Capstone Proposal

by Konstantin Rink
V1.2

The project's domain background

The domain for this capstone project proposal is the retail industry in Germany. The retail industry is facing many challenges nowadays. A high number of competitors, frequently changing products and low margins (compared to other industries like automotive). Decision-makers need to plan their campaigns, pricing strategies, and budgets to compete with other competitors.

Data Science or Machine Learning can support them perfectly. As an example, Machine Learning models can not only provide sales forecastings for better future budget planning but also provide an evaluation of the past marketing campaigns.

This capstone project will focus on sales forecasting for Germany's second-largest drug store chain Rossmann. Rossmann started four years ago a challenge on Kaggle[1] to predict sales figures for each of their 1.115 stores. The motivation for this challenge was that store managers are asked to predict the sales for the next 6 weeks.

Problem statement

By the structure of the data (each id represents a store on a respective day) the original problem can be easily solved by just predicting sales data for the next data point. Therefore, many submitted notebooks used XGBoost or RandomForest models. Over 769 notebooks with solutions and visualizations have been submitted so far. **However, only(!) 17 of them use time series models and 0(!) of them use time series models with external regressors.**

Therefore the idea of this capstone project is to use the data in a different way and predict the sales for the next 90 days. Based on the later performance of the model the 90 days predictions will be in total and/or for each of the 1.115 stores.

Data inputs

- Data set from 2016 contains data for 1.115 stores.
- Data available from 2013-01-01 to 2015-07-31 (1.017.209 rows).

Explanation (see Kaggle link):

Files

- **train.csv** - historical data including Sales
- **test.csv** - historical data excluding Sales - **PROBABLY NOT USED**
- **sample_submission.csv** - a sample submission file in the correct format - **NOT USED**
- **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.

- **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

Train.csv

In [45]:

```
pd.set_option('float_format', '{:f}'.format)
train = pd.read_csv("rossmann-store-sales/train.csv",low_memory=False)
train.head(3)
```

Out[45]:

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	1	5	2015-07-31	5263	555	1	1	0	1
1	2	5	2015-07-31	6064	625	1	1	0	1
2	3	5	2015-07-31	8314	821	1	1	0	1

In [46]:

```
train.describe()
```

Out[46]:

	Store	DayOfWeek	Sales	Customers	Open	
count	1017209.000000	1017209.000000	1017209.000000	1017209.000000	1017209.000000	101
mean	558.429727	3.998341	5773.818972	633.145946	0.830107	
std	321.908651	1.997391	3849.926175	464.411734	0.375539	
min	1.000000	1.000000	0.000000	0.000000	0.000000	
25%	280.000000	2.000000	3727.000000	405.000000	1.000000	
50%	558.000000	4.000000	5744.000000	609.000000	1.000000	
75%	838.000000	6.000000	7856.000000	837.000000	1.000000	
max	1115.000000	7.000000	41551.000000	7388.000000	1.000000	

In [52]:

```
print("Min date: "+train.Date.min())
print("Max date: "+train.Date.max())
```

```
Min date: 2013-01-01
Max date: 2015-07-31
```

Store.csv

In [54]:

```
store = pd.read_csv("rossmann-store-sales/store.csv",low_memory=False)
store.head(3)
```

Out[54]:

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	Competit
0	1	c	a	1270.000000	9.000000	
1	2	a	a	570.000000	11.000000	
2	3	a	a	14130.000000	12.000000	

In [55]:

```
store.describe()
```

Out[55]:

	Store	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSince
count	1115.000000	1112.000000	761.000000	761.000
mean	558.000000	5404.901079	7.224704	2008.668
std	322.017080	7663.174720	3.212348	6.195
min	1.000000	20.000000	1.000000	1900.000
25%	279.500000	717.500000	4.000000	2006.000
50%	558.000000	2325.000000	8.000000	2010.000
75%	836.500000	6882.500000	10.000000	2013.000
max	1115.000000	75860.000000	12.000000	2015.000

Solution statement

Therefore the idea of this capstone project is to use the data in a different way and predict the sales for the next 90 days. Based on the later performance of the model the 90 days predictions will be in total and/or for each of the 1.115 stores. Several Machine Learning models like LSTM, FB Prohpet but also ARMIAX will be used.

Benchmark model

Two options for benchmark models are given. First, a simple time seires (ARIMA) model that uses only one predictor/regressor, the sales past values over time. Second, a Naïve 2 (see set of evaluation matrics). Both benchmark models are used to see if future models like SARIMAX, VAR, LSTM, FB PROPHET with more than one predictor/regressor perform better.

Set of evaluation matrics

According to Makridakis, Spiliotis, Assimakopoulos (2020)[2] sMAPE and MASE as well as OWA will be used. In addition to that RMSE will be used as well.

Classical evaluation matrics

RMSE = $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}$

MASE = $\frac{\sum_{t=n+1}^{n+h}|Y_t-\widehat{Y_t}|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t-Y_{t-m}|}$

sMAPE=$\frac{2}{h}\sum_{t=n+1}^{n+h}\frac{|Y_t-\widehat{Y_t}|}{|Y_t|+|\widehat{Y_t}|}*100(\%)$

OWA (overall weighted average)

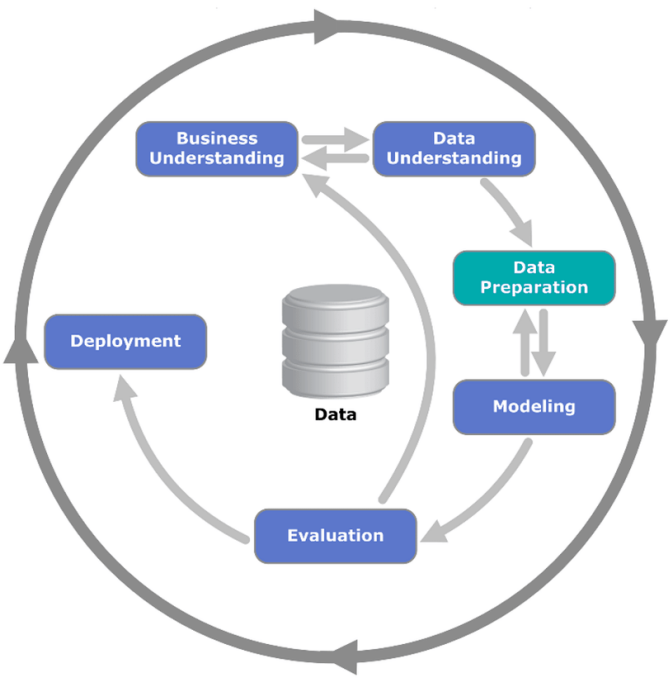
1. Divide all Errors by that of Naïve 2 (random walk) to obtain the Relative MASE and the Relative sMAPE.
2. Compute the OWA by averaging the Relative MASE and the Relative sMAPE
   For more information please see[3]

Outline of the project design

In [63]:

```
from IPython.display import Image
from IPython.display import display
Image(filename='img/crisp_dm.png',width=350)
```

Out[63]:



The project design follows the CRISP DM framework:

1. The first two steps are about Business and Data Understanding which are already covered in this proposal.
2. The third step is about data prepareation like cleaning the data and bringing the data into the right format for the next steps.
3. The 4th step is about modeling. In this step several models will be created.
4. In the Evaluation step the models will be evaluated based on their performances and their performance compared to the baseline model. Then the best model for (each store) will be selected.
5. The deployment step is in this case to present the final results. As an optional bonus an AWS endpoint can be created to offer end users (store managers) a dashboard where they can run forecasts for n days.

References

[1] https://www.kaggle.com/c/rossmann-store-sales/data (https://www.kaggle.com/c/rossmann-store-sales/data)
[2] https://www.sciencedirect.com/science/article/pii/S0169207019301128 (https://www.sciencedirect.com/science/article/pii/S0169207019301128)
[3] https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf (https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf)