

4.2 New Subsection

It can get tricky typesetting Tcl and C code in LaTeX because they share a lot of mystical feelings about certain magic characters. You will have to do a lot of escaping to typeset curly braces and percent signs, for example, like this: "The `%module` directive sets the name of the initialization function. This is optional, but is recommended if building a Tcl 7.5 module. Everything inside the `%{, %}` block is copied directly into the output. allowing the inclusion of header files and additional C code."

Sometimes you want to really call attention to a piece of text. You can center it in the column like this:

```
_1008e614_Vector_p
```

and people will really notice it.

The noindent at the start of this paragraph makes it clear that it's a continuation of the preceding text, not a new para in its own right.

Now this is an ingenious way to get a forced space. `Real *` and `double *` are equivalent.

Now here is another way to call attention to a line of code, but instead of centering it, we noindent and bold it.

```
size_t :  fread ptr size nobj stream
```

And here we have made an indented para like a definition tag (dt) in HTML. You don't need a surrounding list macro pair.

> `fread` reads from `stream` into the array `ptr` at most `nobj` objects of size `size`. `fread` returns the number of objects read.

This concludes the definitions tag.

4.3 How to Build Your Paper

You have to run `latex` once to prepare your references for munging. Then run `bibtex` to build your bibliography metadata. Then run `latex` twice to ensure all references have been resolved. If your source file is called `usenixTemplate.tex` and your `bibtex` file is called `usenixTemplate.bib`, here's what you do:

```
latex usenixTemplate
bibtex usenixTemplate
latex usenixTemplate
latex usenixTemplate
```

4.4 Last SubSection

Well, it's getting boring isn't it. This is the last subsection before we wrap it up.

5 Results

6 Supplementary Results

A polite author always includes acknowledgments. Thank everyone, especially those who funded the work.

7 Discussion

It's great when this section says that MyWonderfulApp is free software, available via anonymous FTP from

```
ftp.site.dom/pub/myname/Wonderful
```

Also, it's even greater when you can write that information is also available on the Wonderful homepage at

```
http://www.site.dom/~myname/SWIG
```

Now we get serious and fill in those references. Remember you will have to run latex twice on the document in order to resolve those cite tags you met earlier. This is where they get resolved. We've preserved some real ones in addition to the template-speak. After the bibliography you are DONE.

8 References Cited

9 Author Contributions