Management Science and Information Systems Studies

Final Year Project Report

Data Mining of Amazon Product Information for Clavis Insight

Simon Major

March 2015
I declare that the work described in this dissertation has been carried out in full compliance with the ethical research requirements of the School of Computer Science and Statistics.

Signed: ___________________

Simon Major
31/03/2015
ABSTRACT

The objective of this project was primarily to investigate relationships between page content and search rankings on online marketplaces, foremost on Amazon.com, although any insights into the nature of search rankings would be of value.

The findings of this project’s analysis showed that there may be some relationships between page content and search rankings, but that there is some external factor which “muddies” examination of these relationships, making hard conclusions very difficult to draw.
PREFACE

Clavis Insight provide a Software-As-A-Service platform to its clients in the FMCG (Fast Moving Consumer Goods) retail space on online marketplaces such as Target.com and Amazon.com. Their core product provides daily dashboards and detailed reports to their customers. These reports are designed to enable their customers to optimise their product distribution, content integrity and placement in the digital channel in order to protect their brand online, and grow their sales.

As very little is known about online marketplaces, especially with regards search ranking, Clavis are interested in investigating the relationship between page content and search rankings. They hypothesise that being able to boost search rankings should lead to greater sales for their customers.

This project aimed to examine primarily text-based data, attempting to determine the existence of a relationship between search terms, page content, and search ranking. In addition to this, an attempt was made to determine good smoothing periods for extremely volatile data.

I would like to thank Clare Conway and Stephen Gormley of Clavis Insight for their assistance and encouragement throughout the project.

I would like to thank Aidan O’Neill and Laura O’Malley for their great help in the production of this report, and I would like to thank my father – for listening to my frustrations, for being patient with my overexcitement, for his guidance and for pushing me to take on a challenge.

Finally, I would like to thank my project supervisor, Arthur White for his great support and guidance over the course of the project, along with Aideen Keaney and Myra O’Regan of the School of Computer Science and Statistics for their assistance and care.
CLAVIS INSIGHT
Data Mining of Amazon Product Information

March 2015

TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>NO.</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Client</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Project Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Terms of Reference</td>
<td>2</td>
</tr>
<tr>
<td>1.4</td>
<td>Summary</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Conclusions</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Recommendations and Future Research</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>DATA DESCRIPTION AND METHODOLOGY</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Problem Definition and Background Research</td>
<td>5</td>
</tr>
<tr>
<td>3.2</td>
<td>Data Description</td>
<td>7</td>
</tr>
<tr>
<td>3.3</td>
<td>Data Handling</td>
<td>9</td>
</tr>
<tr>
<td>4.</td>
<td>ANALYSIS</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>Initial Analysis</td>
<td>12</td>
</tr>
<tr>
<td>4.2</td>
<td>Text Summary Analysis</td>
<td>13</td>
</tr>
<tr>
<td>4.3</td>
<td>Stable Data and Modelling</td>
<td>16</td>
</tr>
<tr>
<td>4.4</td>
<td>Volatile Data</td>
<td>21</td>
</tr>
</tbody>
</table>
## APPENDICES

<table>
<thead>
<tr>
<th>NO.</th>
<th>CONTENT</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>Original Project Outline</td>
<td>A.1</td>
</tr>
<tr>
<td>B.</td>
<td>Interim Project Report</td>
<td>B.1</td>
</tr>
<tr>
<td>C.</td>
<td>RuleFit (Excluding SKU Number) Top 15 Rules</td>
<td>C.1</td>
</tr>
<tr>
<td>D.</td>
<td>Linear Regression Model Outputs</td>
<td>D.1</td>
</tr>
</tbody>
</table>

REFERENCES
1. INTRODUCTION AND SUMMARY

This chapter describes the client, project background, terms of reference and a summary of the further chapters of the report.

1.1 Client

Clavis Insight works in e-commerce analytics, with clients primarily in the FMCG (Fast-Moving Consumer Goods; also CPG – Consumer Packaged Goods) retail space – these are products that are sold quickly and at relatively low cost, e.g. nappies, sanitary products, bottled soft drinks, etc. Clavis’ clients sell their products on a number of e-commerce marketplaces, notably Amazon.com and Target.com - Amazon being the biggest, with an estimated 240 million SKUs on sale in 2014 (Rivas, 2005).

1.2 Project Background

Online marketplaces are unlike traditional marketplaces in that the science behind sales generation is not yet fully understood. Market research has been carried out since at least the 1930s (Nielsen, 2014) to investigate how and where a product should be displayed in order to maximise revenue in traditional marketplaces (e.g. supermarkets). In contrast, online marketplaces have existed for less than twenty years, only gaining prominence in the last decade or so. As such, there is much that is unknown about the marketing of a product in these online marketplaces. However, online marketplaces are not necessarily unbiased; any algorithm used to determine product search rankings has been written by a human, and so may rank products according to factors desirable to the marketplace (or its proprietor), rather than the end consumer.

It is believed that Clavis’ clients (including 9 of the top 10 FMCG manufacturers) would benefit if there was some better understanding of how these algorithms rank products, and how to ensure that their products rank as highly as possible in order to maximise visibility.
1.3 Terms of Reference

The main aims of this report are to:

- Analyse data from product pages on Amazon.com and Target.com to investigate relationships between page content and search ranking;
- Investigate whether a link exists between product stock levels and search ranking; and
- Identify a method of smoothing highly volatile data to determine a good estimate of a product’s overall search ranking performance.

1.4 Summary of Remaining Chapters

The chapters of this report are outlined as follows:

- **Chapter 2** discusses the key conclusions and future research recommendations from analyses;
- **Chapter 3** outlines the data used for analysis, issues and solutions to handling and transformation of the data, and methods undertaken prior to analysis; and
- **Chapter 4** describes the analyses performed on the data, and discusses some of the results.
2. CONCLUSIONS AND RECOMMENDATIONS

This chapter summarises the main conclusions and recommendations found in this report.

2.1 Conclusions

It is difficult to provide firm, actionable methods to improve search rankings based solely on product page content. While analysis has pointed to some relationships which are believed to contribute to search rankings, it is difficult to be very confident of these results. From exploration of the data, along with client hypothesis and general conjecture, it appears that some factor other than product page content is the primary metric for determining search ranking.

Also, due to the volatility of search rankings on both marketplaces, the models are fit to median values of search rank - this indicates a rough estimate of where a product will rank over time, rather than a more granular day-to-day estimate.

Phase 1 Data and Target.com

The analysis suggests that Target.com search rankings benefit from an increased number of matches of search terms in the product title and perhaps suffer from having an over-long description. However, there is not enough agreement between model types to definitively state this without further investigation into external factors. Without better understanding of these external factors, it is unwise to treat any of the analysis results as gospel, although experiments could be constructed to test the hypotheses of these analyses, particularly to determine the effect of search term occurrences in product title and description.

Phase 2 Data and Amazon.com Volatility

Due to the relatively small timeframe, and the complex nature of the Amazon.com ranking mechanism, it has been difficult to assess factors that contribute to the volatility inherent in the Amazon.com datasets. However, preliminary results from model-fitting over different time periods have shown that the median search ranking over 4 weeks are reasonably predictable (at least as predictable as the Phase 1 dataset, using basic linear regression), with broadly similar R-squared values but higher RMSE values. The higher RMSE values are expected, however, as the data are so wildly volatile that prediction errors are going to be somewhat inflated.
Results from the best-fitting linear regression model over the 4-week median do show significant results with regards keywords and description search term matching, indicating that more matches in these fields lead to better search rankings. Again though, these results must be used with caution.

2.2 Recommendations and Future Research

In datasets from both marketplaces, there seems to be large influence on search rankings that is external to page content data. This has been hypothesised to be based on a combination of sales performance and conversion rate of products - it would appear that Amazon.com and Target.com both weight their search rankings in order to maximise revenue per customer, rather than on a pure text-based search. In order to more accurately determine the existence of some relationship between page content and search ranking, this weighting factor should be examined further. It may be that conversion rate data will never be available; sales performance should be at least estimable, and its influence on search ranking should be examined further. Beyond this, periodic smoothing of volatile search rankings could be further assessed - if the external factor cannot be measured, perhaps it can be mitigated or accounted for. If so, and if enough of the external factor is accounted for, better analysis can be performed as to page content's relationship with search ranking.
3. DATA DESCRIPTION AND METHODOLOGY

This chapter defines and researches the problem; describes the data provided; and discusses the methods used to handle the data, coercing it to a workable form.

3.1 Problem Definition and Background Research

The problem is better understood through a simple example: a customer wants to buy nappies from the Amazon.com marketplace. From the homepage, this customer types “nappies” into the search box (see Figure 3.1.1), then clicks on the magnifying glass icon to initiate the search. From here, the page is directed to the search results as seen in Figure 3.1.2 - note the Huggies product appearing 7th in the list. Kimberly-Clark, manufacturers of the Huggies brand and client of Clavis, aim to have their product place as close to the top of this list in order to maximise visibility of their brand, hoping that this will boost their sales performance.

Amazon do not publicly disclose exactly what factors affect a product’s search ranking - unlike, say, Google (Google, 2015) - but these have been hypothesised by the client to fall under three broad categories of data:
● Sales performance - this measures how well a product sells;
● Conversion rate - this refers to how often a customer purchases a product after viewing the product page, including data such as Time on Page and Bounce Rate (this is the number of times a potential customer views a product page but then returns to the search query or to a related product etc.);
● Page data - this refers to the information available on a product’s page, including all the description data; pricing information; customer reviews; and stocking levels.

However, only some of this data is accessible to the client - Amazon do not share sales performance (it is assumed that suppliers have at least implicit access to this data from their own distribution centres), nor data relating to conversion rates with its suppliers. Thus, the only data available to this analysis is data scraped from the product pages on the Amazon.com and Target.com marketplaces, as described below.

Amazon do disclose some information about how their A9 ranking algorithm works:

“One of A9's tenets is that relevance is in the eye of the customer and we strive to get the best results for our users. Once we determine which items are good matches to the customer’s query, our ranking algorithms score them to present the most relevant results to the user” (Amazon, 2015).

This indicates that relevance is first checked for, then ranking is determined based on other factors - this could mean that text-based search/text indexing is used to determine relevancy for a search, and that ranking is determined based on factors such as conversion rate or sales performance (particularly sales rank within a category/sub-category).

As the marketplace is constantly developing, and is still very much a “black box” - Amazon are opaque about how the algorithm works - there is not much publicly available information about the algorithm. Some conjecture (Mitchell, 2015) does exist, and agrees with client hypothesis of sales performance/conversion rate being highly contributory factors to search ranking.

3.2 Data Description

Nature of the Datasets Provided

Datasets were provided from Amazon.com and Target.com product pages. Figures 3.2.1 and 3.2.2 show the standard layouts of these pages, and label some of the primary sections of data scraped for the datasets. A number of datasets were provided in two phases:
• Phase 1 - 2014/early 2015:
  ○ 7 sets of daily data from Amazon.com product pages for each day from 30/10/2014 to 5/11/2014. In raw form, these consisted of ~9,500 rows of 100 variables each;
  ○ 29 sets of daily data from Target.com product pages over the month of April 2014 (no data were available for the 6th of April). In raw form these consisted of ~350 rows of 74 variables each; and
  ○ 28 sets of daily data from Target.com product pages over the month of November 2014 (no data were available for the 8th or 9th of November). In raw form these consisted of ~1,400 rows of 100 variables each.

• Phase 2 - late February 2014:
  ○ 181 sets of daily data from Amazon.com product pages from 1/09/2014 to 27/2/2015 (some days were missing; some days were duplicated). In raw form these consisted of ~9,500 rows of 100 variables each.

These datasets contain daily scraped data from the product pages, as well as search terms used for the products and their associated rankings. However, in their raw forms, these datasets were unworkable in format, and required special handling to transform them into a more convenient form (as detailed below). The reason for this format being inconvenient/unworkable is that the language used for performing analysis (the R statistical language) is designed to build models where data are recorded with a single entry per row, rather than multiple data points per row.
Structure of the Data

Both the Amazon and Target datasets were broadly similar in variables measured, so similar handling and analyses could be performed on all sets. Some variables of interest from both datasets are described in Table 3.2.1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type of Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIN</td>
<td>String; primary key (Amazon.com)</td>
<td>This is a uniquely identifying product code for each product in the Amazon.com dataset.</td>
</tr>
<tr>
<td>SKU Number</td>
<td>String; primary key (Target.com)</td>
<td>This is the “most unique” identifying product code for each product in the Target.com dataset.</td>
</tr>
<tr>
<td>Search Terms</td>
<td>String (character-delimited array)</td>
<td>This is a character-delimited list of search terms a product is ranked under.</td>
</tr>
<tr>
<td>Search Terms Ranking</td>
<td>String (character-delimited array)</td>
<td>This is a character-delimited list of rankings for each of the (above) search terms.</td>
</tr>
<tr>
<td><strong>Product Description</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the title of the product page: i.e. the name of the product.</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Additional Description</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the “overview” description of a product in the case of Target.com; and the bullet-pointed list of short descriptives on an Amazon.com product page.</td>
</tr>
<tr>
<td><strong>Keywords</strong></td>
<td><strong>String (character-delimited array)</strong></td>
<td>This is a (character-delimited) list of 5 * 50-character text fields, not actually visible on a product page but found in the page source.</td>
</tr>
<tr>
<td><strong>Product Image</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the URL of the primary image for a product.</td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the primary category of a product.</td>
</tr>
<tr>
<td><strong>Sub-Category</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the sub-category of a product.</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td><strong>String</strong></td>
<td>This is a string representing the availability status of a product to buy online.</td>
</tr>
<tr>
<td><strong>Enhanced Content</strong></td>
<td><strong>String (Amazon.com)</strong></td>
<td>This is a string representing the manufacturer “blurb” placed towards the bottom of an Amazon.com product page.</td>
</tr>
<tr>
<td><strong>Brand</strong></td>
<td><strong>String (primarily Amazon.com)</strong></td>
<td>This is a string representing the brand associated with the product e.g. Braun, Neutrogena</td>
</tr>
<tr>
<td><strong>Secondary Image</strong></td>
<td><strong>String (character-delimited array)</strong></td>
<td>This is a character-delimited list of URLs to additional images used in a product page (i.e. images which are not the primary image).</td>
</tr>
<tr>
<td><strong>Number of Customer Reviews</strong></td>
<td><strong>Numeric; integer</strong></td>
<td>This is an integer representing the number of customer reviews a product has been given.</td>
</tr>
<tr>
<td><strong>Customer Review Rating</strong></td>
<td><strong>Numeric; integer</strong></td>
<td>This is a decimal (to 1 decimal point) of the average customer review score.</td>
</tr>
</tbody>
</table>

The structure of some of these data make analysis difficult – particularly the delimited lists of search terms and their rankings. These are key variables in the data: rankings are used as outcome variables, and later new variables will be added which also rely on having the search terms isolated. In order to overcome this issue, the data were split by search term/ranking pairs (detailed below; for an explanation see *A Less Abstract Example* in Section 3.3); this allowed for analysis of the ranking variable, especially with regard search term occurrences.

### 3.3 Data Handling

Analysis of the data required that the data be formatted with a row for each unique product-search term/ranking pair, thus providing a daily search performance metric and target variable
for analysis. This required that an algorithm be written that could split the search_terms and search_terms_ranking variables into pairs and return a new, larger dataset, with an individual data point per search term per product.

Once this preprocessing was completed, the datasets were merged into longitudinal datasets. This merging isolated products whose basic page content (e.g. description, keywords, additional description) remained constant over periods of time. This allowed for visualisation and analysis of daily search rankings of products with consistent page content. Some merged datasets were created:

- the full 7-day Amazon.com dataset;
- a merged 14-day dataset from the November Target.com data (this provided the largest sequential merged dataset from each of the November datasets - some product/search term pairs were not present across the full month); and
- a merged 13-day dataset from the April Target.com data (again, this provided the largest sequential merged dataset due to lack of presence of some product/search term pairs across the full month).

**Breaking character-delimited arrays into product-search term pairs**

To coerce the data into a workable form (product-search term/ranking pairs), an algorithm was provided by the client, written in the R statistical programming language (http://www.r-project.org/). This algorithm works by splitting the Search Terms and Search Terms Rankings (using R’s `strsplit()` function) by the character-delimiter (in this case, a “|” symbol), and creating a new row for each of these pairs for every given product, while keeping the rest of the information about the product intact.

This algorithm - and all other data transformation - proved very computationally expensive: each of the Amazon.com datasets took about two to three and a half minutes to preprocess on a quad-threaded machine with 6GB of system memory.

**A Less Abstract Example**

Table 3.3.1 shows three columns from an example row of the raw Amazon.com data. Each of the columns contain only text data; each term/ranking has an added newline character for easier legibility, but these were not present in the data. However, this is not a convenient form for analysis, so the handling algorithm converted this row into multiple rows (as seen in Table
3.3.2). These rows were then merged with the rest of the page data to form much larger datasets (due to the creation of additional rows).

### TABLE 3.3.1

<table>
<thead>
<tr>
<th>ASIN</th>
<th>Search Terms</th>
<th>Search Terms Rankings</th>
</tr>
</thead>
<tbody>
<tr>
<td>B000052XHI</td>
<td>Pregnancy test</td>
<td>pregnancy</td>
</tr>
</tbody>
</table>

### TABLE 3.3.2

<table>
<thead>
<tr>
<th>ASIN</th>
<th>Search Terms</th>
<th>Search Terms Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>B000052XHI</td>
<td>Pregnancy test</td>
<td>1</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>pregnancy</td>
<td>2</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>good condoms</td>
<td>3</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>condoms online</td>
<td>4</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>condoms brand</td>
<td>13</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>best condoms</td>
<td>21</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>ovulation test</td>
<td>22</td>
</tr>
<tr>
<td>B000052XHI</td>
<td>best time to get pregnant</td>
<td>30</td>
</tr>
</tbody>
</table>
4. **ANALYSIS**

This chapter will provide broad analysis of the data provided, and will outline the main findings of this analysis.

4.1 **Initial Analysis**

Initial examination of the merged datasets raised some concerns - while the Target.com datasets from April were relatively stable in day-to-day search rankings, both the November Target.com and 7-day Amazon.com datasets showed high levels of volatility in their day-to-day search rankings. This volatility made model-fitting infeasible, as there was no longitudinal consistency across each of the days to give a “true” search ranking value to use as a target variable.

In an attempt to maximise the amount of workable, stable data, some k-means clustering methods (see below) were applied to the more volatile November Target.com and 7-day Amazon.com datasets. This analysis provided some data from the November Target.com merged dataset which were stable enough to be used in analysis, however clustering of the Amazon.com dataset returned a least-volatile cluster which still had high levels of volatility - some product/search term pairs had day-to-day changes of up to 50 places in rankings: much too unstable to provide a fair summary search ranking over any period of time. This was also the case for the later Phase 2 Amazon.com datasets, which proved too volatile to allow for immediate model-fitting.

As such, the analysis of the data was split into two parts: 1) Analysis of and model-fitting to stable data; and 2) Analysis of volatile data.

**Cluster Analysis**

Cluster analysis is a technique used to find distinct “groups” of data within a larger dataset. The aim is to generate groups of data that are very internally similar (all of the component data points are quite similar to each other) but are very distinct from other groups. K-means clustering is a quick and efficient method of partitioning a dataset into a number of clusters.

In an attempt to find product-search term pairs that experience little volatility in their day-to-day search rankings, clustering was performed on the daily change in search ranking, rather than daily ranking itself - this matrix of daily change rather than daily recordings is known as a difference matrix.
4.2 Text Summary Analysis

As many of the variables of interest are text-based, some summary statistics needed to be generated in order to analyse these. Simple summary statistics (number of characters; number of words) were applied to the primary text fields of interest: product description (title); additional description (Amazon: bullet-pointed summary information; Target: manufacturer-supplied product description); keywords (vendor-supplied keywords for search relevancy); and enhanced content (large text field towards bottom of Amazon.com product pages, containing detailed manufacturer-supplied product description).

Further summary statistics were generated from these text fields - search term occurrence-frequency in these fields seemed like a valid statistic to use, as it is hypothesised that this may have some impact on search rankings. A more detailed description of the statistics generated and the algorithms used in generation is below.

Search Term Frequency Statistics

A simple text-matching algorithm (detailed below) was written to count the number of times given search terms appeared across some of the text variables in the data. The algorithm produces three different metrics for search term occurrences:

- [variable name]_all – this is a measure of the number of times each word in the search terms appears in a variable (i.e. if search term is “toilet paper,” this counts the number of times “toilet” appears in the variable and adds the number of times “paper” appears in the variable);
- [variable name]_tuples – this is a measure of the number of times the full search terms appear in a variable (i.e. if search term is “toilet paper,” this counts the number of times “toilet paper” appears in the variable, but ignores any occurrences of “toilet” or “paper” in isolation;
- [variable name]_singles – this is a measure of the number of times any of the words in the search terms appears without being part of a tuple (i.e. if search term is “toilet paper,” this calculates the number of times “toilet” appears without being followed by “paper,” and adds the number of times “paper” appears without being preceded by “toilet”).

The primary aim of the algorithm is to count the number of “needles” appearing in each of the “haystacks” in the data, where “needle” refers to search terms (or component words thereof),
and “haystack” refers to the text variable of interest, e.g. *Keywords*. Continuing the metaphor of needles in haystacks, the search terms for each product/search term pair are like a bundle of needles (words), and the three summary statistics generated can be understood as:

- **Tuples**: how many times does the *whole, ordered bundle* of needles appear in the haystack?
- **All**: how many times does each needle (word) appear in the haystack?
- **Singles**: how many times does each needle appear in the haystack *without being part of the bundle*?

This algorithm was applied to *Keywords, Product Description* and *Additional Description* in the Target.com datasets, and also to *Enhanced Content* in the Amazon.com datasets to calculate search term frequencies in fields in which these may contribute to final search ranking.

**Search Term Frequency by Simple Pattern-Matching**

More formally, the algorithm used to generate these statistics comprises three components:

- **Tuple Count**:
  - Convert the search terms “bundle of needles” into all lower-case (using R’s `tolower()` function), and remove any punctuation (using the `stringr` package for R’s `str_replace_all()` function). The “haystack” description string is similarly transformed.
  - Count (using R’s `gregexpr()` function – a simple pattern-matching function) the number of times the “bundle of needles” appears in the “haystack.”

- **All Count**:
  - Split (using R’s `strsplit()` function) the search terms “bundle of needles” into each of its constituent “needles” (if any).
  - Ignore 1-character search terms, or generic “stopword” search terms (“the,” “to,” “in”, “and,” etc.)
  - Pass each needle to the Tuple Count algorithm and sum results.

- **Singles Count**:
  - Take the number returned by the All Count algorithm, and subtract the number of words (ignoring stopwords etc. similarly to All Count) in the search terms multiplied by the number returned by the Tuple Count algorithm.
A Worked Example

Perhaps the easiest way to understand these algorithms is through example, rather than metaphor or simile. Table 4.2.1 shows the Search Terms and Keywords fields of a row of the Amazon.com datasets. Table 4.2.2 shows the processed versions of these fields as processed by the Tuple and All Count algorithms, along with their calculated values.

**TABLE 4.2.1**

<table>
<thead>
<tr>
<th>Search Terms</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pregnancy test</td>
<td>First Response Early Result Pregnancy Test, 3 tests, Packaging May Vary, First Response, 2260090127</td>
</tr>
</tbody>
</table>

**TABLE 4.2.2**

<table>
<thead>
<tr>
<th>Summary Function</th>
<th>Count</th>
<th>Processed Search Term</th>
<th>Processed Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Count</td>
<td>1</td>
<td>pregnancy</td>
<td>firstresponseearlyresultpregnancytest3testspackagingmayvaryfirstresponse2260090127</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>test</td>
<td>firstresponseearlyresultpregnancytest3testspackagingmayvaryfirstresponse2260090127</td>
</tr>
<tr>
<td>Tuple Count</td>
<td>1</td>
<td>pregnancytest</td>
<td>firstresponseearlyresultpregnancytest3testspackagingmayvaryfirstresponse2260090127</td>
</tr>
</tbody>
</table>

The terms are highlighted as they would be found by the respective algorithms. We can see that All Count would return 3 (1 for “pregnancy” + 2 for “test”) occurrences, and Tuple Count would return 1. Given this information, Table 4.3.3 can be constructed, allowing for Single Count to be calculated as:

\[
\text{Single Count} = \text{All Count} - (\text{Tuple Count} \times \text{Number of Search Term Words})
\]

\[
= 3 - (1 \times 2)
\]

\[
= 3 - 2
\]

\[
=> \text{Single Count} = 1
\]
TABLE 4.3.3

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Terms</td>
<td>“Pregnancy test”</td>
</tr>
<tr>
<td>Number of Search Term Words</td>
<td>2 - “Pregnancy” &amp; “test”</td>
</tr>
<tr>
<td>Tuple Count</td>
<td>1</td>
</tr>
<tr>
<td>All Count</td>
<td>3</td>
</tr>
<tr>
<td>Single Count</td>
<td>1</td>
</tr>
</tbody>
</table>

4.3 Stable Data and Modelling

The presence of stable data allowed for a median search ranking to be used as the target variable of analysis. The median was selected as a more appropriate measure than the mean of the data, as even the stable data still saw “spikes” - there were some days that saw large changes in ranking, but overall the data saw longer periods of stability. A mean value would be more prone to weighting the days with “spikes,” to which the median is more robust. Once a clear target variable was chosen, model-fitting and analysis could be performed.

In order to examine - and take into account - the effect of the sales performance and conversion rate, analysis was run with and without inclusion of the product’s SKU number as a predictor variable. It is hoped that, by attempting to capture this latent variable, model performance will be improved. If this is the case, the results from the models including SKU number should give better indication of importance of the remaining variables, which no longer have to “pick up the slack” of accounting for sales performance and conversion rate.

The client’s aim for this analysis was not necessarily to be able to make good predictions about the data, but to be able to understand what contributes to search ranking. This means that a “good” model (i.e. one which predicts search ranking with high accuracy) is not necessarily as
desirable as a model which is easily interpretable, as the client’s marketing department wish to be able to use the model to inform future work, rather than simply use it to predict past work. This trade-off between model performance and interpretability led to a decision to use three model types:

- **Linear Regression** - these models are quite easily interpretable, and can be reasonably accurate for data which have linear relationships with target variables. These models handle linearity well, but do not deal well with interactions - particularly step-wise interactions where an interaction can be different within different bounds of one of the variables;
- **Decision Trees** - these models are very easily interpretable, and provide good accuracy for data wherein multiple variables interact with each other, but do not handle linear relationships well; and
- **RuleFit** - a modelling method designed by Jerome Friedman (Friedman and Popescu, 2005), this is a bit like a mix of regression and trees. The model first builds a “forest” of trees (this is a technique known as ensembling), using different subsets of the data for each tree, then fits a regression model to the “paths” from the “root nodes” (starting points) to the “terminal nodes” (ending points) of the trees. It then outputs a number of “rules” about the data for a good level of interpretability. Theoretically, a model of this type benefits from the strengths of both regression and of trees, and should provide better results than either of the two if the data are neither wholly linear nor completely nonlinear.

The dataset was split into two sets: 67% (around 4900 rows) used for model training; and 33% (around 2450 rows) used for assessing model performance. Each of the models was initially trained using the same variables:

- (sku_number) - each product’s unique identifier (models were fit with this included and excluded);
- keywords_all ,keywords_tuples and keywords_singles - search term occurrences in the keywords field;
- description_all, description_tuples, description_singles - search term occurrences in each product’s title;
- add_desc_all, add_desc_tuples, add_desc_singles - search term occurrences in each product’s additional description;
• description_len, add_desc_len - the number of characters in each of the product title and additional description fields; and
• description_wordc, add_desc_wordc - the number of words in each of the product title and additional description fields.

Fitting a Regression Model
A linear regression model was first fit to the data, as it is a well-known and relatively simple model to understand. The model was initially fit including all variables as predictors. In order to improve (at least theoretical) model performance, the model was tweaked using stepwise regression to determine which predictors carried the best “signal” of the data, and which were “noise” variables. Table 4.3.1 (below) shows the results from models fit using different sets of variables. This model type did not perform exceptionally well, but nor did it perform so badly as to be dismissable.

While the effect of the latent marketplace performance appears to be a key contributor to search ranking, there does seem to be some relationship between page content and search ranking, although this appears to be a complex relationship. The model has a far higher R-Squared value when SKU number is included, although there is only a very slight improvement in root mean square error - this reflects a higher proportion of search ranking variance being explained by the model, but without a much-improved prediction accuracy. Including some measure of latent product performance does lead to a better model, but it is clear that some variable outside of page content is still having a large effect.

Fitting a Decision Tree
In order to explore multiple nonlinear interactions between variables, and provide an easily interpretable model, a decision tree was fit to the data. The model that included the SKU number as a predictor used this variable almost exclusively, perhaps lending weight to the hypothesis that page data is less relevant to overall search rankings on Target.com than some other aspect of product performance on the marketplace.

In truth, neither model performed very well with the data. This is a feature of the way that decision trees are built - no new “branch” is added if the trade-off between added complexity and performance is not above a certain threshold. This contributes to the theoretical power of trees - they will generalise well, without being so complex as to be uninterpretable. However, in this data, this proved to be a hinderance to model performance. There were so few terminal
nodes in the resulting models that “banding” occurred in prediction - this is where many predicted values for test data fall into the same terminal node, and so take the same value. This can be seen in the predicted vs. observed plot in Table 4.3.1.

**Using RuleFit Methods**

In an attempt to marry these two types of model - and gain some better understanding of the relationship between search rankings and page content - a rulefit model was fit to the data. Table 4.3.2 shows variable importances from these models.

This model type performed admirably on the test dataset. The combination of decision trees, ensembling methods, and regression seems to be a good one, and benefits from a high level of interpretability and better performance than either of the previous model types.

**Interpretation of Results**

Table 4.3.1 shows graphs of predicted values against actual values for the test data set from each of the best-performing models. The blue line along the diagonal shows the line of perfect fit; the distance from each point to this line is the model prediction error for that point. We can see the “banding” of the predictions from decision tree very clearly - this is clearly not a good model for the data. The regression and rulefit models show much better test prediction, although the rulefit model shows slightly tighter predictions, especially to the line of perfect fit.

<table>
<thead>
<tr>
<th>Model</th>
<th>Linear Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted vs. Observed</td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>RMSE</td>
<td>17.518</td>
</tr>
</tbody>
</table>

**TABLE 4.3.1**
Also included in Table 4.3.1 are estimates for root mean square error (RMSE) and R-squared values for the predictions. Lower RMSE values indicate that a model is making predictions that are reasonably close to the line of perfect prediction, and so RMSE acts as a good indicator of
model performance across the different model types. The R-squared statistic measures goodness-of-fit of regression lines by calculating the proportion by which the variance of errors is less than the variables of the outcome variable. It is hard to define a "good" value for R-squared, but given the "noise" in the data - or the hypothesis of expected weak signal - low values are not bad values; being able to reduce even 20% of the variance of the data may give some information of general interest.

Table 4.3.2 shows variable importance plots from the rulefit modelling. Where SKU number was used, that appears to be by far the most important variable, with the next most important variable being description-singles, although this is only about 10% of the importance of SKU number. When SKU number is removed, description-tuples becomes the most important variable, followed closely by additional description word count. Indeed, the linear regression model finds description-tuples and additional description word count to be important (at the 5% and 1% confidence levels respectively), but description-singles are seen as "noise" in a purely linear model - it appears this may be a variable that interacts with many other variables, but is not a good predictor on its own.

<table>
<thead>
<tr>
<th>RuleFit Variable Importance including SKU number</th>
<th>RuleFit Variable Importance excluding SKU number</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph 1" /></td>
<td><img src="image2.png" alt="Graph 2" /></td>
</tr>
</tbody>
</table>

Both linear models agree that description-all is important at the 0.1% confidence level, and roughly agree on its coefficient - for each time a word in the search term appears in the product title, its search ranking should be a few places better. These models also agree on keywords-all being important to some degree, although with a small coefficient - since this is a size-limited field, and conjecture pointing to this being a search relevancy factor rather than ranking factor,
may be better to be more diverse and have a product rank under another term than to use up character limits to repeat a term. However, it is the rulefit model that performs the best on test set prediction, with an R-squared of around 55% when excluding SKU number, and around 52% when including it. One would expect that inclusion of more variables would boost R-squared scores (as it has with standard linear regression), but here we see the opposite. However, the model that includes SKU number paces a very high relative importance on it rather than on the other variables, which may indicate the other variables “picking up the slack” in the model built without SKU number. This would be indicative of overfitting to the training data, and so the model may not generalise well. The model achieves low RMSE values (around 10.5) on test set prediction, which is a far better fit than the values from regression.

4.4 Volatile Data

In order to narrow analysis, and ensure analysis validity (Target and Amazon may use different ranking methods), only the Phase 2 Amazon.com datasets were analysed for volatility. There are two primary reasons for this. Firstly, there was considerably more data in the Phase 2 Amazon.com volatile datasets than in the Phase 1 Target.com datasets. Secondly, Amazon is a bigger (and thus more strategically important) marketplace to understand.

Initial Examination

Of the nearly 5.75 million points of data, containing over 12,000 unique product/search term pairs, 379 product/search term pairs related to products whose page contents did not change dramatically; their Product Description, Keywords, Additional Description, and Enhanced Content sections remained constant over the 6 month period. Were the search rankings of the data only related to these sections (the primary variables of interest to a naïve text search), one would not expect to see much volatility in this subset of the data. However this is not the case – Figure 4.4.1 shows a graph of the search rankings of these 379 products over the 180 days, which show a high overall volatility, with only a few periods of consistency. This alone confirms the hypothesis that page text data is not the only (or even, perhaps, the primary) determinant of search ranking.
Analysis of Stock Availability and Volatility

Holding these same variables constant, we can assess whether the volatility is caused by stocking levels - for the same 379 product/search term pairs, each day’s availability was also measured (with anything other than “In Stock” treated as being out of stock). However, again, we find that stocking levels are not a key contributor to the volatility – Figure 4.4.2 shows the same plot of the 379x181 search rankings, along with the same number of stock availabilities for each of the products. Here we see that each of these products are almost always “In Stock,” so this does not seem to be a measure by which to analyse the volatility of the search rankings.
Identification of Some Smoothing Period for Volatility

In order to determine a smoothing period for the Amazon.com data, linear regression models were fit to data with median values over 7 days, 14 days, 28 days and 56 days, then tested on out-of-sample data from the following 7, 14, 28, and 56 days to assess performance. The aim of this modelling is to determine whether any period of time shows good predictability, which would allow for some method to overcome or work around the high volatility of the data and begin to analyse the page content-search ranking relationships.

Due to time constraints, not much analysis could be performed in this area, although preliminary results from this analysis show that median search ranking over 4 weeks is better than over 2 weeks, which is better again than over 1 week (RMSE values: 26.73; 27.89; 29.92; R-squared values: 20.58%; 17.62%; and 12.88% respectively) when used to predict an out-of-sample median search ranking over the same periods. However, these are still not particularly good results, and warrant further investigation in this area.
Caveats

While these variables do not seem to explain the volatility of the search rankings of the data, they may contribute to the overall search ranking of a product. The reason such a small subset of the data was used (379 product/search term pairs) for stock level analysis was to provide a dataset where one could fix many variables to a constant value. In theory, were the fixed values the sole determinants of search rankings, there would be little volatility in the search rankings. However, this initial examination of volatility points to an external source of noise in the data - this may include sales performance or conversion rate data as hypothesised by the client (and from general conjecture). In order to better examine the relationship between page data and search rankings, the effects of these external factors must be mitigated or accounted for. Some of this data may be unavailable (e.g. Amazon will probably not share conversion rate data), but sales performance should be available - or at least estimable - and should be examined in order to determine its effect on search rankings.
Appendix A: Project Outline

Client: Clavis Insight

Project: Data mining of Amazon Product Information

Location: 7th Floor, O’Connell Bridge House

Client Contact: Clare Conway, +353 1 2543445, clare.conway@clavisinsight.com

Dept. Contact: Aideen Keaney

Client Background

Clavis Insight delivers eCommerce Insights and Online Store Audits to leading Global FMCG Companies across the Food & Beverage, Personal Care, Baby Care, Pet Care and Household sectors. These reports are designed to enable our Customers to optimize their product distribution, content integrity and placement in the digital channel in order protect their brand online and grow their sales. Clavis is headquartered in Dublin on D’Olier Street. We have offices in Cambridge, MA, Shanghai and London.

Project Background

Acquiring in excess of 100 million data points of product information monthly, Clavis test this product information for content accuracy and effectiveness. The core Clavis Insight product is a SAAS based and provides daily dashboards and detailed reports to Customers. Looking at a retailer product page such as http://www.amazon.com/LOreal-Paris-EverPure-Sulfate-Free-Moisture/dp/B001P1ZELS we examine the component text fields to ensure that the content being displayed is in line with what the manufacturer expects. We also examine the location of this product page with respect to other products and test how easy it is to find with key search terms.

Client Requirement

We have a number of areas of interest. We would propose starting with Amazon.com data and then extending to other US and possibly UK retailers. Students could examine one of the following:

- Is there statistical evidence to support the hypothesis that an improvement in page content quality will positively impact search placement results? Search results are a key driver for sales conversion on a retailer website. The analysis should focus on identifying and quantifying the key variables that can be used to explain improvement in search performance over time.
• Of the main metrics we track such as price, ratings & reviews, search – are there metrics that are strongly correlated and equally are there metrics that appear to be independent of overall performance.

• Analysis of volatility of daily metrics, with a view to determine the minimum time-period for aggregation that will “smooth” trends in the daily metric values – e.g. 7-day / 14-day…

What is involved for the student?

The main technical emphasis in this project will be on data mining and handling extensive data sets. The student will need to become thoroughly familiar with the client’s business processes, interpret the client’s requirements in practical terms and provide actionable information on interrelationships in the data base. Some knowledge of data mining and data handling techniques is assumed, but an imaginative approach to data mining will be essential. (Infrastructure includes SQL and PostGres, familiarity with R and Python would be beneficial.)
Appendix B: Interim Report

Project: Data Mining of Amazon Product Information
Client: Clare Conway, Clavis Insight
Student: Simon Major
Supervisor: Arthur White

Review of Background and Work to Date

Clavis Insight is an e-commerce analytics firm, with many very large clients. Their clients sell products on a number of different e-commerce platforms, with Amazon being one of the largest (and most important).

Clavis have provided both daily and monthly data on a large number of products, presumably stripped from the product pages on Amazon.com, along with data on search terms and rankings that their clients wish to optimise in order to boost sales. Our goal is to identify which factors most influence these search rankings.

To date, we have done work on preprocessing of the data to allow for analysis, and some basic examination of two of the variables in the data against the search rankings. So far, these have not provided much insight into the data.

Terms of Reference

The data consists of “large N”, consisting of c. 18,000 rows, with some preprocessing required before the data could be analysed effectively. Some of the variables are continuous, quantitative data, but most are strings of text which can be handled in many different ways.

Ideally, some model can be built which either provides insight into the Amazon search algorithm for rules-of-thumb or which aids prediction of search ranking of products given data.

There seems to be some external factor playing a part in the variance of the search rankings of the products, outside the scope of the data currently provided (this could be sales data), which could lead to difficulty in analysis as the “signal” of the data being examined looks to be lost behind the “noise” of this external factor.

Should this be the case, it may be that more work can be done to identify which factors may be of greater interest than those currently under examination.

Further Work

Investigation into some external factor’s contribution to variance of search rankings must be performed. Once the results of the investigation have been taken into account, further
analysis of the data will be performed. The first analysis I wish to perform relates to the number of times the search terms appear in a product’s page, as this could (and should) be a good indicator of search term relevance, and thus search term ranking. This requires the use of text-mining techniques to find this count, and then further analysis into the effect of count of search terms on a page on the product’s search ranking.

Conclusions

We have a clear objective from the client, and the data in a workable (although maybe not final) form. There is much work yet to be done, with particular importance on the investigation into the variance of the search rankings from external factors.
Appendix C: RuleFit (excluding SKU number) top 15 rules in order of importance:

Rule 1: 2 variables
support = 0.5880  coef = 4.587  importance = 100.0

description_tuples: range = -0.9900E+36 1.500
add_desc_len: range = -0.9900E+36 527.5

Rule 2: 2 variables
support = 0.4654  coef = 3.248  importance = 71.76

keywords_singles: range = 3.500 12.50
description_tuples: range = -0.9900E+36 4.000

Rule 3: 3 variables
support = 0.1464  coef = 3.962  importance = 62.03

keywords_singles: range = -0.9900E+36 1.500
description_tuples: range = -0.9900E+36 0.5000
add_desc_wordc: range = -0.9900E+36 155.5

Rule 4: 2 variables
support = 0.7165  coef = 2.988  importance = 59.65

keywords_singles: range = 1.500 8.500
description_wordc: range = 5.500 0.9900E+36

Rule 5: 2 variables
support = 0.2822  coef = -2.663  importance = 53.09

description_len: range = -0.9900E+36 54.50
add_desc_len: range = -0.9900E+36 285.0
Rule 6: 3 variables

support = 0.7719  coeff = 2.855  importance = 53.06

description_all: range = -0.9900E+36  3.500

description_tuples: range = -0.9900E+36  1.500

add_desc_singles: range = -0.9900E+36  10.50

Rule 7: 2 variables

support = 0.7221  coeff = 2.430  importance = 48.22

description_tuples: range = -0.9900E+36  1.500

description_len: range = -0.9900E+36  73.50

Rule 8: 2 variables

support = 0.1579  coeff = 2.906  importance = 46.93

description_all: range = -0.9900E+36  2.500

add_desc_len: range = 554.0  1253.

Rule 9: 1 variables

support = 0.8506E-01  coeff = 3.794  importance = 46.88

add_desc_wordc: range = 126.5  287.5

Rule 10: 3 variables

support = 0.1042  coeff = 3.114  importance = 42.14

description_tuples: range = -0.9900E+36  1.500

description_singles: range = -0.9900E+36  0.5000

add_desc_len: range = -0.9900E+36  398.0

Rule 11: 3 variables

support = 0.3967  coeff = 1.943  importance = 42.10

description_tuples: range = -0.9900E+36  1.500

description_singles: range = -0.9900E+36  1.500

keywords_wordc: range = 36.50  0.9900E+36
Rule 12: 3 variables

support = 0.6028E-01  coeff = 3.907  importance = 41.19

keywords_singles: range = 7.500  0.9900E+36

description_all: range = -0.9900E+36  2.500

keywords_wordc: range = -0.9900E+36  51.50

Rule 13: 2 variables

support = 0.4169  coeff = 1.797  importance = 39.25

keywords_all: range = -0.9900E+36  5.500

add_desc_wordc: range = 30.50  0.9900E+36

Rule 14: 3 variables

support = 0.3137  coeff = 1.826  importance = 37.53

keywords_singles: range = 5.500  0.9900E+36

description_tuples: range = -0.9900E+36  1.500

description_wordc: range = 6.500  0.9900E+36

Rule 15: 2 variables

support = 0.4427  coeff = -1.618  importance = 35.60

keywords_all: range = 5.500  0.9900E+36

description_wordc: range = -0.9900E+36  12.50
## Appendix D: Linear Regression Model Outputs

<table>
<thead>
<tr>
<th>Model</th>
<th>Terms</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>P-value</th>
<th>Significance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression with SKU Number – tuned using step-wise variable selection</td>
<td>(intercept)</td>
<td>-57.6667</td>
<td>22.9583</td>
<td>0.012</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>keywords_all</td>
<td>1.1372</td>
<td>0.2831</td>
<td>6.01e-5</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>description_all</td>
<td>-7.953</td>
<td>0.9137</td>
<td>&lt; 2e-16</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>description_tuples</td>
<td>-2.1493</td>
<td>0.6923</td>
<td>0.0019</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>add_desc_tuples</td>
<td>0.7938</td>
<td>0.3863</td>
<td>0.0399</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>description_len</td>
<td>1.4214</td>
<td>0.3769</td>
<td>0.000165</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>add_desc_len</td>
<td>-0.6531</td>
<td>0.29</td>
<td>0.02438</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>add_desc_wordc</td>
<td>4.2114</td>
<td>2.1901</td>
<td>0.05456</td>
<td>10%</td>
</tr>
<tr>
<td>Linear Regression without SKU Number – tuned using step-wise variable selection</td>
<td>(intercept)</td>
<td>40.14</td>
<td>0.9334</td>
<td>&lt; 2e-16</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>keywords_all</td>
<td>0.68386</td>
<td>0.21</td>
<td>0.0012</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>description_all</td>
<td>-4.9812</td>
<td>0.6553</td>
<td>3.48e-14</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>description_tuples</td>
<td>-1.508</td>
<td>0.71</td>
<td>0.03393</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>add_desc_all</td>
<td>-0.4026</td>
<td>0.1366</td>
<td>0.00322</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>add_desc_tuples</td>
<td>0.8858</td>
<td>0.4158</td>
<td>0.03322</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>description_len</td>
<td>-0.0452</td>
<td>0.01669</td>
<td>0.00677</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>add_desc_len</td>
<td>-0.0152</td>
<td>0.008334</td>
<td>0.06911</td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td>add_desc_wordc</td>
<td>0.08269</td>
<td>0.0543</td>
<td>0.12789</td>
<td>&gt; 10%</td>
</tr>
</tbody>
</table>