|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Title:** | Data Mining Group Project | | | |
| **Programme/Session:** | MSc (Data Science) / 2015-16 | | **Subject:** | COMP6237 Data Mining |
| **Institute:** | ECS, University of Southampton | | | |
| **Project:** | Predict the relevance of search results on homedepot.com | | | |
| **Project Reference External:** | www.kaggle.com | | | |
| **Project Reference Internal:** | Coursework 1: Predictive Data Mining Group Project | | | |
| **Project internal Timelines:** | **Start Date** | 29-Jan-16 | **End Date** | 05-May-16 |
| **Project Supervisors:** | Dr. Jonathon S Hare & Dr. Markus Brede | | | |
| **Team Name:** | Phi1337 | | | |
| **Team Members:** | **First Name** | | **Last Name** | **Email** |
| Αdisorn | | Jaroensathayatham | aj7g15@soton.ac.uk |
| Antonios | | Andronis | aa3e15@soton.ac.uk |
| Emmanouil | | Petsanis | ep1g15@soton.ac.uk |
| Napoleon | | Koskinas | nk5g15@soton.ac.uk |
| Nikolaos | | Perrakis | np4g15@soton.ac.uk |
| Syed M. Ali | | Shah | smas1c15@soton.ac.uk |
| **Github Link:** | https://github.com/lolaum/Phi1337 | | | |

**DRAFT REPORT**

**(Please change straightaway if you think that the description is not correct)**

**To be copied to TEXMAKER and final draft.**

1. **Abstract**

To be written in the end.

1. **Project Background**

Businesses are always working to maximize the profit and exceed the customer services expectations. Maximization of profit is directly proportional to the increase in both offline and online product sales.

Online retailers use complex algorithms to ensure that the relevance of customer searches are as accurate as possible. If the customer finds the product they need, it is highly likely that the customer will commit the purchase. If the product searched by the potential customers is not in business product line then online retailers use recommender system algorithm to suggest the true alternate or similar product.

This scientific approach to the online business not only help customers to get the right product of their choice but also increase the profit and revenue figure to the company.

This paper focuses on our research to evaluate all possible option of data pre-processing and modelling to improve the search relevance i.e., to improve the chances that the search results matches the search query.

1. **Literature Review**

Two paper found

1. **Methodology: TO BE UPDATED WITH EACH MODEL USED**

The client has provided the search queries and the title of product that was returned for the query. The development process will start by understanding the aim and relevancy of the search query. Each query will be compared to the resulting product displayed. We will try to understand the intention of making the query and will document what the potential customer was looking for and what product he got as search result. This comparison will be classified as:

* 1 for irrelevant search result i.e., the results are not at all related by any mean.
* 2 for partially relevant search result i.e., the result generally matches the query but may differ in brand, size, colour or specifications
* 3 for relevant search result i.e., the product is exactly what the customer is looking for

Both train and test datasets provided are well normalized.

1. **Data Cleansing/Pre-processing**

After review and analysis of different approaches to data cleansing and text processing. We developed our hypothesis based on the approach used in ‘CrowdFlower winning solutions’ [Reference 1]. Here is a detailed description of our data pre-processing approach:

1. Misspelled words: As the search queries has spelling mistakes including typographical errors, we used the approach followed by ‘Stebk’ [Reference 2] for removing the spelling checks. Steubk created a simple function that uses Google to remove typos. We also did some manual spelling corrections. We also used ‘difflib.SequenceMatcher()’ [Reference 3] to find the similarities between misspelled word and correction made.
2. Stopwords: In order to create the feature vector, we have removed the words with no meaning by using the simple python library function ‘stopwords’.
3. Special Characters: The special characters were also removed by simple function.
4. Character cases (Upper case/Lower case): All upper cases were made lower.
5. Stemming:
6. Tagging Words: We used ‘NLTK.pos\_tagger()’ [Reference 4] function to separate nouns, verbs, adjectives and adverbs. This function is also used to find the most relevant words in query and title.
7. Term Frequency-Inverse Document Frequency (TF-IDF): In this project, not only words but the number of letter in the words are also important. The sum of weight in words was was calculated using the feature weights X number of letters in word. The important point to note is that we used stemmed words in TFIDF.
8. Dimensionality Reduction ---???
9. EXPLORE: Jaccard Similarity Coefficients <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_similarity_score.html>
10. EXPLORE: Damerau-Levenshtein <https://pypi.python.org/pypi/pyxDamerauLevenshtein>
11. **Models**
12. Xgboost:
13. Gradient Boosting Machine (GBM):
14. Random Forest:
15. Support Vector Regression:
16. ExtraTrees:
17. **Score from Model used**
18. **Conclusion**
19. **Reference**
20. <http://blog.kaggle.com/2015/07/22/crowdflower-winners-interview-3rd-place-team-quartet/>
21. <https://www.kaggle.com/steubk/home-depot-product-search-relevance/fixing-typos/comments>
22. <http://stackoverflow.com/questions/4802137/how-to-use-sequencematcher-to-find-similarity-between-two-strings>
23. <http://www.nltk.org/book/ch05.html>