WS 2016 Project 1

Martin Simon Haugaard
cdl966@alumni.ku.dk

1. INTRODUCTION

The scope of this project it to try and create a prediction model which, based on Google term history, aims to predict the real-life sales of vaccines. The terms are extracted by short layman descriptions from two danish websites `sundhed.dk` and `ssi.dk`, which contain various information regarding public health. The terms extracted from these two websites were then parsed through Google Trends[1] resulting in relative search history regarding each term, this allowed for constructing a model in order to predict search trends for words regarding each vaccine, which may prove useful in real world sales predictions.

The following project was all done in python, using premade modules to assist in the mining and processing of trends.

In this project, the vaccines **PVC** & **HPV** were examined. Any two vaccines should be fine, but I'm happy with these two, as their data differs in size, with PVC only creating 16 query items, while HPV has 65. Query items are also called Trends, and are described in the following section.

2. METHODOLOGY

This section contains brief descriptions of the key areas of the project.

2.1 Trends

In this project, trends are words which are commonly used when describing vaccines. Stopwords, which are words that only appear in text for grammatically reasons, do not count as trends. The trends are determined by reading multiple (in this case two) descriptions of a given vaccine, and then stripping each description of symbols and stopwords. Both descriptions are then compared to each other, and only words which occur in both descriptions qualify to be trends. This is a rather simple way of creating the trends, and some words, which only differ grammatically, such as *alvorlig* and *alvorlige* are considered different, even through they are functionally the same. However it proves sufficient for the scope of this project.

2.2 Mining of Trends

In order to mine the trends from Google, I resorted to using a python package called pytrends[2], which provides for a simple framework for downloading .csv files from Google Trends. It works by signing into Google, and then, with different intervals, making a request after which it allows for download of the corresponding .csv file.

When downloading the .csv files, they can come with data in either a weekly or a monthly interval, most likely due to the popularity of certain terms, which prompts for more indept information. In order to accommodate for the difference of format, a simple python script was written, which would convert weekly data into monthly data, by making a crude estimation of a monthly value, based on average values and full week durations, which meant some weeks overlapped into the following month. A more advanced script should probably be written for a more precise estimation of monthly values, but for this project, I found the crude script to work well.

Certain trends may result in empty .csv files. For example, for PVC, *vaccinen*, *pneumokoksygdom* & *pneumokokvaccinen* came up empty, thus reducing the usable number of trends from 16 to 13. In these cases I simply skip the trend when doing further analysis.

The data values given in each .csv file will be a value in the interval [0, 100], being relative measurements of the popularity of the trend for a given period of time, 100 being the most popular interval during the measured time scope. This means that a trend scoring highly doesn't necessarily mean it was a highly searched term on Google overall, but rather, it's relatively highly searched compared for other time intervals regarding the same search term.

Lastly, it's worth noting that the trends mined were only for the Danish region of Google Trends, as the main scope for this project is the danish market.

2.3 Prediction

The choice of prediction method was randomly chosen to be that of Lasso. Lasso requires a Matrix, X, of dimensions NxM where N is the time frame we examine, here 57 months, and M is the number of trends (13 and 61 after empty results have been removed). The provided data served as a ground truth when preforming fitting of the model, and even

[1] `https://www.google.dk/trends/`

[2] `https://github.com/GeneralMills/pytrends`

Web Science 2016 DIKU, Denmark

through it holds data for 60 months, I only use the first 57, as to match the X Matrix.

Fitting of the model was done as part of a 5-fold cross-validation. The python package *sklearn*[3] provided all the necessary tools for doing the splitting needed in the 5-fold cross-validation, as well as the model fitting and prediction.

```
k_fold = cross_validation.KFold(len(X), 5)
```

would provide a *k_fold* consisting on five iterations on form $k, (train, test)$, where train is list of index values for training, and test is a smaller list for testing.

These values would allow me to gradually train the module, results of which are in the next section.

3. FINDINGS

First I examined PVC. Figure 1 shows how my predictions for each of the five folds went. Looking at these 5 folds, the predictions being made, in blue, does generally follow the provided data, in red, however not every fold is as spot on as fold 5. In figure 2 the prediction are is spot on Fold 2, as it predicts the sudden drop at 6 months, and then it practically follows the curve from there.

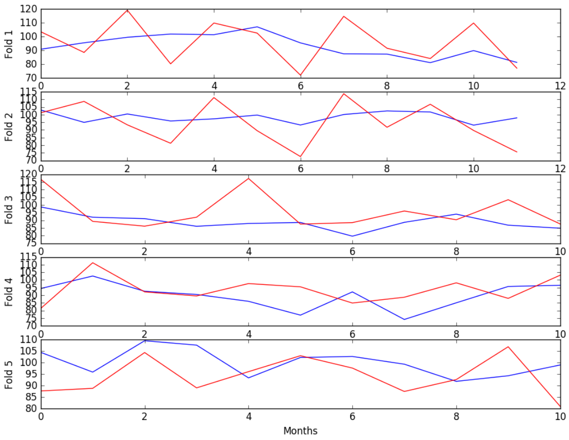

Figure 1: PVC predictions done in 5-fold cross-validation, red line is ground truth, blue is prediction.

Other than simply looking at graphs, the way to determine how well a prediction model is doing, is by calculating the *Root-mean-square error*, or RMSE $= \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$, with n being the number of folds and e_t being the difference, or error, between the prediction and the real-life observed data for the corresponding fold. RMSE is useful for evaluating the accuracy of a model, as a low value indicates a very too correlation, as the error is generally low, and a higher value means a bigger average error.

In table 1 I've provided the average RMSE over all five loops, for each vaccine, when looking at each of the provided clinical data files. Next to my results, I've entered the results which the original report[1], which this project was based on, was able to achieve.

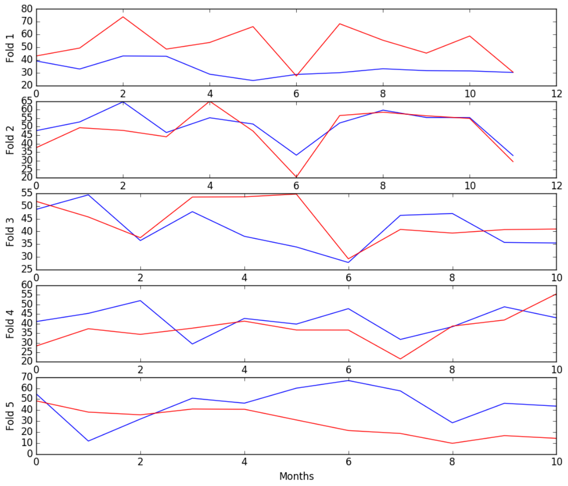

Figure 2: HPV predictions done in 5-fold cross-validation, red line is ground truth, blue is prediction.

For RMSE, the bigger the value you get, the worse your prediction probably is, and it's clear from the table, that my predictions were strictly worse than those provided in the original report, leading to the conclusion that my model is not quite as good.

Table 1: Lasso RMSE comparison, with Prediction Vaccination Uptake using Web Search Queries report.

	Original Report	My Findings
HPV-1	12.701	17.07
HPV-2	18.423	23.202
HPV-3	23.074	24.436
PCV-1	7.845	12.685
PCV-2	9.770	14.448
PCV-3	10.368	17.367

4. CONCLUSIONS

Examining table 1 it's clear that my predictions did not preform as well as those in the original report. Especially not for HPV. I find the HPV to be interesting through, as the clinical data for the last 11 months are drastically lower than the previous 49 months, as plotted in figure 3. The model tries to predict a curve, as close to the previous 49 months, but since there's a sudden change, it fails to predict the vaccine frequency.

As for PVC, figure 4, the prediction is hard to compare to the clinical data. For each spike it predicts, it fair to say it misses one as well. This may be due to the, compared to the HPV vaccine, there's a limited amount of trends, 13 versus 61, and while the RMSE is not too far off, compared to the original paper, the actual usefulness of the current state of the predictions are very risky, should one try to invest based on these data point.

Reason may be that the terms which were mined using

[3] http://scikit-learn.org/

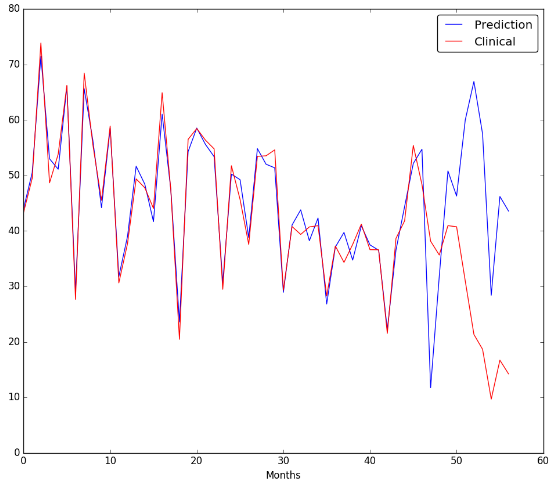

Figure 3: HPV Model predicted over full duration of data sample. red line is ground truth, blue is prediction.

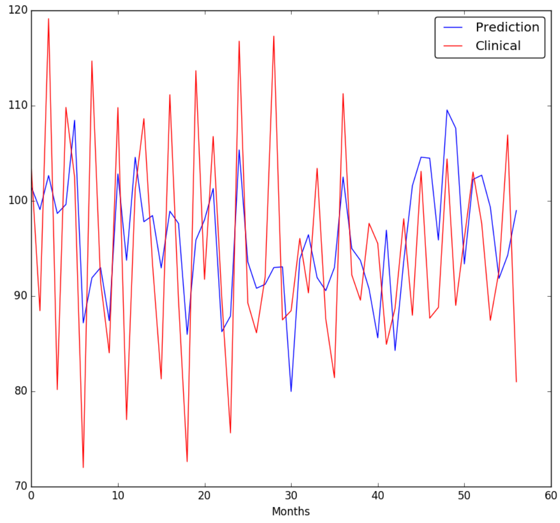

Figure 4: PVC Model predicted over full duration of data sample. red line is ground truth, blue is prediction.

Google all have equal influence on the prediction model, as they are all relative measurements, which gives no indication on the most important terms. Some terms may be irrelevant or misleading, as they can be too general for this specific area of interest.

Overall, the exercise was engaging and a learning experience. But it's hard to be sure of the exact usefulness of the models as is.

5. REFERENCES

[1] N. D. Hansen, C. Lioma, and K. Molbak. Predicting vaccination uptake using web search queries. *in press*, 2016.