# Machine Learning in Practice

Marc de Groot, Marta Parada Segui, Patrick Tan, Suzanne van den Bosch, Zhuoran Liu

## 1 Problem description

We are working on Home Depot Product Search Relevance. Our goal is to accurately predict the relevance of a search result on the home depot website. For background information we would refer to https://www.kaggle.com/c/home-depot-product-search-relevance.

## 2 Approach

We have tried several approaches at once. Some more fruitful than others.

### 2.1 Levensthein Distance

This approach checked if the product listed in the test data occurs in the training data and if that was the case it would check the Levensthein distance between the query used in the test data and those in the training data. If a match was found then the relevance listed in the training data would be used. This approach focuses on using the human knowledge that the training data contains.

### 2.2 Support Vector Machines

This approach construct a hyper-plane do the regression. Since the method SVR has the advantages that it is effective in high dimensional spaces, memory efficient and different kernels can be used. Also only when feature greater than samples give poor performance. So in this HomeDepot case it works. We have much more dimensions of samples than the features, even we use the vector space method. The SVR constructs the equation of hyper-plane to do the regression. Here it is efficient, because support vectors set is just a small part of the whole samples.

### 2.3 Random Forests

## 3 Results

The results of the different approaches varies. Not everything has been completed at the time of writing.

### 3.1 Levensthein Distance

It turned out that exact matches were quite rare. Using Levensthein Distance did make this algorithm less vulnerable to typos since a low distance would still suggest a high probability that it was actually a match.

### 3.2 Support Vector Machines

Firstly, we tried to use the method vector space and convert the training set into a vector space with every dimension a specific word. But when the total amount rose to 1000 samples, we have 256 dimensions already. So it is not reasonable. Then we used the features as same as the Random Forest Trees and it worked. Due to the long running time, we tried firstly the SVR with rbf kernel. The result seems like a little bit overfitting. It got a result 0.53856 in Kaggle submission. Then we used different kernels(polynomial and linear) and tried different parameters(gamma, degree and so on) to tune the model well. When writing the report, best score is 0.5386.

### 3.3 Random Forests

## 4 Discussion

We are currently trying to refine the algorithms that work and see if we can combine the results in a way that they will be better than when we only use a single algorithm.

### 4.1 Things that went well this competion

- Good group atmosphere
- Regular meetings
- Fair distribution of tasks

### 4.2 Things we can improve on for the next competition

- Perhaps planning to do more work in the beginning stages so that the last few weeks will be less stressful. This was due to our scheduled exams and other courses though.
- Sharing our work more regularely. Not sharing unfinished work made it harder for us to help each other.

## 5 Individual contributions

### 5.1 Marc

-

-

## 5.2 Marta

-
-

## 5.3 Patrick

- Communication with the teachers and coach
- Reserve meeting areas
- Program the algorithm using Levensthein
- Create the group account used for submissions
- Trying to assist Liu with SVMs
- Write part of this report

## 5.4 Suzanne

-
-

## 5.5 Zhuoran

- Find and share materials and books about SVM method, vector space and python plot.
- Summarize idea and share it in Github.
- Made the SVR method work and tuned it.
- Write part of this report.
-
-