Udacity

Machine Learning Engineer Nanodegree

Capstone Proposal

Catherine Sai
March 23rd, 2020

Proposal: Predicting Retailer Store Sales

The chosen project which is described below is based on this Kaggle competition:

https://www.kaggle.com/c/rossmann-store-sales/overview

Domain Background

The domain for this use case are multinational retail companies with thousands of stores. The data for the project is from “Rossmann”, a drug store company in Europe. My motivation to tackle this problem is that I will start a Data Scientist position at a textile retailer in May 2020 and think that this kind of use case is likely to be relevant for my future employer. Also, I think this use case is relevant for many companies in general – all companies selling a product (not just drug stores and clothing retailers, but any B2C business like car manufacturers, coffee shops and restaurants predicting demand). With a slightly different input data, parts of this code can probably also be used for similar use cases in B2B companies.

Problem Statement

Each retail store has to predict their demand ahead of customer purchase in order to get the products from the central distribution center of the company. In some domains the prediction period can be one or two weeks (I would assume this for e.g. supermarkets). For other industries like clothing companies it’s often a couple month so the head office can plan production accordingly as products change a lot over the year.

In this case the managers of the stores have to predict the demand for each product in their drug store 6 weeks in advance. The influential factors on store sales are complex and thus it seems a good data science task to analyze the available data in order to predict which features have the highest impact on sales prediction and how well a data science model could predict sales. Another reason to tackle this issue with

a data science model is, to generalize the prediction – calculating it with the same model instead of having each manager make their individual assumptions.

The target value “sales” per day are a continuous number, thus it is a regression problem with the goal to predict a scalar value.

Datasets and Inputs

The Kaggle competition includes the historical data of 1,115 Rossmann stores. Following is a list of the csv files provided as well as the columns within these files: (Resource: https://www.kaggle.com/c/rossmann-store-sales/data)

Files

- train.csv - historical data including Sales
- store.csv - supplemental information about the stores

Data fields

Most of the fields are self-explanatory. The following are descriptions for those that aren't:

- Store - a unique Id for each store
- Sales - the turnover for any given day (this is what I am predicting)
- Customers - the number of customers on a given day (this feature is not available for prediction)
- 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

At a first glance the provided features seem beneficial as store sales can be influenced by many factors. I am sure there are other features which are not included in the datasets that would also be beneficial. But the provided features give a good first insight in different areas of influence to sales (like seasonality, promotions, competition, etc.).

The train.csv will be split into a train- and a test-set. The original test.csv of the competition will not be used as it gives me no option to evaluate the model (as I am not going to turn it in for the already closed competition). The train.csv contains data from 01.01.2013 until 31.07.2015. To only use whole weeks, I will use the data from Monday 07.01.2013 until (including) Sunday 26.07.2015. This are 133 weeks of data. To be consequent with the original challenge, I am going to predict 7 weeks.

Thus my

train range will be 07.01.2013 – 07.06.2015 (94,7% of data)

and my

test range will be 08.06.2015-26.07.2015 (5,3% of data).

Both this train part of the train.csv and the store.csv will be used as input data for the model.

Solution Statement

This data science project should lead to a model that can forecast the "Sales" of every from the 1115 stores for 7 weeks. A hold-out period of 7 weeks will be separated from the data as test data to measure the prediction against. The sales prediction can clearly be measured against the actual sales and thus gives a quantifiable solution. There will only be one model for all stores, not a very overfitted model for each store – this way the replicability of the solution is given. Also as I will explain in the section “evaluation metric”, the results of the tested models will be compared using the Root Mean Square Percentage Error and thus making it clearly measurable which model delivers the best prediction.

Benchmark Model

As a benchmark for this project I am going to take a simple exponential smoothing model which will not take the features into consideration but only have past dates and sales per store as input. The simple exponential smoothing is a basic algorithm for time series forecasting and relatively easy to implement. Depending on how the alpha parameter is set, it can be equal to just the mean of the past. Therefore this will be a great baseline for further development. If another models prediction is not better than a simple exponential smoothing it is definitely not worth the effort. Through the RMSE and RMSPE the results of this Benchmark will be directly comparable with the other tested model.

Evaluation Metrics

In the Kaggle competition, submissions are evaluated on the Root Mean Square Percentage Error (RMSPE). Therefore this will also be considered for this projects. In addition to the RMSPE, the slightly simpler RMSE will also be calculated. The RMSPE is calculated as:

$$\text{RMSPE}= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{r-p}{r}\right)^2}$$

The variables have the following meaning:

- r represents the real sales of one store on one day
- p represents the predicted sales of the same store
- n represents the number of times the variable is forecasted

If a store has zero sales on a day, this day doesn't influence the calculation of the score.

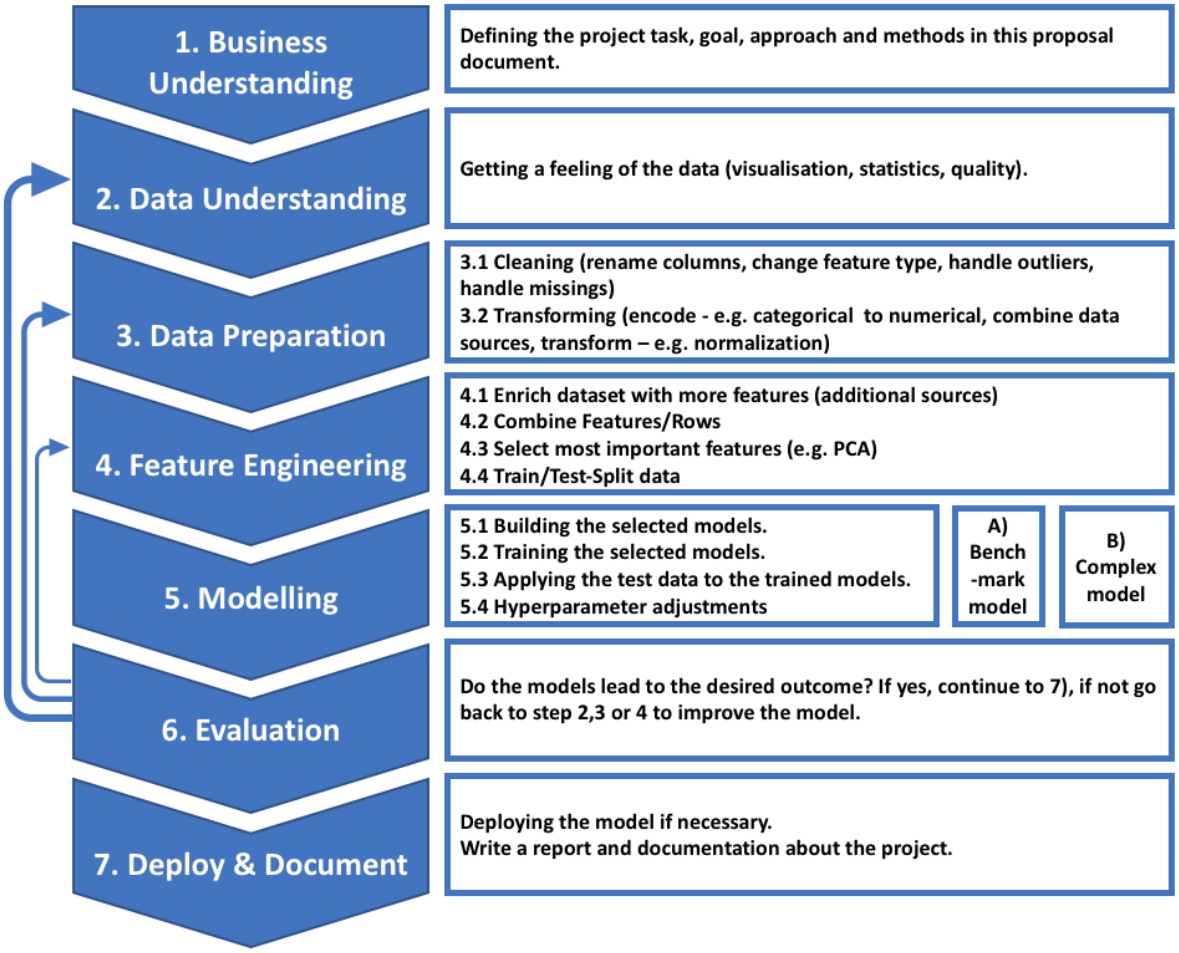
The RMSPE basically calculates the difference between the predictions and the target, squares and builds the average. The percentage part of the equation simply means that the absolute error is divided by the target. This way, the error is presented in relative terms of predictions instead of absolute numbers (e.g. otherwise 9 out of 10 correct predictions would be equally good to 99 out of 100 correct predictions). Finally, the square root adjusts the scale of the errors to be the same as the scale of the target.

For this metric, an optimal model would get the value 0. A negative value is not possible due to the squares of the individual errors. The higher the value the worse the predictions.

(Source: Kaggle Competition Site as stated above and https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-2-regression-metrics-d4a1a9ba3d74)

Project Design

For the outline of the project I plan to work along the CRISP-DM approach. You can view my planned workflow with included task overview in the picture below (own graphic).



For 3.1, “handle missing’s” it’s to be noted that this means single missing values in columns as well as missing rows. The latter is especially important in timeseries data as I deal with in this use case. If a day, week or year of data for one store is missing we won’t see missing fields in the csv file because in this case the whole row of data is missing. Therefore, the dates of all stores should be checked for consistency and missing dates – I then have decided if possible missings should be added with a sale of zero or if they should be excluded. As stated above, if a store has zero sales on a day, this day doesn’t influence the calculation of the RMSPE score. But it can still be important for the training of a model to have consistent time series data - even if the store had a sale of zero this might be important input information.

For 4.2. I thought of e.g. trying not only prediction on day basis but also week aggregations of the input and test-data. Purchase usually doesn’t happen on a day basis and if orders are only made once per week it is also enough to predict the demand per store per week. This is much easier than predicting the demand per day.

For 4.3. I plan to perform an Amazon SageMaker Principal Component Analysis Algorithm (https://docs.aws.amazon.com/sagemaker/latest/dg/pca.html)

For 5.1 I plan to use the an Amazon SageMaker DeepAR forecasting algorithm (https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html) as I have a quite large data set with many features to take into consideration and this RNN supposedly copes well with this kind of use case.