ESTIMATING THE CAUSAL EFFECTS OF REVIEWS ON SALES
AMAZON KINDLE EBOOK SALES

AMAZON LOGO

PROJECT FINAL REPORT

PROJECT SUMMARY

REPORT DATE	PROJECT NAME	PREPARED BY
11/25/2018	ESTIMATING THE CAUSAL EFFECTS OF REVIEWS ON SALES	Seth Turnage

INTRODUCTION

It is no secret that Amazon Kindle is having sales problems. Along with formatting issues (resulting in poor reviews), Kindle eBooks often don't inspire enough confidence for consumers to chance purchasing a Kindle eBook as opposed to a physical copy. For potential eBook publishers, having an analytical tool to help quantitatively gauge how their reviews affect the number of sales (downloads) of a Kindle eBook, in conjunction with the effect that the description has (whether it is interesting, misleading, reassuring, etc.) is vital to helping map go-to-market strategies for potential Kindle eBook creators.

PROBLEM STATEMENT

Our task is to create an analytical model which analyzes reviews using textual analysis in conjunction with a TF-IDF to identify certain keywords consistent between reviews in different products from related genres. Then, a model which identifies consumer 'concerns' will be correlated to sales-ranking (with the proper coefficient applied) of an eBook, using regression

METHOD

First, an array of Amazon id's are fed to item_compare_scrape() in amazon_scrape.py. This outputs json files for each product in a dump directory ('../json'). These folder is then fed to TF_IDF in textual_analysis.py, which reads from the json files and assembles a dictionary of terms, which a unique id ('AmazonId_ReviewNumber') iterated for each occurrence of the word in the dictionary. Then, the dictionary is read into a TF-IDF dictionary, which outputs a value for each key-value, and a row index for each product page. Finally, this outputs an array containing two values, a TF-IDF dictionary for reviews above 3 stars, and a TF-IDF dictionary for those below three stars. The star range initially fed into item_compare_scrape() is an optimization which will directly affect these outputs. Finally, these dictionaries are each fed to regressionFrom_TF_IDF() from data_model.py, which assembles a basic linear regression model for each using the sklearn package, and outputs a mean squared error for each (for each function call respectively).

PROJECT OVERVIEW

TASK	% DONE	DUE DATE	PROJECT MEMBER	NOTES
AMAZON REVIEW TEXT SCRAPER	100	11/26/2018	Seth Turnage	Uses lxml parser, loads pages from asin
AMAZON REVIEW TEXT ANALYZER	100	11/26/2018	Seth Turnage	Analyzes test using TF-IDF
REVIEWER 'KEYWORD' CLUSTERS	100	11/26/2018	Seth Turnage	Creates a Dictionary; Uses autocorrect & stop word filtering
REVIEW - SALES REGRESSION MODEL	73	11/26/2018	Seth Turnage	Keywords to sparse for meaningful regression