CHARLES UNIVERSITY
FACULTY OF SOCIAL SCIENCES
Institute of Economic Studies

Predicting purchasing intent on ecommerce websites

Master's thesis

Author: Bc. Marek Vařeka

Study program: Economics and Finance

Supervisor: doc. PhDr. Ladislav Krištoufek, Ph.D.

Year of defense: 2020

Declaration of Authorship

The author hereby declares that he or she compiled this thesis independently, using only the listed resources and literature, and the thesis has not been used to obtain any other academic title.

The author grants to Charles University permission to reproduce and to distribute copies of this thesis in whole or in part and agrees with the thesis being used for study and scientific purposes.

Prague, April 26, 2020

Marek Vařeka

Abstract

This thesis analyzes behavior of customers on an e-commerce website in order to predict whether the customer is willing to buy something or is just window shopping. In addition the secondary model predicts, if the customer is going to leave the e-commerce website in next few clicks. To answer this questions different frameworks are tested. The base model used is the Logit model. The base model is compared with more sophisticated methods in machine learning - with neural networks. The best results were yielded by Recurrent neural network - the Long Short-Term Memory (LSTM). The results of the analysis confirm importance of the click stream data and calculated features that track user behavior on the e-commerce website, type of the page (product, category, information), product variance and category variance. The thesis emphasizes practical implications of this models. Two possible practical implementations are presented. The models are tested in novel ways to see how would they perform if implemented on the real e-commerce website.

JEL Classification	C45, C52, L81
Keywords	e-commerce, LSTM, purchasing intent, website abandonment

Title	Predicting purchasing intent on ecommerce websites
Author's e-mail	`marek.vareka@icloud.com`
Supervisor's e-mail	`ladislav.kristoufek@fsv.cuni.cz`

Abstrakt

Tento článek analyzuje chování zákazníků na webových stránkách elektronického obchodu s cílem předpovědět, zda je zákazník ochoten si něco koupit nebo se jen dívá. Kromě toho sekundární model předpovídá, zda zákazník během několika málo kliknutí opustí web elektronického obchodu. Pro zodpovězení těchto otázek jsou testovány různé metody řešení. Použitý základní model je Logit. Základní model je porovnán se sofistikovanějšími metodami strojového učení - s neuronovými sítěmi. Nejlepší výsledky byly dosaženy pomocí rekurentní neuronové sítě - Long Short-Term Memory (LSTM). Výsledky analýzy potvrzují důležitost údajů o tocích kliknutí a napočtených proměnných, které sledují chování uživatelů na webové stránce elektronického obchodu, typ stránky (produkt, kategorie, informace), variance produktu a varianci kategorie. Práce zdůrazňuje praktické využití těchto modelů. Jsou představeny dvě možné praktické implementace. Modely jsou testovány novými způsoby, aby se zjistilo, jak by fungovaly, kdyby byly implementovány na skutečné webové stránce elektronického obchodu.

Klasifikace JEL	C45, C52, L81
Kličová slova	e-commerce, LSTM, záměr nákupu, opuštění webových stránek
Název práce	Předpovědi spotřebitelského chování v eshopech

E-mail autora	`marek.vareka@icloud.com`
E-mail vedoucího	`ladislav.kristoufek@fsv.cuni.cz`

Acknowledgments

The author is grateful especially to doc. PhDr. Ladislav Krištoufek, Ph.D. for his help and guidance with my thesis.

This thesis is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 681228.

Typeset in LaTeXusing the IES Thesis Template.

Bibliographic Record

Vařeka, Marek: *Predicting purchasing intent on ecommerce websites.* Master's thesis. Charles University, Faculty of Social Sciences, Institute of Economic Studies, Prague. 2020, pages 77. Advisor: doc. PhDr. Ladislav Krištoufek, Ph.D.

Contents

List of Tables **ix**

List of Figures **x**

Acronyms **xi**

Thesis Proposal **xii**

1 Introduction **1**

2 Literature review **3**

3 Data **8**

3.1 Availability of data 8

3.2 General overview 9

3.2.1 Traffic features 9

3.2.2 Behavioral features 10

3.2.3 Transaction features 10

3.3 Key performance indicators 10

3.3.1 General overview 10

3.3.2 Geographical analysis 12

3.3.3 Cart abandonment 12

3.3.4 Time analysis 13

3.3.5 Conversion rate 13

3.3.6 Path analysis 14

4 Methodology **15**

4.1 Artificial neural network - general model 15

4.1.1 Activation function 16

4.1.2 Cost function 18

4.1.3 Gradient descent . . . 18
4.1.4 Number of layers and neurons . . . 20
4.1.5 Exploding and Vanishing gradient . . . 20
4.2 Recurrent neural network . . . 22
4.2.1 Long Short-Term Memory . . . 23
4.3 Word embedding . . . 26

5 Data preparation 28
5.1 Data quality . . . 28
5.2 Missing variables treatment . . . 29
5.3 Feature creation . . . 30
5.3.1 Time features . . . 31
5.3.2 Behavior features from click stream data . . . 31
5.3.3 Stock markets . . . 33

6 Model - Predicting purchasing intent 35
6.1 Correlation analysis . . . 35
6.2 Base model - Logit . . . 38
6.2.1 Results . . . 38
6.3 Vanilla neural network . . . 39
6.3.1 Model architecture . . . 39
6.3.2 Results . . . 40
6.4 LSTM . . . 42
6.4.1 Model architecture . . . 42
6.4.2 Results . . . 42

7 Model - Predicting website exit 45
7.1 Correlation analysis . . . 45
7.2 LSTM . . . 47
7.2.1 Model architecture . . . 48
7.2.2 Results . . . 49

8 Practical implementation 51

9 Conclusion 53

Bibliography 60

A Logit results I

B Used variables **II**

List of Tables

3.1 Basic statistics 11
3.2 Basic statistics (Buying event only) 11
3.3 Geographical analysis 12
3.4 Weekday analysis - averages 13
3.5 Path analysis 14

5.1 Stock markets overview 34

List of Figures

4.1 ANN diagram 17
4.2 Sigmoid function 21
4.3 RNN diagram 22
4.4 RNN detail 23
4.5 LSTM 24
4.6 LSTM shapes 25

5.1 Google query 30

6.1 Correlation matrix - purchasing intent 1 36
6.2 Correlation matrix - purchasing intent 2 37
6.3 Logistic regression results OoS 39
6.4 Vanilla ANN diagram 40
6.5 Vanilla ANN results - AUC OoS 41
6.6 LSTM diagram 43
6.7 The LSTM model results 44

7.1 Correlation matrix - exit 1 46
7.2 Correlation matrix - exit 1 47
7.3 LSTM diagram - exit 48
7.4 The Exit model results 50

8.1 Proposed solution diagram 52

Acronyms

ANN Artificial neural network

ANNs Artificial neural networks

AUC Area under curve

CNN Convolutional neural network

IS In sample

knn k-nearest neighbors

LSTM Long Short-Term Memory

NAG Nesterov accelerated gradient

OoS Out of sample

RNN Recurrent neural network

SGD Stochastic gradient descent

Master's Thesis Proposal

Author	Bc. Marek Vařeka
Supervisor	doc. PhDr. Ladislav Krištoufek, Ph.D.
Proposed topic	Predicting purchasing intent on ecommerce websites

Motivation Over the past decade, the evolution of the hardware and the software led to the rise of the internet. Based on OECD statistics the percentage of the people with the access to the internet, has risen from 57% in 2007 to 87% in 2018 in the OECD countries. The rise of the internet is directly correlated to rise of the ecommerce business. Currently there is rise in online marketplaces and tremendous growth of online advertising and marketing. It is not a coincidence, that the biggest company in terms of market capitalization is the internet retailer Amazon.

In today's fast paced interconnected world there is a big demand for the integration between offline and online point of sales – Retail 4.0. In order to succeed, retailers around the world have to provide tailored solutions to their customers to ensure their loyalty. This approach is progressively demanding the accurate prediction of consumer's purchase intention which leads to effectively targeted marketing campaigns.

From perspective of the online seller it is important to know which web-sites visitors are the most probably going to buy something and what do they want. Based on that information, tailored marketing strategy can be developed to address customer's needs and increase sales. This information can be gathered by latest methods in statistics, econometrics and machine learning based on the behavior of the customer on the website. Recently Bag et al. (2019) used linear regression and neural network analysis in order to correctly predict purchasing intent in camera e-store. Sheil et al. (2018) utilize use of neural networks to find out which customers are going to buy goods from online store.

Hypotheses

Hypothesis #1: Stock market index is valid feature that can help in prediction of purchasing intent on ecommerce website.

Hypothesis #2: Consumer mood indicators (sunshine and temperature) are valid features that can help in prediction of purchasing intent on ecommerce website.

Hypothesis #3: The Hybrid neural network based on recurrent neural network that utilizes stock market index and mood features among other common microeconomic features, is the best solution for predicting purchasing intent and recommending products to those customers who intend to buy something on the ecommerce websites

Methodology First of all, an extensive research of all features in dataset will be performed. Google provides documentation to every variable (microeconomic data) in its dataset. Applying those information huge variety of features will be selected/generated based on the best practices in the field and economic intuition. Using geographic information from the dataset (country and city) stock market index and mood indicators will be added to the dataset. In order to train neural network more efficiently and reduce overfitting feature selection/elimination algorithms will be used. There are three most used categories of feature selection algorithms: filters, wrappers and embedded methods. Filters select features independent to the predicator whereas wrappers assess features according to their usefulness to a predictor Guyon et al. (2003). Embedded methods work similarly as wrapper methods and are generally more efficient. Sakar et al. (2019) tested different filter methods for predicting purchasing intent on ecommerce website: Columbia in Turkey and achieved the best results with Minimum Redundancy Maximum Relevance feature selection method (mRMR).

Filter methods will be used on generated features due to their more simplistic and efficient nature. Generated features will be tested for the relevance using chi-squared (chi?) statistical test for non-negative features, mutual information (MI), correlation matrix with heat map and mRMR. After performing the analysis it should be clear if the hypothesis one and two is true or false.

Selected features will be fed to neural network's model. Although winning solution for predicting purchasing intent competition: RecSys Challenge 2015 by Romov et al. (2015) used Gradient Boosting Machine method with extensive feature engineering, Sheil et al. (2018) used recurrent neural network (RNN) and achieved better results on the same dataset. Therefore the base model will be based on RNN. Model will be further developed into Hybrid Neural Network in order to achieve the best possible prediction.

For the analysis the data from Google BigQuery - Analytics sample data from the Google Merchandise Store will be used. Google Merchandise Store sell apparel, drinkware, bags, notebooks, pens etc. with Google logo. Each row in the dataset

contains more than 250 features about customer and his or her behavior on the website. It includes: • Traffic source data • Content data • Transactional data

Expected Contribution The expected contribution of the thesis is to find relevant features which can help predict customer's intent to buy goods online. Furthermore to develop the best possible model which utilizes those features and predicts which customers are the most probable to buy something and what they want to buy.

The dataset from the Google Merchandise store contains much more microeconomic features than datasets used in previous studies which focus on prediction of purchasing intent. Also inclusion of the stock market index and mood variables to the best of my knowledge was not tested before.

All this information can be very useful to retailers from online stores to banks to better target their customers utilizing data about their behavior. This smart solution enables to do marketing campaigns more efficiently and provide better customer services which should increase revenues.

Outline

1. Motivation
2. Literature review
3. Data
4. Selection of valid features based on the best practices and the economic theory
5. Testing features for significance
6. Methodology – Neural networks
7. Model
8. Conclusion

Core bibliography

1. Bag, Sujoy, Manoj Kumar Tiwari, and Felix TS Chan. "Predicting the consumer's purchase intention of durable goods: An attribute-level analysis." Journal of Business Research 94 (2019): 408-419.
2. Edmans, Alex, Diego Garcia, and ?yvind Norli. "Sports sentiment and stock returns." The Journal of Finance 62.4 (2007): 1967-1998.
3. Furnham, Adrian, and Rebecca Milner. "The impact of mood on customer behavior: Staff mood and environmental factors." Journal of Retailing and Consumer Services 20.6 (2013): 634-641.

4. Graves, Alex, and Jürgen Schmidhuber. "Offline handwriting recognition with multidimensional recurrent neural networks." Advances in neural information processing systems. 2009.

5. Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." Journal of machine learning research 3.Mar (2003): 1157-1182.

6. Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." arXiv preprint arXiv:1511.06939 (2015).

7. Hirshleifer, David, and Tyler Shumway. "Good day sunshine: Stock returns and the weather." The Journal of Finance 58.3 (2003): 1009-1032.

8. Jannach, Dietmar, Malte Ludewig, and Lukas Lerche. "Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts." User Modeling and User-Adapted Interaction 27.3 (2017):351-392.

9. Lokhande, P. S., and B. B. Meshram. "Analysis and design of web personalization systems for E-Commerce." (2015).

10. Loyola, Pablo, Chen Liu, and Yu Hirate. "Modeling user session and intent with an attention-based encoder-decoder architecture." Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, 2017.

11. Romov, Peter, and Evgeny Sokolov. "Recsys challenge 2015: ensemble learning with categorical features." Proceedings of the 2015 International ACM Recommender Systems Challenge. ACM, 2015.

12. Sakar, C. Okan, S. Olcay Polat, Mete Katircioglu, and Yomi Kastro. "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks." Neural Computing and Applications 31, no. 10 (2019): 6893-6908.

13. Sheil, Humphrey, Omer Rana, and Ronan Reilly. "Predicting purchasing intent: Automatic feature learning using recurrent neural networks." arXiv preprint arXiv:1807.08207 (2018).

Author

Supervisor

Chapter 1

Introduction

"While the massive amount of data collected from selling online on numerous platforms can certainly create a challenge, moving forward, it is one that online merchants will need to handle effectively. Learning to translate that data into actionable information for driving future customer engagement could prove to be a significant asset." (Brett Relander)

Over the past two decades, the evolution of the hardware and the software led to the rise of the internet. Based on OECD (2008) and Group (2018) statistics the percentage of the people with the access to the internet, has risen from 57% in 2007 to 87% in 2018 in the OECD countries. The rise of the internet is directly correlated to rise of the e-commerce business. Currently there is rise in online marketplaces and tremendous growth of online advertising and marketing. It is not a coincidence, that the biggest company in terms of market capitalization is the internet retailer Amazon (as of 04/2020).

In today's fast paced interconnected world there is a big demand for the integration between offline and online point of sales – Retail 4.0. In order to succeed, retailers around the world have to provide tailored solutions to their customers to ensure their loyalty. This approach is progressively demanding the accurate prediction of consumer's purchase intention which leads to effectively targeted marketing campaigns.

From perspective of the online seller it is important to know which web-sites visitors are the most probably going to buy something and what do they want. Based on that information, tailored marketing strategy can be developed to address customer's needs and increase sales. This information can be gathered by latest methods in statistics, econometrics and machine learning based on the

behavior of the customer on the website. Recently Bag *et al.* (2019) used linear regression and neural network analysis in order to correctly predict purchasing intent in camera e-store. Sheil *et al.* (2018) utilize use of neural networks to find out which customers are going to buy goods from online store.

The aim of this work is to broaden the horizons of the current best practices in the machine learning field. Two models will be introduced in this thesis. The main model predicts purchasing intention of the customers of the e-commerce website. The model is further tested based on two possible practical applications based on business needs. The second model predicts whether the customer will leave the website in next few clicks. This enables for the e-commerce website to distinguish between shoppers which should increase sales a thus profits. Furthermore, the combined solution is presented, which utilizes both models achieving synergistic effects.

The thesis is structured as follows: Chapter 2 gives comprehensive overview of the available literature concerning topics of machine learning and e-commerce, Chapter 3 is focused on the data set itself. The data is categorized in the main categories and key statistics are presented in clear way. Chapter 4 is focused on the methodology of the newest methods in machine learning namely the Artificial neural networks and word embedding. Chapter 5 presents data preparation for the analysis taking into account the methodology and the best practices in the field. It includes missing variable treatment and feature engineering. Chapter 6 contains models that predict purchasing intent from most simple to the most complex. Chapter 7 develops secondary model which predict whether a visitor of the e-commerce website will leave the website in next few clicks. Chapter 8 shows practical examples of possible implementation of the model predicting purchasing intent and the model predicting exit. Lastly Chapter 9 summarizes findings of the thesis.

Chapter 2

Literature review

"Later perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech and writing in another language, it was predicted." (New York Times 1958)

Artificial Intelligence (AI) is considered to be the most powerful and defining technology of the 21st century. If harnessed properly it can lead to vastly more productive and efficient economy. It already surrounds all of us in some form. The biggest tech companies like Google, Apple, Microsoft and Amazon invest heavily in development of the Artificial Intelligence because of its great potential to increase their profitability. It is widely used in transportation, medicine, electricity distribution, banking, e-commerce and many other industries. Important part of AI field are Artificial neural networks (ANNs).

The first use of ANNs dates back to 1958 when Rosenblatt (1958) published his paper about the Perceptron which is considered to be the first one layer Artificial neural network (ANN). The Perceptron project was funded by the United States Office of Naval Research. Its primary objective was image recognition. Rosenblatt was very proud on his invention and praised its potential on many press conferences. This led to controversy in the AI community Olazaran (1996).

Significant part of the AI community - symbolic AI branch, believed that further progress in ANNs or the Perceptron was not possible therefore it would be illogical to devote scarce resources developing this concept further. The main researches of symbolic AI Marvin Minsk and Seymour Papert wrote their famous book Perceptrons. In the book they proved impossibility of the single

layer ANNs to learn XOR function[1] Minsky & Papert (1969). It was falsely believed that same limitation were also true for multiple layer ANNs and generalization of the Perceptron algorithm (ANNs) to multilayer architecture was not feasible. Nevertheless the controversy led to significant drop of interest in ANNs Olazaran (1996).

Almost 20 years later ANNs were revived after paper published by Ackley *et al.* (1985) where authors overcome conjecture by Minsky. One of the key reasons of the revival of the ANNs was cheaper and more available computing power.

The availability of relatively cheap computing power lead to the other great invention, the Internet. In the begging the Internet was also funded by United States Department of Defense and was used primary for the scientific purposes. The Internet began to be more widespread around early 90s when English computer scientist Timothy John Berners-Lee invented the World Wide Web which is used today. The rise of the Internet led also to the rise of the e-commerce.

Many traditional retailers began to offer their products on the Internet. The rise of the internet led to the new business like Netflix or Amazon which supplied goods and services only via the Internet. As it is almost cost less for the Internet users to visit e-commerce website, many online retailers had issue with too much traffic on their websites but with relatively small conversions rates[2]. As the resources are scarce companies had to be creative to prioritize among users. Victoria Secret redirected buying visitors to the faster servers while browsing users were hosted by slower and more congested ones Quick (1998). Purchasing customers were identified after they placed desired item in the cart. But this practice leads to bad shopping experience for all users even those who are going to buy from the store.

One of the papers tackling this issue was published by Moe & Fader (2000) where they developed model for predicting each customers probability of purchase based on observed history of visits and purchases. They defined four roles of e-commerce website visit: directed purchase visits, search/deliberation visits, hedonic browsing visits, and knowledge-building visits. The model allowed for heterogeneity and not-stationarity of the customers. Furthermore Moe & Fader (2001) developed ideas about differentiation of the customers, efficient

[1] XOR is an exclusive disjunction.

[2] Conversions rate is a percentage of customers to the website who complete transaction.

marketing strategies and importance of the conversion rate of the costumers. An e-commerce store can have increasing traffic (more visitors per day) which is good but with overall decreasing conversion rates can be a signal for management to change their marketing strategies as higher traffic and less purchases translates into higher costs per purchase.

Later Mandel & Johnson (2002) discussed ideas about design of the websites and their effects on customer preferences. They experimented with different website design priming product attributes for novice and expert test subjects. The priming had effect on both categories but in the different way. Novices were affected though change in their external search which caused change in preference. Expert's external external search remain the same but their preferences changed suggesting their internal prioritization between attributes changed. This study showed that people create their preferences while shopping online therefore more customization can possibly increase sales.

Later Bucklin & Sismeiro (2003) and Moe (2003) argued about importance of the behavioral click stream data for predicting customer's intentions. Bucklin & Sismeiro (2003) is first to analyze click stream data with respect to time. They used type II Tobit model on website data from automotive industry and modeled probability for visitors to continue browsing and length of time spent on pages. They found that repeated visitation means fewer page visits with no effect on duration. Moe (2003) analyzed click-stream data to categorize visits according to their intent. To do so she firstly categorized each page viewed into three categories: product, category and information page. During each session customers click on all types of the pages but their proportion differs based on the intent. Author used K-means in order to cluster customers into four types: buying, browsing, searching and knowledge-building based on their expected intentions on website. Click stream data became key to understanding shopper's behavior online.

Montgomery *et al.* (2004) expanded click stream shopping data research by adding memory component to the model. This enables to benefit from sequential information from the click stream data. Studies before used aggregated data where information about path was lost. Authors compared several methods including Markov Chains, Latent Class, Vector autoregression and multinational Probit model on bookstore's data (B&N bookstore). They achieved best results with Probit model.

Rohm & Swaminathan (2004) studied e-commerce from point of view of the customer. They performed mailed survey to analyze motivation for online

shopping. Other scholars focused more on automatic recommendation systems based on customer behavior Zhang & Jiao (2007) and Lazcorreta *et al.* (2008). To maximize sales it is not only important to know which e-commerce visitors are willing to buy something but also why are visitors leaving website, especially those who are buying ones. When item is put into shopping basket there is big probability that customer is willing to buy it. Nevertheless there are cases when website is abandoned when items are in shopping cart. Rajamma *et al.* (2009) conducted survey where they found out that main reasons for shopping cart abandonment are perceived risk, transaction inconvenience and waiting time.

When retailer wants to efficiently target customer on their website it is insufficient to know only who is willing to buy something but also to know probability of them leaving the website. This approach can help prioritize among customers and give customized promotions to the customers with highest probability to buy something but also with highest probability to leave website. This approach was tested by Sakar *et al.* (2019). They predicted purchasing intent with Multilayer perceptron and website abandonment likelihood with Recurrent neural network (RNN). Their study was conducted on aggregated data from Columbia e-shop in Turkey. In order to achieve best feature selection for the model Sakar *et al.* (2019) tested different filter methods for predicting purchasing intent on website. They achieved the best results with Minimum Redundancy Maximum Relevance feature selection method (mRMR).

Different methods for data mining were discussed by Carmona *et al.* (2012) who used k-Means, Apriori algorithm and NMEEF-SD algorithm to extract useful information from an e-commerce website selling extra virgin olive oil called www.OrOliveSur.com. As the machine learning became more and more popular among scholars and practitioners Pai *et al.* (2014) explored k-nearest neighbors (knn), Hidden Markov Model (HMM), support vectormachine (SVM) and random forest. The knn achieved significantly better results compared to other methods. Once again authors conclude that click stream data is useful in predicting purchasing intent and provide better predicting capabilities than demographic data. The knn method is considered to be lazy machine learning method because it does not generalize training data set into a function but "memorizes" the training data set. Therefore each time prediction is made knn compares input to all data in training set which is very demanding on processing power during implementation of the solution.

Importance of prediction of purchasing intent became so important that

there were two similar challenges on Kaggle website with hefty price pools. Although winning solution for predicting purchasing intent competition: RecSys Challenge 2015 by Romov & Sokolov (2015) used Gradient Boosting Machine method with extensive feature engineering, Sheil *et al.* (2018) analyzed data sets from RecSys Challenge 2015 and Retail Rocket challenge, applied RNN without extensive feature engineering and achieved better results compared to the winning solution of the Retail Rocket challenge and almost as good as the winning solution of the RecSys Challenge data set. The idea to use RNN for purposes of purchasing intent predictions was firstly explored by Toth *et al.* (2017) who compared RNN to the mixture of Markov chain models.

In recent years there are many researchers experimenting with hybrid neural networks, combining different types of layers ANN together in one model. Significant improvement in terms of accuracy was in Image Captioning[3], Text-to-image[4] and Visual Question Answering[5]. In terms of Image caption there is interesting work by Xu *et al.* (2015) where authors combined Convolutional neural network (CNN) with RNN. Similarly Hu *et al.* (2016) combined emebedding layers with CNN and RNN to achieve state of art results in Text-to-image. Recently Xie *et al.* (2017) and Gao *et al.* (2018) achieved excellent results in Visual Question Answering using novel architectures of Hybrid Convolution neural networks.

In conclusion there is strong evidence of usefulness of predicting purchasing intent. Related work suggested usefulness of click-stream data and some frameworks to analyze them. The best results were achieved by RNN but there are relevant arguments for use of hybrid approach combining multiple layers of different types of ANN.

[3]Image Captioning is process of generating text description of an image.

[4]Text-to-image is method of retrieving images based on textual search queries.

[5]Visual Question Answering is a process of answering question based of images, for example: "What color are her eyes?"

Chapter 3

Data

"There are three kinds of lies: lies, damned lies, and statistics." (Mark Twain)

The Data part of the thesis presents general overview of used data set for the thesis. In order to perform more complicated analysis it is necessary to understand data in question. In the beginning of the section there are general information about availability of data sets and general information about used data set. Moreover there are key performance indicators which are necessary for performing the predictions in later chapters.

3.1 Availability of data

Availability of relevant data for the analysis was a big issue. In studies concerning related topics authors usually had access to data of the e-commerce website that are not publicly available (Moe & Fader (2000), Moe (2003), Pai *et al.* (2014) and Sakar *et al.* (2019)). Those studies provided only some aggregated summaries of the data sets due to confidentiality issues. Therefore recheck of the results is almost non-existent. Others used data sets from past Kaggle[1] competitions (RecSys Challenge 2015 and Retail Rocket challenge (Romov & Sokolov (2015), Sheil *et al.* (2018)). The data sets from competitions provided only a few variables to work with and there were already sufficient number of studies analyzing them from different perspectives.

For the purposes of the analysis the data from Google Analytics sample data from the Google Merchandise Store will be used. The data set is relatively

[1]Kaggle is a website where data scientists can share, collaborate, analyze and explore data sets. Kaggle partners with many firms and publishes commercial datasets, competitions and challenges with hefty price pools for the community. It is subsidiary of the Google.

recent and is freely publicly available. The Google Analytics data set is used primary to promote services from Google. It provides huge variety of different features that should be useful for analysis. To the best of my knowledge data set from Google Analytics was not used for purposes of predicting purchasing intent.

3.2 General overview

Google Analytics is a free web analytic services offered by Google. It launched in 2005 and is offered since. It tracks huge amounts of data and can provide useful insights about website's visitors or potential customers. It provides many different statistics and reports. As of 2019 it was the most used analytic web analytic service on the web SimilarTech (2019). The data are accessed through reports or API[2].

The data set contains data from Google Merchandise store which is an real store that sells branded merchandise from Google and its subsidiaries (Android, Google Cloud, YouTube and Waze). The store operates around the globe in more than 200 countries. Google Merchandise Store sells apparel, drinkware, bags, notebooks, pens and others. The sample data set provided by Google contains one year of data from 01.08.2016 to 01.08.2017. The data set contains 714 167 unique visitors which translates into unique 902 755[3] of sessions[4] meaning that some visitors visited the merchandise store more than once. Each season has more than 300 unique features, but some of them are deprecated. The features are categorized into three groups: traffic, behavioral and transaction. The full list of all variables is available on *Google Analytics features link* and the used variables are present also in Appendix B.

3.2.1 Traffic features

Traffic features include information about origin of the user. The most important features are geographical data (continent, country, region, city), origination information about visitor (organic, cpc[5] , referral), website from which

[2]Application programming interface

[3]The data set has 903 653 rows but some sessions are cut into two because they start at the end on one day and end on the next day.

[4]A session is a group of specific user interactions on website taking place within a given time frame. For more information: *link*

[5]Cpc is cost per click.

he/she originated (google.com, youtube.com, etc.), keyword used in search to get to the store (YouTube Merchandise, google clothing store, google hoodie, etc.) information about Google ad-group (Google Search, Content, Search partners, etc.), device (Mobile, Tablet, Desktop, etc.), internet browser (Chrome, Edge, Firefox, etc.) operating system (Macintosh, Windows, Chrome OS, etc.).

3.2.2 Behavioral features

Each distinct session (902 755), defined by combination of unique keys fullVisitorId and visitId, has at least one hit[6]. Each hit represent different sub actions that visitor performed on the website. It includes hit number, hour, minute, number of milliseconds from start of the session, action type (Browsing, Click through of product lists, Product detail views, Add product(s) to cart, Remove product(s) from cart, Check out, Completed purchase, Refund of purchase, Checkout options), hit type (PAGE, EVENT) information about latency tracking (lookup time, loaded time, active time etc.), screen name (www.googlemerchandisestore.com/home), information about social networks (network passed with the social tracking code), information about promotions.

3.2.3 Transaction features

When session includes buying event (11 515) there are information about the transaction from Google Merchandise store. The most important features are product name , product category, product price, product quantity, total revenue, information about shipping and information about tax.

3.3 Key performance indicators

To better understand basics of the data set some key statistics is presented in this subsection. The selection of the indicators is based on best practice in the industry, literature review and Google Analytics documentation.

3.3.1 General overview

General overview is based on aggregated indicators provided by Google Analytics described as totals. Based on much higher mean than median in majority of the cases there is a stark difference between sessions with and without buying

[6]Used extensively in literature and Google documentation as representation of one click.

event. The statistics are present in the tables Table 3.1 and the Table 3.2 where n_hits is number of clicks, pageviews represents number of distinct page visits, timeOnSite is time spend on the e-commerce website, totalRevenue represents total revenue from one user in US dollars, transactions number of transactions and visits number of visits.

Should we compare the Table 3.1 and the Table 3.2 there is an significant difference between basic statistics of whole data set and part of data set with buying event is partly caused by high (49.92%) bounce rate[7]. It means that user enters web page but does not interact with web page. For example bounce could be triggered by simply returning to search engine after quick glance on the website. The majority of bounces have only one hit (there are some with 2) with mean value of timeOnsite 0.002 seconds while majority of observations do have zero.

Table 3.1: Basic statistics

	mean	std	min	25%	50%	75%	max
n_hits	4.6	9.7	1.0	1.0	2.0	4.0	500.0
pageviews	3.9	7.0	0.0	1.0	1.0	4.0	469.0
timeOnSite	131.5	368.3	0.0	0.0	1.0	84.0	19017.0
totalRevenue	2.0	83.2	0.0	0.0	0.0	0.0	47082.1
transactions	0.0	0.1	0.0	0.0	0.0	0.0	25.0
visits	1.0	0.0	1.0	1.0	1.0	1.0	2.0

Table 3.2: Basic statistics (Buying event only)

	mean	std	min	25%	50%	75%	max
n_hits	36.4	30.3	2.0	19.0	28.0	44.0	500.0
pageviews	28.4	21.7	2.0	16.0	23.0	34.0	469.0
timeOnSite	1065.6	952.0	9.0	459.0	777.0	1359.0	15047.0
totalRevenue	154.6	720.5	1.2	30.0	55.6	116.6	47082.1
transactions	1.0	0.5	1.0	1.0	1.0	1.0	25.0
visits	1.0	0.1	1.0	1.0	1.0	1.0	2.0

[7]Bounce rate is percentage of sessions that are single-page sessions.

3.3.2 Geographical analysis

Google Merchandise store sells goods around the world. There are visitors from more than 200 countries where in 68 countries there is at least one buying session. The distribution of visits, buying visits and revenue is not uniform. As you can see on the Table 3.3 the majority of revenue comes from the USA. It is interesting that Venezuela has much less buying sessions than Canada but has similar total revenue. This is caused but rather hefty purchases in Venezuela. The first three countries account for more than 97% cumulative percentage of revenue. The prediction analysis will be done only on visits from the USA, Venezuela and Canada due to low number of buying events in other countries which could cause problems for classification.

Table 3.3: Geographical analysis

	country	nObs	n_hits	totalRevenue	n_buying_ses	cumul_%
0	United States	364402	2483593	1664261	10953	93.5
1	Venezuela	2129	21056	36082	63	95.5
2	Canada	25850	159092	34922	190	97.5
3	Japan	19670	74251	7629	17	97.9
4	Kenya	771	2472	5286	3	98.2
5	Nigeria	1445	3718	3314	2	98.4
6	Indonesia	9266	24164	2678	11	98.5
7	Taiwan	12950	55745	2016	19	98.7

3.3.3 Cart abandonment

Shopping cart abandonment is a situation when visitor adds item into a shopping cart thus indicates intention to buy an item but he/she leaves an e-commerce website without order finalization. Therefore any shopping cart that was filled with items but the items are not bought is considered abandoned. This metric is considered to be very important as mention before. Rajamma *et al.* (2009) conducted a study when they analyzed reasons for shopping car abandonment. High abandonment rate could signal poor user experience from checkout. It can be caused by several reasons, for example misleading and complicated checkout, expensive shipping, untrustworthy payment methods.

Cart abandonment on Google Merchandise store was approximately 80%. The calculation is defined as all unique sessions that are not buying and have at

least one event 'add to cart' divided by total unique sessions that have at least one event: 'add to cart'. In the data set not all buying sessions have event 'add to cart' (1 440 out of 11 515). This is partly caused by the fact that someone can visit store twice, in the first session he/she adds items into cart and in second session he/she finalizes transaction. Also it might have been possible to buy items directly without need to add them first into shopping cart. There is also possibility to visit store from different devices (resulting in two different ids), on the one device log in in the account, add items to the cart and on the another device log in and finalize transaction.

3.3.4 Time analysis

To understand shopping intentions it is important to know shopping patterns in the data set with respect to time. It is usual that sales are driven during weekends when people have more time. In case of the Google Merchandise store it is the other way around. Should you see the Table 3.4, the most visits, revenue, hits happened during working part of the week. It is possible to some extant that some of the customers of the shop are employees of the Google who buy staff for their clients as a gift.

Table 3.4: Weekday analysis - averages

weekday	nObs	nHits	Revenue
Monday	2659	12491	5133
Tuesday	2779	12915	5801
Wednesday	2822	13238	5740
Thursday	2738	12747	5176
Friday	2468	11594	4974
Saturday	1878	7981	1123
Sunday	1930	8424	1458

3.3.5 Conversion rate

Conversion rate is percentage of all unique visits resulting into transaction. The importance of the indicator was discussed by Moe & Fader (2001). The traffic on e-commerce website should not be considered alone, without conversion rate. The increase in conversion rate allows eshops to increase sales with same

amount of traffic. The conversion rate on the Google Merchandise store is 1.27% which is relatively small. Should we consider only not bounce sessions then the conversion rate is about twice as large (2.54%). Should we taken into account only session that originated from the USA, Canada and Venezuela the overall conversion rate is 2.86%.

3.3.6 Path analysis

To predict purchasing intention it is important to understand motives behind e-commerce website visit. The importance of the click-stream behavior data and path analysis was firstly argued by Moe (2003). The variable action_type provides general information about behavior of the visitor on the e-commerce website. The analysis was performed on all sessions that are not bounce. While mere browsing doe not include much interaction of a visitor on website, actions: product list, product detail and add to cart include more interaction of the visitor on the website. The ratio of interaction with website for buying sessions is much higher than in case of non buying sessions. The difference can be seen clearly on the Table 3.5.

Table 3.5: Path analysis

	browsing	product list	product detail	add to cart
All obs	0.881	0.058	0.048	0.013
Buying	0.680	0.120	0.104	0.096
Not buying	0.886	0.056	0.047	0.011

Unfortunately categorization of websites to product lists, products, cart and page information are not available in the data set. This categorization will be created based the web page paths. As suggested by Moe (2003) proportions of these types of views should also be very useful in classification of sessions.

Chapter 4

Methodology

"Real stupidity beats artificial intelligence every time." (Pratchett 2008)

The Methodological part of the thesis describes in greater detail methods or tools used to perform analysis. In the beginning general concepts of the Neural networks are explained. As the chapter progresses more advanced method are described with focus on Recurrent neural networks and their specification. In the end of the chapter word embedding is described in general matter.

4.1 Artificial neural network - general model

Neural networks are a basic element of deep learning, a sub field in machine learning. The algorithms in machine learning are inspired by functions of human brain. The basic concept of Neural networks is that neural networks takes input, analyzes the input, finds patterns in the data and gives us output. Firstly the Neural network is trained on training set and then it can give us predictions of output on similar data sets. Similar to the human brain when you want to learn something new.

To be more precise neural network is constructed from layers of artificial neurons. The neurons process data. Each neural network has to have an input layer where a input is fed into a neural network and an output layer where an output is predicted. Inputs are fed to the neurons at the input layer (x_i). Each neuron in the input layer is connected to neurons in the hidden layer though channels. Each channel is assigned a numeric value (w_j[i,a]) a weight. Values from input neurons are multiplied by weight and their sum is fed to the neurons in the hidden layer(initial values of (w_j[i,a]) and b_a are selected

randomly). Each neuron in hidden layer is assigned a bias b_a which is added to the input sum. The sum (see Equation 4.1) is passed into an activation function which determines whether the neuron is activated or not.

$$(x_1 w_{1[1,1]} + x_2 w_{1[2,1]} + b_1) \rightarrow Activation function \tag{4.1}$$

Activated neurons transmit data to next layer through channels with different set of weights. The input data is propagated through the network to the output layer in form of probability. This is called feed-forward propagation. During training of neural network, not only input is fed to the network but also an output which is compared to the predicted one and error in prediction is calculated via a cost function. The magnitude of an error signals to the neural network how wrong was the prediction.

The information about the error (magnitude and sign) is propagated back to the network. This process is also know as Back-propagation. Based on the information the weights and biases are adjusted to minimize the error. The adjustment is calculated by the gradients of the cost function with respect to weights and biases. Forward propagation and back-propagation is performed iteratively with thousands of inputs. A simple visualization is presented on Figure 4.1[1].

4.1.1 Activation function

An activation function in ANNs defines the output of the neuron based on the input. It is one of the essential parameters in ANNs. For each neuron in hidden layers of the ANNs the input is calculated as an output from previous layer multiplied by weights from channels plus the bias from the neuron as described in the Equation 4.1. The sum will be refereed as the summed activation. A linear activation, the simplest possible activation function, is a situation where there is no transformation of the summed activation at all. On the one hand linear activation is very easy to learn (during learning process first derivative-gradient is calculated to minimize the error) but on the other hand ANN is unable to learn more complex relationships between data. For more complex data structures, the non-linear activation functions are used. Probably the most used in literature and by practitioners are sigmoid and hyperbolic tangent activation

[1]Technical Note: All diagrams in Methodology section were created by author. The coloring of neurons is consistent - orange for input, green for hidden layer and blue for output. The arrows show direction of the data flow.

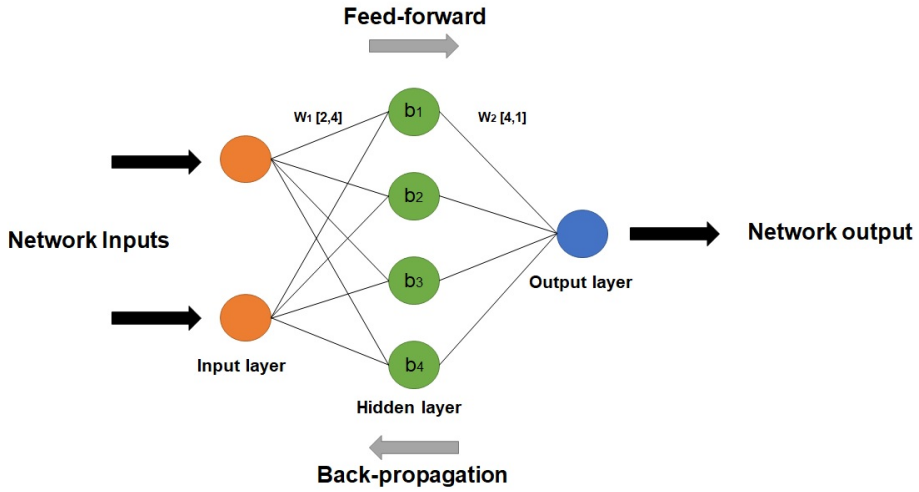

Figure 4.1: ANN diagram

The ANN diagram shows how is the input data fed-forward through hidden layer and how the network gives the output. The weights and are showed as ($w_j[i,a]$) and b_a respectively.

functions.

The sigmoid function also knows as "S"-shaped curve is defined by the equation 4.2. It transforms input into values between 0 and 1. The sigmoid activation function is one of the most used Sibi *et al.* (2013).

$$S(x) = \frac{1}{1 - e^{-x}} \tag{4.2}$$

The hyperbolic tangent activation function is defined as tanh(x) 4.3. It has values between -1 and 1. It generally gives better results as a sigmoid function. Karlik & Olgac (2011) analyzed performance of different activation functions and achieved the best results with tanh(x) function.

$$tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \tag{4.3}$$

The main limitations of sigmoid and tanh functions comes from their definition as very large inputs snap to upper bound (1.0) and very small input snap to lower bound (1.0; 0.0). The functions are sensitive only around middle (0.0). Some of these issues are solved by rectified linear unit (ReLU 4.4). The function is partly linear and partly non-liner. The benefits of ReLU function were argued by Glorot *et al.* (2011). The main benefits are simplicity to calculate first derivative and somewhat linear behavior. Even though for majority of uses is

the ReLU function efficient option, for RNN its use is quite limited because of very large outputs that can lead in explosion of gradient Le *et al.* (2015).

$$ReLU(x) = max(x, 0) \tag{4.4}$$

When setting up ANN it is possible to set different activation functions for different types of layer. Liu & Lane (2015) used the softmax function 4.5 for a multi-class classification problem - understanding spoken language using RNN. The softmax function takes as an input N real numbers and gives back N probabilities based on input numbers.

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{4.5}$$

4.1.2 Cost function

Sometimes refereed as a loss function, the cost function in context of ANN is used to evaluate vector of weights and biases. There are many possibilities for the cost function. The most commonly used cost function for classification problems is cross-entropy error function Reed & MarksII (1999). For regression problems it is mean squared error.

The Cross-entropy function 4.6 measures difference between two distributions for given input - random variable x, where p(x) is the true distribution and q(x) is the estimated distribution.

$$H(p, q) = -\sum_j (p(x_j) log(q(x_j))) \tag{4.6}$$

4.1.3 Gradient descent

The gradient descent is one of the most popular optimization algorithm currently used to to optimize weights and biases in ANNs. It utilizes training data, makes predictions, comperes predicted variable to real ones and adjusts the model to minimize an error. This process is done via minimizing cost function J(θ) (sometimes referred to as an objective function or a loss function) where θ is defined as a vector of all model's parameters. Therefore the gradient descent

is defined as ∇_θ J(θ). The gradient descent is described by formula 4.7 where η is a learning rate[2].

$$\theta = \theta - \eta \nabla_\theta J(\theta) \tag{4.7}$$

There are three main types of gradient descent mentioned in literature. The difference is in the amount of data used to calculate gradient leading to trade off between parsimony (computing power and time) and accuracy.

Stochastic gradient descent The Stochastic gradient descent (SGD) takes one example from the training data set and computes the gradient of the cost function, calculating one update at the time. This process is generally much faster compared to the Batch gradient descent but the frequent updates cause the cost function to fluctuate heavily.

Batch gradient descent The Batch gradient takes the whole training data set and computes the gradient of the cost function. Therefore compared to SGD it takes much more time and resources to compute one update of the parameters (weights and biases).

Mini-batch gradient descent The Mini-batch gradient descent is something in between of two extremes. It performs update of parameters per small batches. Ruder (2016) suggested that Mini-batch gradient descent can lead in reduction of variance and more faster calculation as not whole data set is evaluated.
The batch sizes are usually two to the power of x. Bengio (2012) recommends using 32 as good default value. The 32 was also suggested in newer study by Masters & Luschi (2018).

Even though the Mini-batch gradient descent is superior to other two, there are still some challenges to be tackled. The main challenge is defining a learning rate η so it is not too small or too high. Too high learning rate could cause high volatility or even to divergence Ruder (2016). In case of too small learning rate it could take too long to train an ANN. Additionally it can be quite challenging to minimize highly non-convex error function mainly around saddle points Dauphin *et al.* (2014). There are many gradient descent optimization algorithms to tackle issues about learning rate and finding minimum. Bengio

[2]Learning rate defines how much to change model's parameters taking into account estimated error

et al. (2013) advocates using Nesterov accelerated gradient algorithm which significantly improved RNN performance on a number of tasks.

The Nesterov accelerated gradient (NAG) is defined by the equation 4.8 where γ is a fraction of the last update used. NAG is based on the Momentum-Based Gradient Descent which lacks $\gamma update_{t-1}$ term in objective function J. Generally momentum algorithm takes advantage from past changes in parameter. Furthermore NAG takes into account prediction of future θ the $(\theta - \gamma update_{t-1}$) which allows to partially correct overshooting.

$$\begin{aligned} update_t &= \gamma update_{t-1} + \eta \nabla_\theta J(\theta - \gamma update_{t-1}) \\ \theta &= \theta - update_t \end{aligned} \tag{4.8}$$

4.1.4 Number of layers and neurons

ANNs have two main parameters that define structure of the network the number of layers and the number of neurons in each layer. As showed on the Figure 4.1 each ANNs has an input layer, an output layer and a hidden layer. The number of artificial neurons in the input layer is defined by the number of inputs. In case of the output layer it depends on the output. In binary classification problems or regression problem there is only one neuron in output layer. In case of Multi-class classification problem there should be one neuron for each possible class.

The hidden layer/s are more complicated. Even though there some studies conserving analytic way to determine optimal number of hidden layers and neurons in them for example Stathakis (2009), the most used method is systematic experimentation. Even though Sheil *et al.* (2018) analyzed also e-commerce data sets, the number of input features was much lower than in case of the Google Analytics data set. Author experimented with different number of layers and neurons. The best results were achieved by three hidden layers and 256 neurons.

4.1.5 Exploding and Vanishing gradient

When training an ANN in each step the gradient of the error function is calculated. Based on the gradient the weights and biases are adjusted. In very deep neural networks or RNNs the gradient can accumulate during calculation resulting into large gradients which translates into large updates of the network. Large updates can cause instability of the network.

The vanishing gradients problem refers to the opposite issue, when gradient is too small making it impossible for the model to improve performance over training as described by Pascanu *et al.* (2013). The issue of the vanishing gradient can be seen on the graph 4.2 bellow where the Sigmoid activation function with its first derivative is presented. If the input for the activation is too large or too small the first derivative (gradient) is close to zero thus an ANN does not learn.

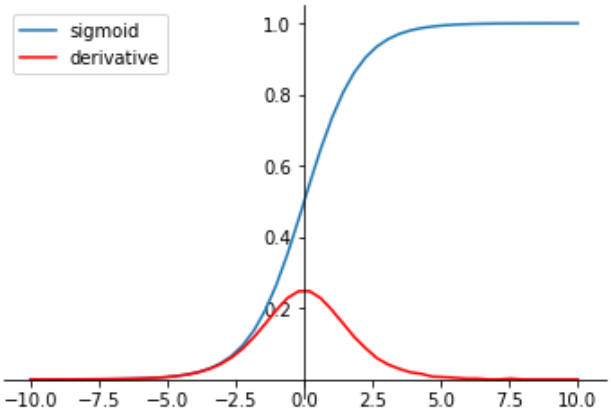

Figure 4.2: Sigmoid function

The Sigmoid function diagram shows plotted sigmoid function with its first derivative (gradient). The main take away from this figure is that for values higher than five or smaller than negative five the gradient is very close to zero meaning that weight/s do not get updated which means that ANN will not learn.

4.2 Recurrent neural network

The RNN is a class of ANNs where an input is not fixed. The RNN takes as an input a sequential data. While it is possible do define standard ANNs in such way that lagged variables are added to the model, the model has to have fixed amount of input variables and can not utilize predictions from past realizations. This means that RNNs remember patterns from past inputs while calculating present outputs which is proved to be useful in numerous fields, for example translations, speech recognition, predicting future changes in stock prices and many others.

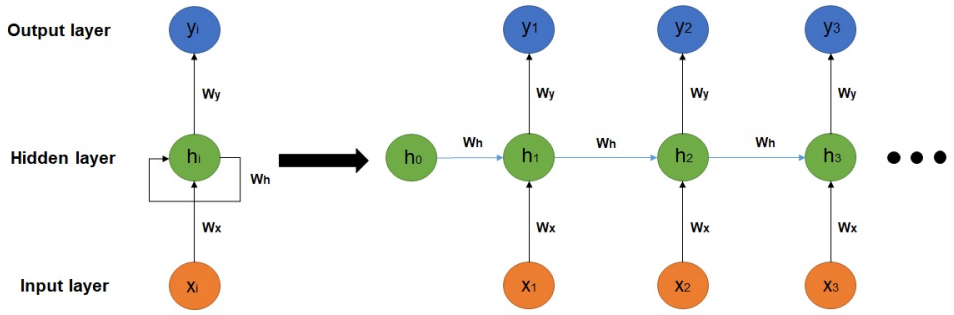

Figure 4.3: RNN diagram
The RNN diagram show basic logic behind the RNN. The connection between past inputs and the output is the main advantage of the RNN.

The RNN takes as an input one or more input vectors and is able to produce one or more output vectors. Outputs are calculated based on weights similarly to ANNs but also by hidden state vector (h_i) which represents information from past data realizations. The basic logic is described on the Figure 4.3. The backward arrow on the left side of the Figure represents the idea about how the information from earlier steps is used later in the system. It is important to point that, the RNN does not change between the time steps, the same weights are used for each time step. "RNNs, once unfolded in time, can be seen as very deep feedforward networks in which all the layers share the same weights" (LeCun *et al.* 2015). The hidden state h_i is shown in more detail in the Figure 4.4. The RNN usually uses tanh function as an activation function.

The RNNs are very powerful ANNs but their training can be problematic. As showed by Gers *et al.* (1999) and Bengio *et al.* (1994) RNNs are problematic to train in cases where input sequences are long. This means that RNNs have issue to store information for longer periods. Also RNNs are prone to either vanishing, exploding gradient Graves *et al.* (2008). In order to correct such

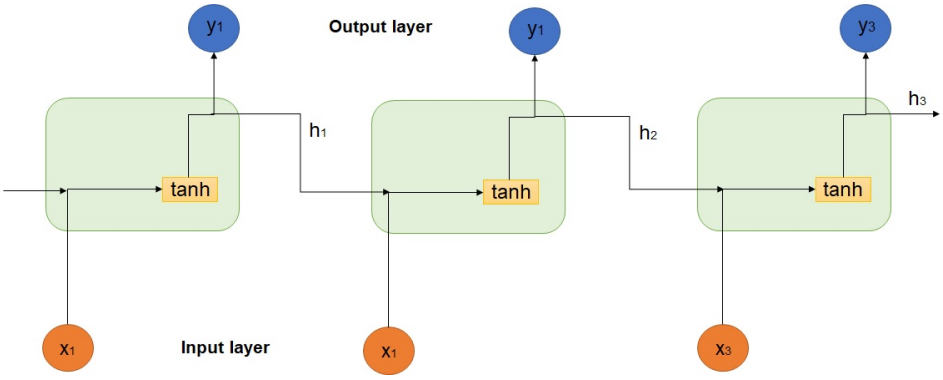

Figure 4.4: RNN detail

The RNN detail show more precise representation of the RNN. The logic behind the RNN is further developed with the LSTM were the structure is build on the RNN but with more sophisticated manner.

issues Hochreiter & Schmidhuber (1997) proposed different architecture with explicit memory called Long Short-Term Memory (LSTM).

4.2.1 Long Short-Term Memory

The Long Short Term Memory networks or LSTMs are more complicated RNNs. The basic concept of the LSTM is described on the Figure 4.5. Compared to the RNN Figure 4.4 the LSTM has four neural network layers which interact with each other[3]. The most important part of the LSTM is the Cell state vector. The Cell state runs through the whole LSTM and passes past information to the future. The LSTM has ability to add new information to the cell state (input gate) but also to forget information via forget gate.

The forget gate takes as input hidden state h_{i-1} and input x_i changes the input to values between 0 and 1 via sigmoid[4] function. The output of the sigmoid function is multiplied by cell state C_{i-1}. Therefore variables that are no longer needed are forgotten.

Next the input gate (based on hidden state h_{i-1} and input x_i) decides what new information is added to the Cell state C_i. The tanh function creates new candidates for cell state, C_i* and again sigmoid function decides which are updated and which not. Then the new information is added to the cell state C_i.

Lastly there is the output layer which takes updated cell state C_i transforms

[3]RNN has only one

[4]Generally the sigmoid function is used to decide what variables to keep and which to forget. For example if the element in C_{i-1} is multiplied by zero it is forgotten, if by one it is remembered

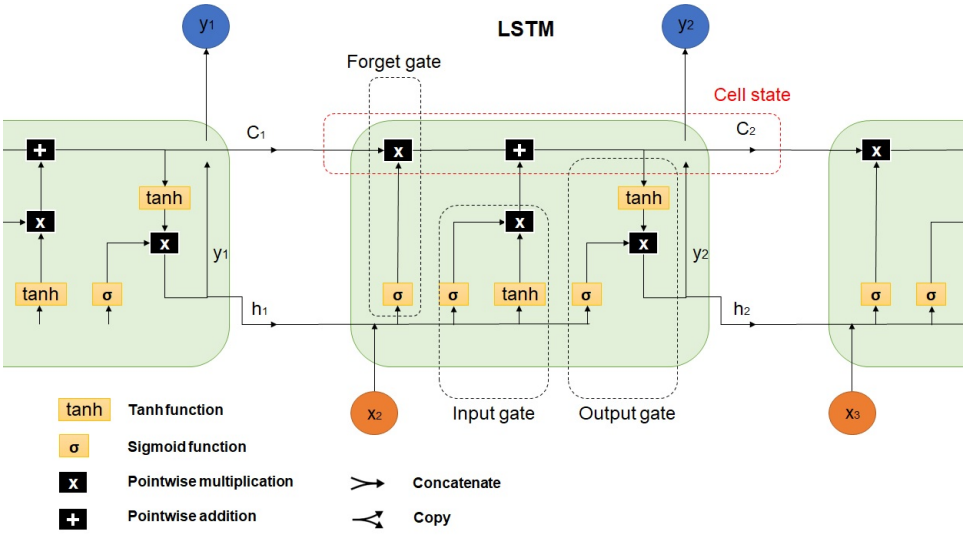

Figure 4.5: LSTM

The LSTM figure show comprehensive detail of the artificial neuron of the LSTM neural network. The main logic is the same as in the case of the RNN with addiction for LSTM to have ability to remember and forget past inputs.

it via tanh function and again sigmoid function decides which elements are kept and which not. The output layer creates hidden state h_i and output y_i.

Input and Output shapes in LSTM

There are many possibilities with input and output shapes in LSTM. Compared to the other types of the ANNs which accept fixed-sized vector as input and produce a fixed-sized vector as output the LSTM operates in more general settings. It accepts sequences of vectors in the input, the output, or in the most general case both. All possibilities in terms of input shapes and output shapes are represented o the Figure 4.6 bellow.

The sequence input/output framework is much more powerful compared to fixed networks. In case of the Google e-commerce click stream data LSTM provides benefits in terms of possibilities to learn patterns between clicks which vary in number. If instead of the LSTM other type of the ANNs was used, the aggregation of the data based on the primary ID have to be performed which inevitably leads to loss of important information from click stream data. The importance of click stream data was argued by Bucklin & Sismeiro (2003) and Moe (2003).

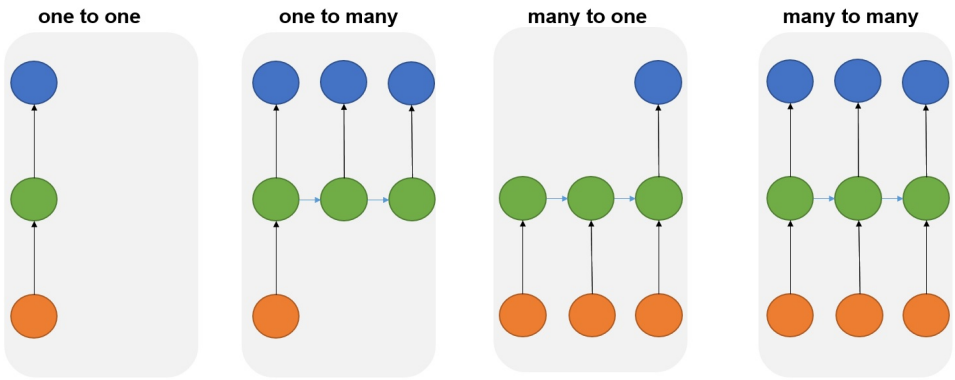

Figure 4.6: LSTM shapes
The LSTM shapes shows all possible set ups of LSTM in terms of input and output shape of the network.

Python Keras - tensorflow framework

The TensorFlow is one of the most popular machine learning frameworks, created by the Google Brain team. It is build on high-performance C++ providing seamless and efficient calculations. The biggest advantage of TensorFlow is abstraction. The researcher can solely focus on overall logic of the model while Tensorflow deals with all complicated math behind the scenes optimized to the limits.

The analysis of e-commerce data set is performed in python environment using popular Keras package. Keras is a high-level neural networks API, written in Python. It is able to run on TensorFlow, Theano and CNTK backend. It was developed to be easy to use with focus on fast experimentation which is ideal for scientific research. Being able to go from idea to result with the least possible delay is key to do good research. It enables to create combinations of different types of neural network for best possible outcomes. It optimized for CPU calculation as well as GPU. Keras provides a convenient front-end API for building models, while those models are executed in back-end in TensorFlow.

Keras LSTM settings

The input layer of Keras LSTM accepts only input in form of 3 dimensional array where the first dimension represents number of batches, the second number of time steps and the last is number of features. The Keras enables to set first two dimension to be not fixed. Only dimension that has to be fixed is number of features. This is ideal settings for click-stream data sets where number of time steps (in this settings number of clicks/hits) vary. The output

shape is defined by parameter return_sequences. When set to false returns LSTM outputs 2 dimensional array, and when to true it outputs 3 dimensional array.

In order to LSTM work properly it is necessary to set statefulness of network correctly. Statefull LSTMs are used when data in batches are highly related, stateless otherwise. For e-commerce data set stateless LSTM will be used.

Moreover as shown Data section Google e-commerce data set is unbalanced, meaning there is approximately 97% of negative events and only 3% of positive. In order to prevent ANNs to set all observations to 0 (not buying) achieving 97% accuracy it is necessary to balance data using balancing weights of observations for non and buying sessions.

4.3 Word embedding

A word embedding is basically learning representation for sentences where words that have the same meaning have a similar representation. Probably the most famous example in the documentation is mathematical representation of the word queen showed as formula 4.9 bellow.

$$queen = king - man + woman \tag{4.9}$$

Word embedding is data science techniques where individual words are represented as real-valued vectors in n dimensional vector space. Each unique word is mapped to the specific vector. The vector values are learned in similar way as ANN. Therefore word embedding is often put together with deep learning. Key to the approach is the idea of using a dense distributed representation for each word, leading to much smaller vector space compared to one-hot encoded where each word has its own dummy variable. The representation of the words is learned based the similarity of word usage capturing meaning of the words. The word embedding is based on distributional hypothesis which was proposed by Harris (1954). The main claim of the paper is that similar context implies similar meaning.

The embedding algorithms learns from fixed sized vocabularies which yield real-valued vector representation. There are two main approaches in learning word embedding. The key difference is you either create new word embedding along training your model or you take existing word embedding trained on much waster data set which is then fitted on your vocabulary.

Embedding Layer

An embedding layer is a word embedding that is learned together with ANN. It takes as an input matrix with words encoded in form as dummy variables. Then the model is set for number of dimensions of the vector space where words are embedded. The vectors are initialized with random numbers and trained via backprobagation in similar way as the rest of the ANN. If there are more layers after embedding layer in a ANN then it is necessary to flatten[5] output from embedding layer to fit into standard neural network. In case of RNN each word may be taken as one input in a sequence. This type of word embedding is reliquaries large data sets, lot of training and processing power but will be tailored to the needs of specific problem.

Word2Vec

Word2Vec is pre-trained embedding developed by Mikolov *et al.* (2013a) as a response to make word embedding for bigger data sets more efficient. It was trained on very large data sets, for example there is Word2Vec model trained on English Wikipedia with Corpus[6] size about 3.5 billion tokens[7], knowing 249 212 different English words. Additionally, authors analyzed learned vectors and explored math on the representations of words. Mikolov *et al.* (2013b) claim that "these representations are surprisingly good at capturing syntactic and semantic regularities in language, and that each relationship is characterized by a relation-specific vector offset. This allows vector-oriented reasoning based on the offsets between words."

[5]Process of concatenation of vectors to reduce number of dimensions
[6]Corpus is a large and structured set of texts.
[7]Token is a vocabulary ID.

Chapter 5

Data preparation

"Without a systematic way to start and keep data clean, bad data will happen."
(Donato Diorio)

In the Data preparation section the whole pipeline from raw data set to the final data set, that is fed into the models is presented. As mentioned in the Chapter 3 Data only sessions originating from USA, Canada and Venezuela will be analyzed. Before constructing main SQL Query for Google Analytics API, the quality of raw data set is assessed. As the the data set is very large, unnecessary parts are deleted and mistakes are corrected in most efficient manner. Lastly relevant features present in related work are calculated and explained in full detail.

5.1 Data quality

In the Google Analytics data set there are 331 unique features. Google provides documentation where there is information about all features, the documentation is available on *Google Analytics features link*. Based on documentation some of the features are deprecated and replaced with newer instances. The main issue in the data set is availability of features due to confidentiality issues. Those data points are marked as *not available in demo dataset* or *null*. In some cases there is combination of both. Furthermore, some features have very high missing rates which renders them unsuitable for model analysis. The most disappointing is very high missing rate in geographical features (approximately half) that more closely describe state or city of visitors. Those could have been used to add mood variables like sunshine or temperature in the state at specific

time to the the data set. It is not clear if information was not available or was deleted for sample data.

5.2 Missing variables treatment

All features in the data set were checked for number of distinct values. If the number was zero, meaning all observations in one column were null, then the column was disregarded. Should the column have only one distinct value, the column was checked further, because some boolean variables had value 1 for positive outcome and null for negative outcome which translates into one distinct value per column. In other cases all observations in the column were the same and so the column was disregarded.

In addition, analysis of remaining features was performed. All variables with prefix promotion, experiment and custom were disregarded. Experiment and custom variables had very high missing rates. The promotion features were unfit for the analysis. After generation of the data set with the promotion features the data set had much more rows than otherwise. This was caused by the relationship between promotions and hits. If the user clicked on page where promotions were available than that unique click was on multiple rows each raw containing different promotion. This causes too much clutter in the data set and there was no mention in related work about usefulness of such data therefore it was disparaged.

Product information

Similarly to the promotion features product features caused also duplicates in rows. As the product information is indeed useful in analysis the issue had to be treated. The root of this issues is system used by Google in general to save information from the pages.

When user searches for some text in the Google search engine, Google returns list of URL records or list of impressions (10 blue links) (see Figure 5.1). Even if the link is not shown on the screen, user usually does not scroll till the end of the page, the page saves all 10 links. The same logic is applied on e-commerce website. If user visits category pages where multiple products are present all product are attributed to that one distinct click even though the visitor have not seen them all. More information is available from official *Google documentation*. Therefore product information on the category pages

is disregarded. The product information is kept on product pages where only one product is present and there is very high probability that visitor clicked on the specific product with some intention in mind and that he/she has seen it.

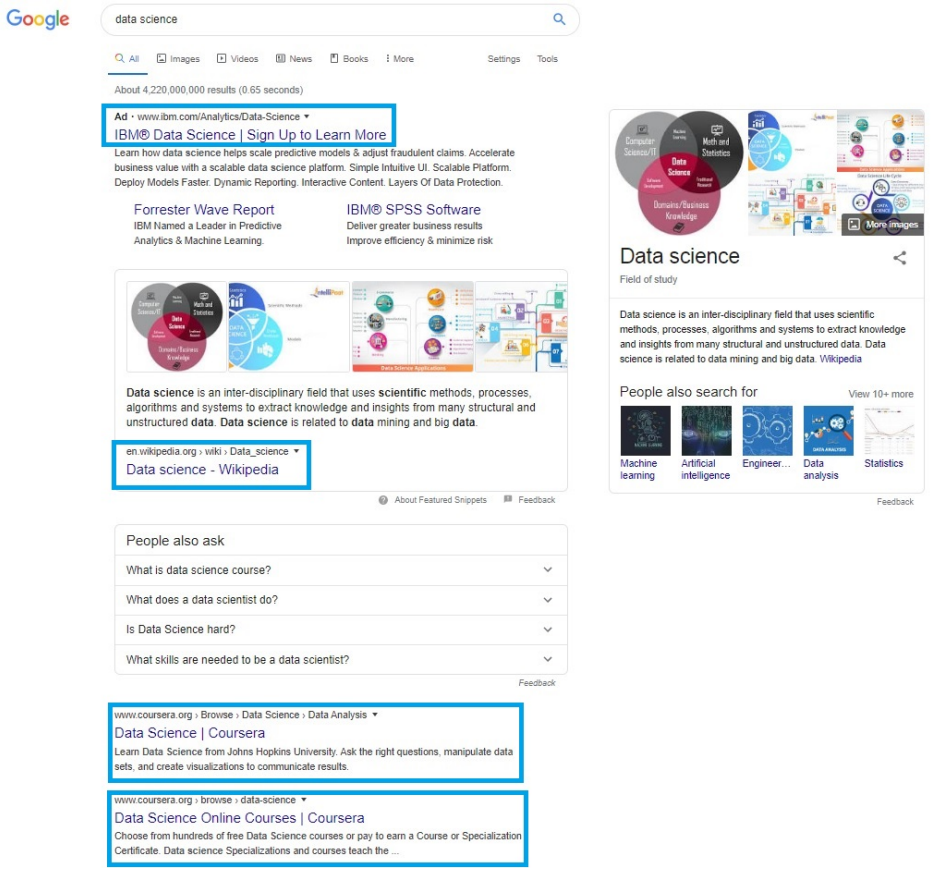

Figure 5.1: Google query

The Google query diagram shows list of impressions (marked in blue rectangles). All ten impressions on the Google query page are considered to be seen even though, a user have not scrolled down but he clicked on the second impression.

5.3 Feature creation

Theoretically very deep ANN should completely replace hand crafted feature engineering. As the the data progresses through the network, network chooses which features are relevant and which not, features are combined and proper weights are assigned. Whilst deep learning has simplified feature creation it still needs human to create proper architecture. Moreover in many cases the network that would be able to standalone create all necessary features would be so deep and complex it would not be feasible to train it. Therefore it should be useful to nudge it into the right direction.

5.3.1 Time features

As suggested by the analysis in Chapter 3 Data visitors behave differently based on the time. Related literature suggests that shoppers buy more stuff during weekends. The opposite is the case considering the Google merchandise store as majority of visits happens during business hours, probably because Google employees shop from the store to buy promotional gift for their business partners. In order to properly take into account this behavior three more features were calculated, day of week, weekday and holiday. The holiday feature is based on country specific holidays in USA, Canada and Venezuela. Weekday should help ANN better distinguish between working and non working days.

5.3.2 Behavior features from click stream data

Browsing behavior have been dived into two main categories, goal-directed search behavior and exploratory search behavior by Janiszewski (1998). Exploratory search refers to a consumer who is less focused and more random in behavior and perhaps not even considering to buy something. In contrast Goal-directed search refers to an situation when consumers use stored search routines to collect information in a deliberate manner. To sum it up there are two basic types of buyers goal-driven and stimulus-driven. The work of Janiszewski (1998) was further developed by Moe (2003) into more detailed characterization of buyers into Directed Buying, Search and Deliberation, Hedonic Browsing and Knowledge Building.

Directed Buying occurs when the shopper intends to make a purchase and is not lacking any substantial information and is very focused and targeted. Search and Deliberation occurs when buyer is goal-directed with a planned future purchase but needs acquire relevant information to help make choice. Hedonic Browsing is exploratory search behavior where utility stems from in-store experience and occasionally results in impulse buying. Lastly Knowledge Building does not intend to shop but his/her objective is to increase product and/or marketplace expertise with focus on informational pages.

To distinguishes between those categories of visitors Moe (2003) suggested following metric: Category Variety, Product Variety and Repeat Product Viewings. Based on those suggestions following features were calculated.

Type of the page

In order to calculate category and product related features it is necessary to know the type of currently viewed Google store page. As such feature was not available it was calculated from feature action_type[1], pagePathLevel1[2] and pagePathLevel2. The pages were categorized into four groups: info_pages, remove_product, product and category.

The distinction between product, category and information pages should be obvious. As the action removing from cart takes place in the cart pages it is clear it is not product page nor category page and not even information page where basic information are stored. Generally all interactions with cart (action type 5,6) are disregarded in training or testing ANN, action type removing item from cart was some special category in between standard shopping and interaction with cart it was given special category and was further analyzed in next section.

Product and category counts and proportions

There were two main types of counts of category and product pages. The vanilla simple count and distinct count[3] per specific a category or product (for example apparel in case of distinct category). The calculation was performed cumulatively per unique visit ID. From technical perceptive general loops are not feasible for big data sets, therefore Python Pandas inbuilt function .itertuples was applied[4]. Generally the best methods for looping big data sets are vecotrized functions where operations are executed on entire arrays. Product and category proportions were calculated as vanilla counts divided by hit number.

Product and category variance

As proved by Moe (2003) product and category variety can help determine type of website visitor. Therefore pseudo product and category variance were calculated. The formula for calculation is defined by equation 5.1. In this case the logic of the pseudo variance is upside down. The product/category page

[1]Click through of product lists = 1, Product detail views = 2, Add product(s) to cart = 3, Remove product(s) from cart = 4, Check out = 5, Completed purchase = 6, Refund of purchase = 7, Checkout options = 8, browsing = 0

[2]PagePathLevel1 is the first level of of categorization of Google's store website.

[3]Distinct count is analogical to "count(distinct *)" in SQL statement.

[4]This loop was 20 000 times faster than standard for loop.

variance of one means, that all hits that occurred on product/category page happened on the one exact product/category page. In contrast values close to zero mean high variance.

$$ProductPageVariance = DistinctProductPageCount \ / \ ProductPageCount \tag{5.1}$$

5.3.3 Stock markets

The market place is marvelous inventions of the man. From small local markets in villages where barter trade took place to the modern stock exchange where millions of trades all over the world occur each day. The market changed profoundly but the key logic behind it stayed the same. The Adam Smith's Invisible hand takes all information from sellers and buyers. The prices are set in a way so the best allocation of the scarce resources is achieved. The prices represent all available information from all sellers and buyers. The information from the stock market can be indeed useful for predicting purchasing intent on e-commerce website. Therefore three stock market indexes were merged to the data set based on the date and country.

The Standard & Poor's 500 - USA

The Standard & Poor's 500 or S&P 500 is a stock market index. It is based on performance of 500 large companies listed in United States. It is one of the most popular indexes in terms of company's equity. It dates back to 1923, when it was tracking 233 different companies on weekly bases. Back then the computing power was still very expensive, therefore it was not calculated on daily basis. The SP&500 index, as we know it today, was born on 4th of March 1957 Wilson & Jones (2002).

The SP&500 is market capitalization weighted, meaning that bigger companies in terms of market capitalization have bigger impact on the changes of the index. It is quoted in US dollars.

The Toronto Stock Market Index - Canada

The Toronto Stock Market Index or the S&P/TSX is the most important stock market in the Canada. It launched in January 1977 comprising 300 (approximately 70 percent of the Canadian equities as of today) of the largest stocks

traded on the Canadian Stock Exchange. It is capitalization weighted index and is quoted in Canadian dollars.

The Caracas Stock Exchange Stock Market Index - Venezuela

The Caracas Stock Exchange Stock Market Index (IBVC) is only available stock market index in Venezuela. It tracks 11 largest companies with most liquid stocks traded in the Caracas Stock Exchange. The index is calculated from 28 August 1997. It is capitalization weighted index and is quoted in Bolivars.

Treatment of differences

The S&P 500, S&P/TSX and IBVC were further analyzed for the relevant period of the Google Analytics data set (01.08.2016 to 01.08.2017). The basic statistics of the Open value can be seen in the Table 5.1. It is obvious that those three stock markets could not be added to the data set in absolute terms. Firstly the stock market data tends to be highly correlated. This can be solved by taking differences. Secondly the same absolute difference has different interpretation for the S&P/TSX and the IBVC as the ranges of indexes differ tremendously. To tackle this, percentage differences between days are calculated and merged with the data set based on country date. For $date_t$ the percentage difference is calculated from $date_t$ and $date_{t-1}$ to ensure availability of data.

Table 5.1: Stock markets overview

	mean	std	min	25%	50%	75%	max
S&P 500	2290.8	110.7	2083.8	2179.2	2287.3	2387.9	2482.8
S&P/TSX	15194.8	391.1	14346.9	14799.6	15301.3	15505.8	15900.9
IBVC	44.5	36.0	11.8	14.1	32.6	57.7	139.4

Chapter 6

Model - Predicting purchasing intent

The main aim of thesis is to predict purchasing intent of the customers on e-commerce website (does user xyz want to buy something – yes/no). Data used in the thesis are from Google merchandise store. The data consists of clicks that users performed on website. The data are quasi panel data – having three dimensions (primary ID, Click number (hit number), features). The base model for the analysis was Logit model. The data for the model was aggregated (grouped by primary ID) taking maximum, mean and variance. The aggregation was performed always on the first 20 hits. The secondary model was Vanilla Neural network taking in also the aggregated data. The main model is Recurrent neural network (LSTM) as it is able to take in as an input 3 dimensional data set and analyze sequences. Furthermore, the results from LSTM were analyzed from two perspectives of practical usage.

6.1 Correlation analysis

In order to better understand relationships between variables it is a good practice to plot the correlation matrix. The data set was firstly aggregated based on the primary ID. Generally it is hard to analyze sequential data sets because of the structure. Although there are some more advanced methods to check relationships the author of the thesis chose more parsimonious approach. All available variables were analyzed. The variables with very low correlation with dependent variable (is_buying) were disregarded. Furthermore pairwise correlation was checked. High correlation was detected between aggre-

gated contentGroupUniqueViews2[1] and calc_category_dist_c[2] approximately 98 percent. For non arrogated data the correlation is bellow 0.5 a therefore contentGroupUniqueViews2 will we used in LSTM case. The results of the correlation analysis are present in Figure 6.1) and Figure 6.2). For the purposes of clear presentation the variables were dived into two groups, variables already present in the data set (First figure) and variables that were calculated (second figure). The description of all variables used for model is present in the Appendix B.

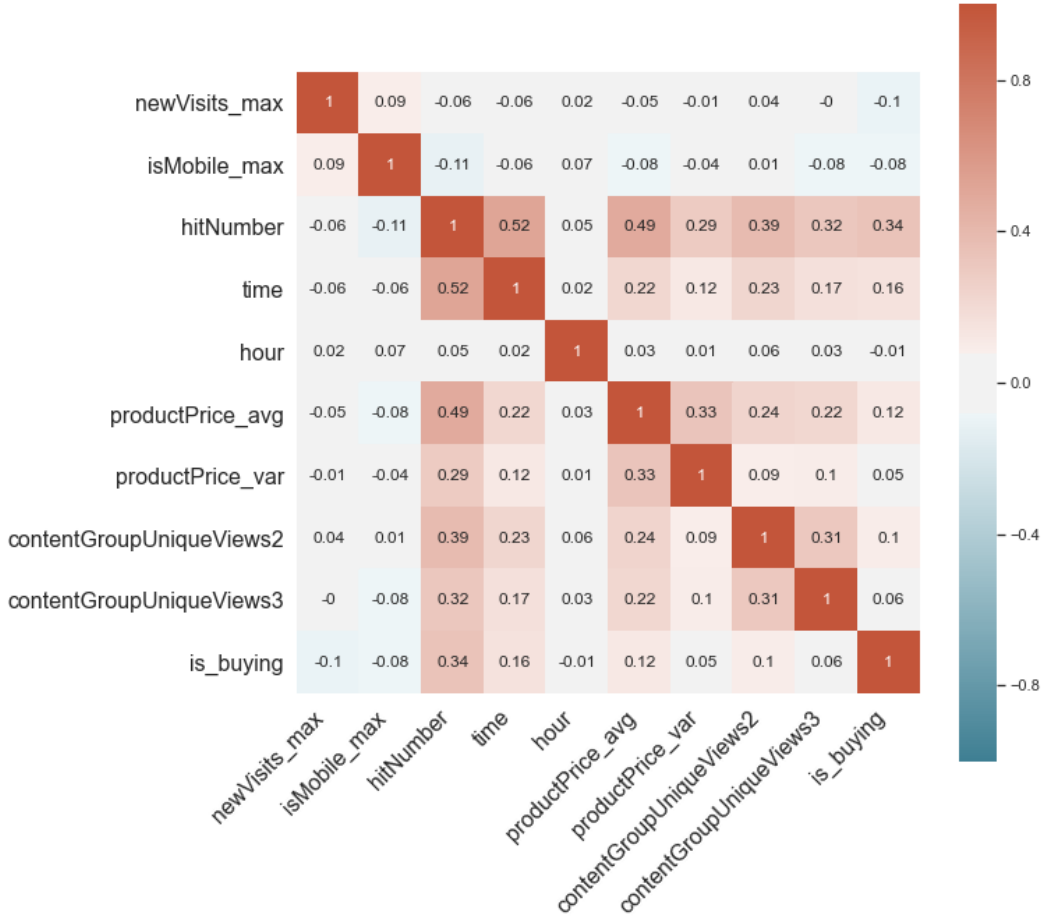

Figure 6.1: Correlation matrix - purchasing intent 1

The Correlation matrix 1 presents all pairwise correlation coefficients between all variables. The analysis is performed on aggregated data. The predicted variable is_buying is dummy variable where one means that transaction occurred in that particular session.

There are some relatively high pairwise correlations in case of calculated variables. Even though correlation is high in aggregated form for the case of not aggregated data set the correlation is usually smaller than 0.5. The calculated

[1]For the definition see Appendix B.
[2]For the definition see Appendix B.

variable pct_diff_open has very small correlation to the model, nevertheless is it was kept as predictor in logistic regression and model deemed it useful, therefore it was kept having (small) positive impact on the model.

The logit model was calculated many times with different variables. Not all variables used in model are present in the figures. The variables with smaller correlation (in absolute terms) were also included in the model because the model perform better with them. All variables used are present in the model summary in the case of the Logit model. The Vanilla neural network uses the same variables as the logit model. The LSTM model was further developed as it was possible to add categorical features that differ across hits (type of the page - information, category and product; type - event, page).

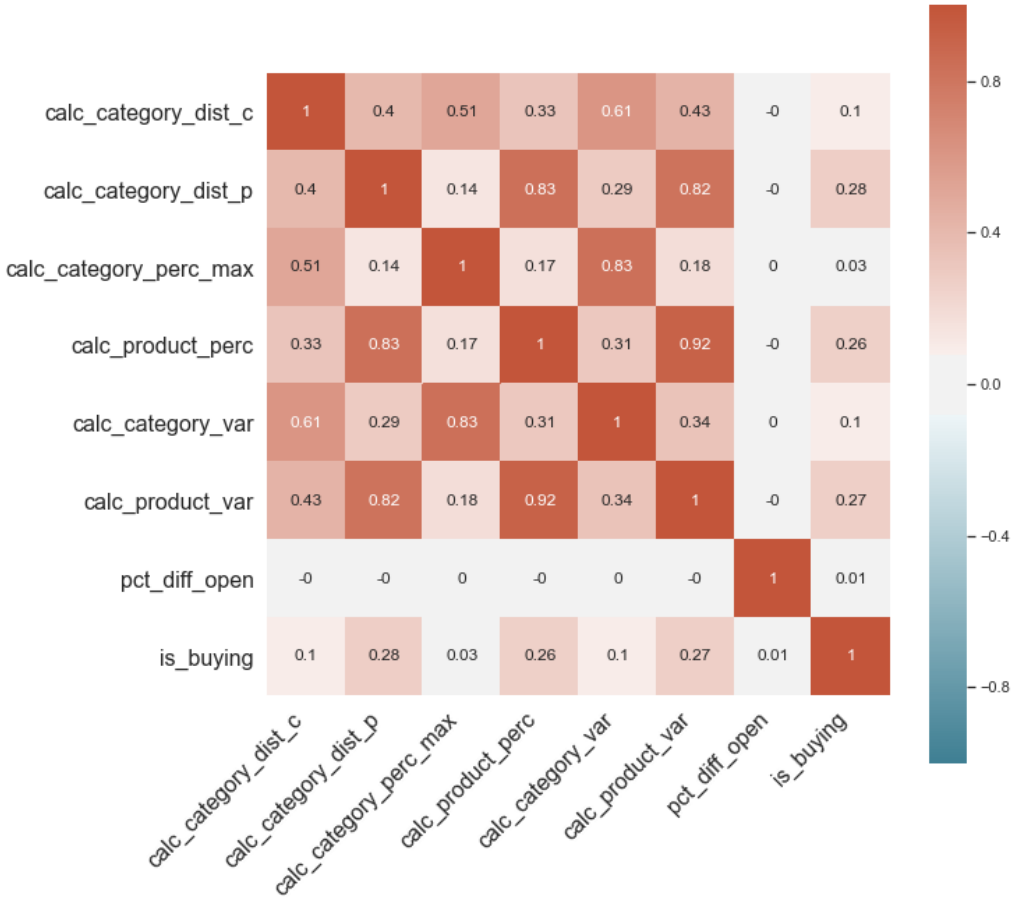

Figure 6.2: Correlation matrix - purchasing intent 2

The Correlation matrix 2 presents all pairwise correlation coefficients between all variables. The analysis is performed on aggregated data. The predicted variable is_buying is dummy variable where one means that transaction occurred in that particular session.

6.2 Base model - Logit

The main reason to calculate the Logit model was set up the base the most simple model, see if it is even possible to predict something reasonable and understand which variables work for the model and which not. Even though the ANNs are usually superior in terms of predicting capabilities to the logit model it is hard to see inside the network and understand what works and what does not without countless iterations that are very costly in terms of time and computing power.

The overview of the final results are present in the Apendix A. The model was a few times recalculated to achieve best possible Out of sample (OoS) performance. The model was not treated for any issues like homogeneity, serial correlation, normality. The sole purpose is the predicting power of the model on the OoS. As the data set is not balanced, meaning that there are much more cases of negative event (not buying session) than positive event (buying session), therefore the parameter setting weights to the observations was set, so the model is balanced, meaning that the positive observations were more important.

6.2.1 Results

In terms of the performance, the model performs fairly good on the OoS with 90:10 split achieving Area under curve (AUC) (see the Figure 6.3) 0.85. The curve is not smooth and probably the performance of the model could be increased in terms of few percent.

The rough shape of the curve is probably caused by the settings of logit function set to be balanced. The probability calculated by logit is between 0.5 and 1. Therefore if the threshold for positive outcome is set bellow or equal to 0.5, the model deems everything to be 1. Therefore the model is linear between 0.2 and 1 False positive rate. If the threshold is set to be higher than 0.5 then we are at the 0.2 at the edge. Should we set the threshold even higher, we are in interval 0, 0.2 of false positive rate.

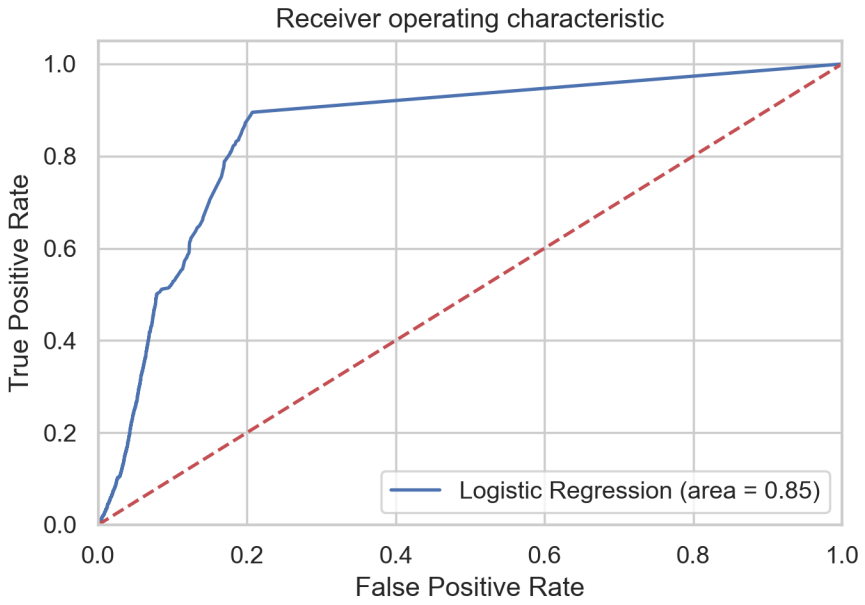

Figure 6.3: Logistic regression results - AUC OoS
The AUC figure shows predicting power of the Base logistic model on OoS data sets. The 0.85 is descent result with good overall performance.

6.3 Vanilla neural network

The Vanilla neural network is simple,intuitive and not very hard to train in terms of computing power. The Vanilla model provided valuable experience for the main model. It also takes as an input aggregated data meaning that information between clicks is lost.

6.3.1 Model architecture

In terms of architecture author experimented with different number of layers, neurons and activation functions. The final model is presented in the figure 6.4. The choice of the activation function was very important because when was model trained in the first instances it worked fine with sigmoid activation function and tanh activation function. In case of ReLU the model did not learn. The ReLU function allows gradients to explode as for each x bigger than zero it gives back x. As suggested by Bishop *et al.* (1995) it is nearly always advantageous to apply pre-processing transformations to the input data before it is presented to a network. Author suggests a few method to pre-process data. One of them is normalization which scales data down based of their max and

min values. The normalization process is standard in Data science and can be easily described by the equation 6.1.

$$y = (x - min)/(max - min) \tag{6.1}$$

The data was therefore normalized achieving better overall results (for all activation functions). The best results were achieved with ReLU activation function. The weighting matrix was also used to balance the data set. When balancing was not applied the ANN deemed all observations to be negative (not buying) having accuracy approximately 97 percent.

The NAG was used as an algorithm for optimization of learning. The NAG was more consistent compared to other optimizers, achieving stable improvement of the network as it was trained. The batch size was set to 128 which was fair compromise between stable and fast learning of the network.

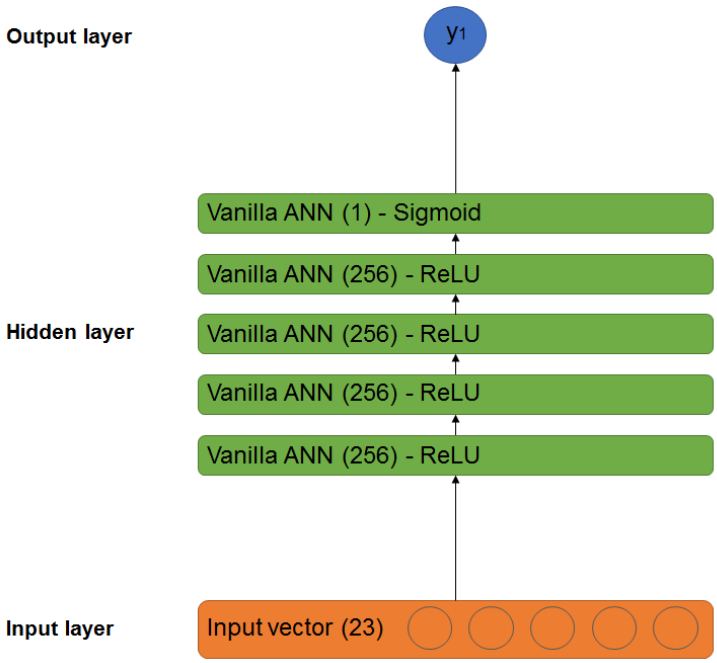

Figure 6.4: Vanilla ANN diagram

The ANN diagram shows the architecture of the base neural network. As input network takes aggravated data based on primary ID. The network has 4 dense layers using ReLU as an activation function. The last layer give output as a probability between 0 and 1.

6.3.2 Results

The results of the Vanilla ANN are present on the Figure 6.5. The overall score of 0.95 is much better compared with the performance of the base Logit model.

The AUC curve is much smoother. Even though some information is indeed lost when data is aggregated the results are tremendous.

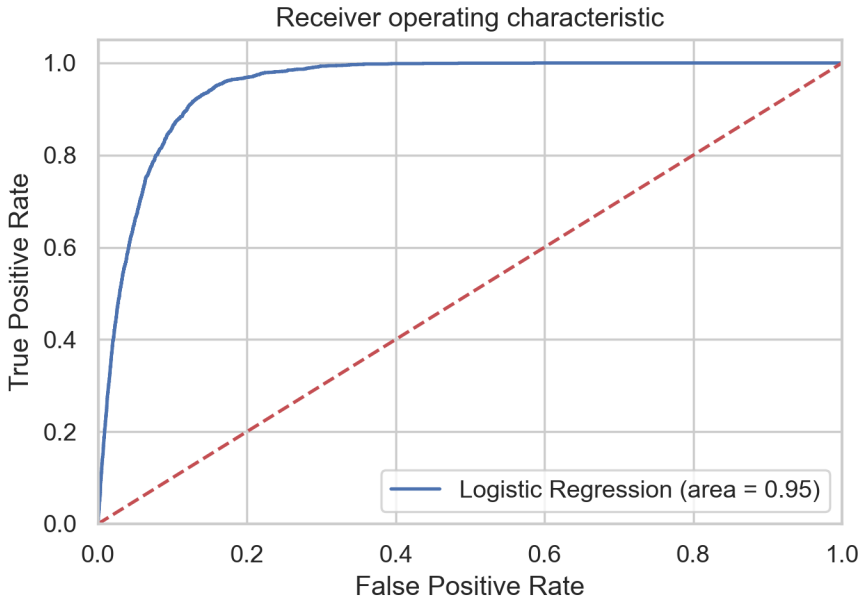

Figure 6.5: Vanilla ANN results

The AUC figure shows predicting power of the Vanilla ANN model on OoS data set. The 0.95 is descent result with good overall performance with much smoother curve compared to the Base model.

6.4 LSTM

The LSTM model represents the final and the most complex model used to predict purchasing intent. The main advantage of the model is that the model is able to take as an input 3 dimensional array. The data is not pre-aggregated and therefore the aggregation is done via neural network internally. The different types of the model were tried including word embedding. The best results were achieved by the model presented in following chapter.

6.4.1 Model architecture

The model architecture is described in the Figure 6.6. The first three layers of the network are LSTM taking as an input non-aggregated data and aggregating it. Furthermore, there are three more standard layers. The model was also tested without three dense layers and the results were similar but generally little bit worse (in terms of 0.001). The following model is presented to show the best possible results as the final although if the model should be implemented the simpler and more parsimonious model would be probably better.

The NAG was used as an algorithm for optimization of learning. The NAG was more consistent compared to other optimizers, achieving stable improvement of the network as it was trained. The batch size was set to 128 which was fair compromise between stable and fast learning of the network.

Technical note

All models were trained on the machine with following specifications, Processor: Intel Xeon L5640 2.27Ghz (2 processors); RAM: 32 GB; GPU: NVIDIA GeForce GTX 1060 3GB. The TensorFlow was set to perform calculation both on processor and GPU. The most time demanding was the LSTM. Instead of standard LSTM, the CuDNNLSTM was used, because it is optimized to run on GPU and runs 10 times faster. Specifications of machine are relatively modest suggesting easy implementation of the framework.

6.4.2 Results

The results of the model are presented in the table 6.7. The model was trained on 5, 10, 15 and 20 hits respectively. The results are presented overall and by the hits for two different practical application of the model. Therefore, there

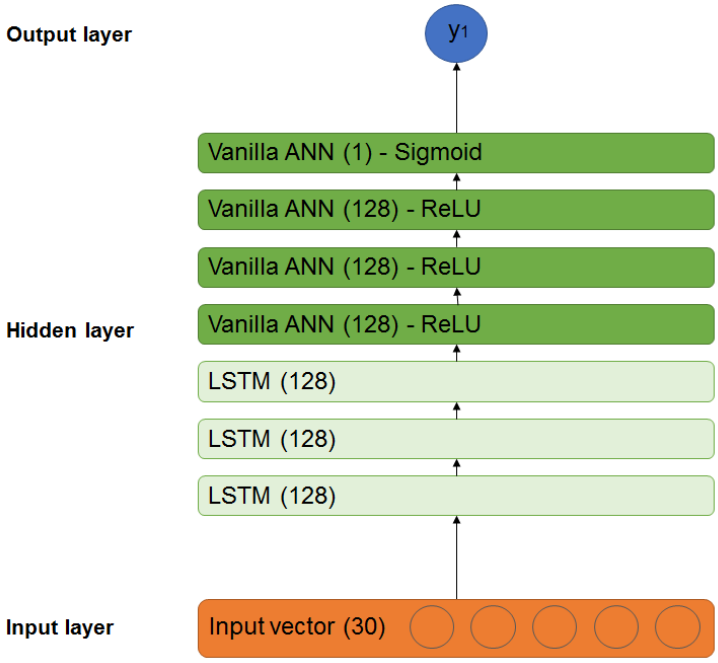

Figure 6.6: LSTM diagram

The LSTM diagram show the architecture of the LSTM model. The first three layers are LSTM and serve as a aggregation of the sequential data. The other three layers work with the aggravated data. The last layer gives one output between 0 and 1.

are two types of evaluation metrics presented that show different point of view on the same model (== ; =>).

The first model, the ex ante represented by => portraits scenario where algorithm actively analyzes all visitors in real time. The tested score means that all observations are tested that have at least x hits (for 2 it is 2, 3 ,4 ... 20). Should the user be predicted to be willing to buy something website could engage with him or her further. This leads to the situation where website actively helps those willing to buy something and does not disturbed those who are just window shopping, achieving better overall customer experience. This evaluation of the performance of the model has not been tried before to the best of my knowledge.

The second model, the ex post represented by ==, portraits scenario where users are evaluated after they finished browsing. The observations are tested where the max number of hits (clicks) is equal to the N_hits. This information is realized after the browsing has already ended therefore, it is called ex post. The use case for this approach should also be obvious. After user has finished browsing, he or she is further analyzed by the algorithm. If he or she was predicted to buy something but he or she did not, he or she probably visited

N_hits	5		10		15		20	
Overal OoS / In	0.91	0.90	0.95	0.94	0.97	0.96	0.97	0.96
Overal	=>	==	=>	==	=>	==	=>	==
1	0.65	0.87	0.66	0.88	0.67	0.89	0.65	0.89
2	0.62	0.80	0.61	0.80	0.63	0.79	0.57	0.83
3	0.68	0.76	0.67	0.82	0.67	0.82	0.62	0.84
4	0.72	0.86	0.68	0.88	0.70	0.88	0.66	0.89
5	0.74	0.79	0.70	0.84	0.72	0.84	0.69	0.87
6	0.76	0.87	0.74	0.89	0.74	0.90	0.71	0.89
7	0.76	0.90	0.76	0.93	0.76	0.93	0.72	0.91
8	0.76	0.85	0.77	0.90	0.77	0.91	0.74	0.92
9	0.76	0.85	0.78	0.91	0.77	0.92	0.74	0.93
10	0.75	0.80	0.78	0.89	0.78	0.91	0.75	0.92
11	0.74	0.82	0.78	0.89	0.78	0.90	0.76	0.92
12	0.73	0.79	0.77	0.87	0.78	0.89	0.76	0.90
13	0.72	0.73	0.77	0.85	0.78	0.89	0.77	0.91
14	0.71	0.70	0.76	0.83	0.78	0.87	0.77	0.90
15	0.70	0.70	0.75	0.83	0.78	0.89	0.77	0.91
16	0.68	0.66	0.73	0.79	0.77	0.86	0.77	0.90
17	0.67	0.63	0.72	0.77	0.76	0.86	0.76	0.89
18	0.66	0.61	0.71	0.73	0.76	0.83	0.76	0.87
19	0.66	0.61	0.70	0.71	0.75	0.81	0.76	0.84
20	0.65	0.65	0.68	0.68	0.74	0.74	0.75	0.75

Figure 6.7: The LSTM model results

The LSTM model results table comperes overall results of the model on the OoS and IS basis. Moreover there are results present also for two practical applications of the model, ex ante (=>) and ex post (==). The ex ante model represents results of live model that in real time predicts intention of the customer. The ex post model predicts probability of the user after user already finished browsing.

other e-commerce stores. Therefore, the website could automatically send him or her email (provided that he or she was registered) or in case of more valuable goods and service (the case of the bank) provide his information to the call center so the salesperson could contact him or her directly with some discount or better service. The ex post model has much better performance as the ex ante model. The literature suggests the most information is present in last click of the session. This information is not always present in case of ex ante scenario where all observations with at least x hits are present. In addition it is generally easier to make correct prediction when observations are categorized in categories (based on number of hits) which is present in ex post model.

Generally, the results of the model are great and show a huge possibilities for implementation especially for cases where the good is expensive. The overall score of the model of 0.97 AUC on OoS outperforms logit and Vanilla models.

Chapter 7

Model - Predicting website exit

The precise prediction in real time whether visitor will or will not leave the website, would allow e-commerce website to bring experience from shopping to the next level. If done precisely it enables e-commerce website to discriminate between users which would increase profits. If the user wants to buy something but he is also looking around meaning that he or she will be leaving in next few clicks, without finishing order, the website could offer discount or limited offer. This model further develops ex ante model. The chapter presents preliminary correlation analysis and the LSTM model.

7.1 Correlation analysis

The correlation analysis was performed on all available variables but in this case the data set was not aggregated. The dependent variable y represents event that in next 3 click visitor will leave the e-commerce website. The variables with very low correlation with dependent variable (y) were disregarded. Furthermore, the pairwise correlation was checked. As some variables were calculated from categorical features, some variable are perfectly correlated. This was used primarily in case of model with embedding layer that did not outperform base model. Nevertheless including or excluding one of the variables did not had any effect on the performance of the LSTM therefore it was kept. The results of the correlation analysis are present in Figure 7.1 and Figure 7.2. For the purposes of clear presentation the variables were dived into two groups, variables already present in the data set (First figure) and variables that were calculated (second figure). The description of all variables used for model is present in the Appendix B. Not all variables are present in this analysis. The

variables with smaller correlation (in absolute terms) were also included in the model because the model perform better with them. The analysis shows the variables with highest correlation.

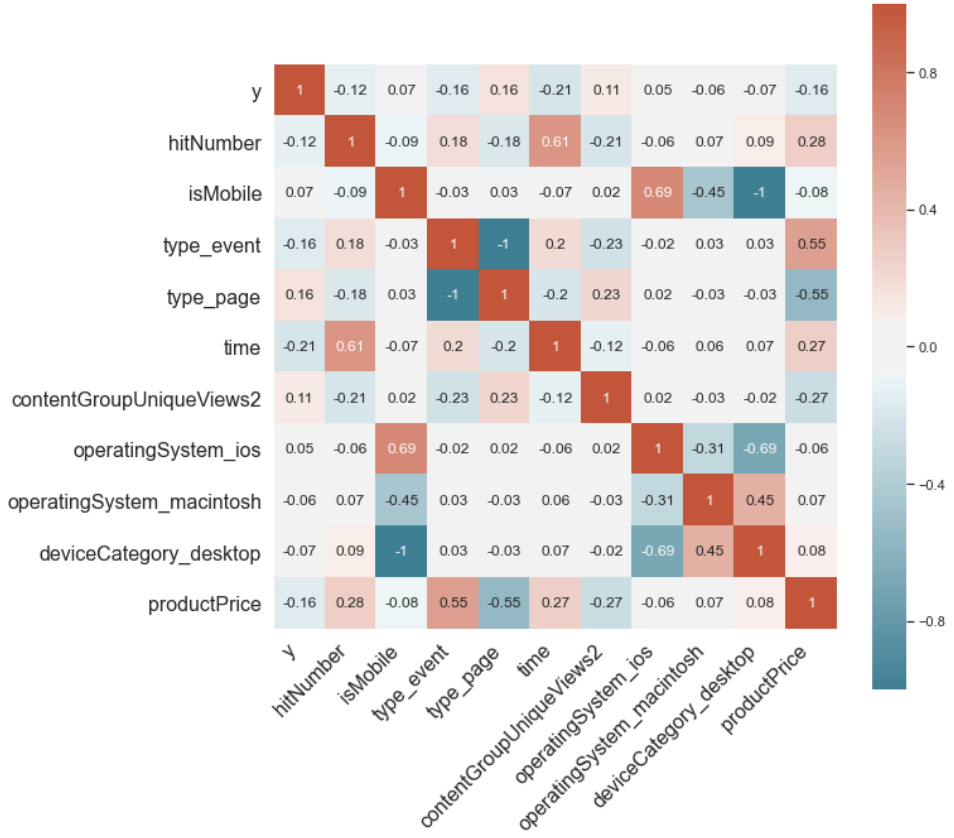

Figure 7.1: Correlation matrix - exit 1

The Correlation matrix 1 presents all pairwise correlation coefficients between all variables. The analysis is performed on non aggregated data. The predicted variable y is dummy variable where one means that visitor will leave in next three clicks.

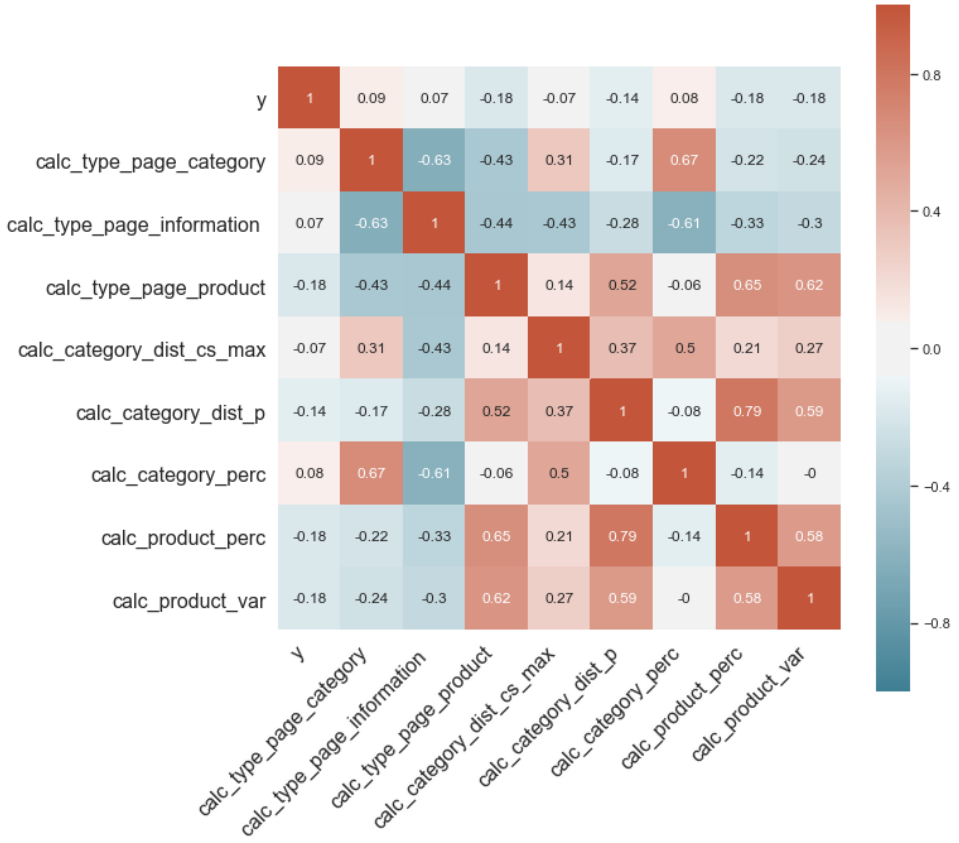

Figure 7.2: Correlation matrix - exit 2

The Correlation matrix 1 presents all pairwise correlation coefficients between all variables. The analysis is performed on non aggregated data. The predicted variable y is dummy variable where one means that visitor will leave in next three clicks.

7.2 LSTM

The LSTM model proved to be the best model for predicting purchasing intent. It is possible to model it with 3 dimensional input which is indeed useful in case of click-stream data for e-commerce website. The Exit modeling is difficult task as there are more hidden endogenous variables. You could be browsing on the website, suddenly you got text from friend inviting you in a bar. You stop browsing a start to get ready. Also you may be distracted by other applications on your computer. Basically there are two types of distraction, from real world and from the computer. The distractions from real world are impossible to model. The distraction from the computer can be to some degree taken care of but there are so many possibilities it is very hard task. Therefore it is much harder to predict correctly what is going to happen.

7.2.1 Model architecture

The LSTM model was in this case used in different specification. Instead of giving just one output per one session[1], it gives prediction per each click in each session taking into account past clicks into account. The model consists only from LSTM layers and the last layer gives us an output sequence[2]. For more information about shapes of LSTM inputs and output see 4.6. The representation of the model is present on the Figure 7.3. There were several specifications of the model with different number of layers, neurons and shapes tried. Also the model with embedding layer was tested. The following representation yielded the best possible outcomes.

The NAG was used as an algorithm for optimization of learning. The NAG was more consistent compared to other optimizers, achieving stable improvement of the network as it was trained. The batch size was set to 128 which was fair compromise between stable and fast learning of the network.

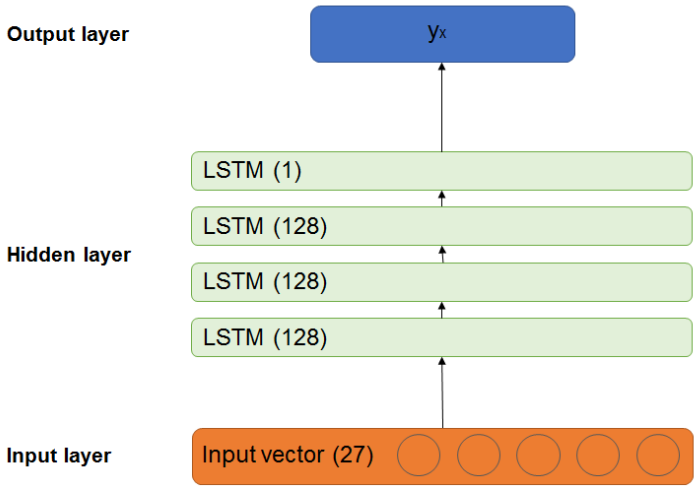

Figure 7.3: LSTM diagram - exit

The LSTM diagram show the architecture of the LSTM model. In terms of the LSTM shapes the many to many model is applied (see 4.6). The first three layers are LSTM and serve to analyze the data. The last layer gives sequential output (vector) for each hit in input layer between 0 and 1.

[1] consists of multiple clicks (hits)

[2] As mentioned before it is desirable to keep output in sequential shape. This means that flattening it would be contra productive.

7.2.2 Results

The results of the model are presented in the table 7.4. The model was trained on 5, 10, 15 and 20 hits respectively. The results are presented overall and by the hits in terms of accuracy and AUC score. Similarly to ex ante model the algorithm actively analyzes all visitors in real time. The tested score means that all observations are tested that have at least x hits (for 2 it is 2, 3 ,4 ... 20). Should the user be predicted to leave the website in next three click the website could engage actively with user. It could recommend him other products suggest some discount or free shipping depending on the business model of the e-commerce website.

The model is intended to be used together with the model predicting purchasing intent the ex ante version. While the user browses the website automatically calculates the probability of exit. If it reaches some threshold the website calculates the probability of the user wanting something to buy. Provided the willingness of the visitor to buy something is high the website could engage with user further.

The results in terms of accuracy are very good and in terms of the AUC are descent taking into account difficulty to predict user behavior. The results show that after only one hit, it is very hard to predict future behavior. As more hits are available model predicts better in terms of accuracy and AUC. The accuracy steadily rises to approximately 0.9 and AUC to 0.7.

	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
N_hits	5		10		15		20	
Overal OoS	0.86	0.84	0.92	0.89	0.94	0.92	0.95	0.94
Overal IS	0.86	0.84	0.92	0.90	0.94	0.92	0.95	0.94
1	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.65
2	0.81	0.64	0.80	0.64	0.80	0.64	0.81	0.63
3	0.81	0.65	0.81	0.65	0.81	0.65	0.81	0.63
4	0.84	0.66	0.84	0.66	0.84	0.66	0.84	0.66
5	0.86	0.68	0.86	0.68	0.86	0.68	0.86	0.67
6	0.87	0.68	0.87	0.69	0.87	0.69	0.87	0.68
7	0.88	0.68	0.88	0.69	0.88	0.69	0.88	0.69
8	0.89	0.68	0.89	0.70	0.89	0.69	0.89	0.69
9	0.90	0.68	0.90	0.70	0.90	0.70	0.90	0.70
10	0.90	0.68	0.90	0.70	0.90	0.70	0.90	0.70
11	0.91	0.68	0.91	0.70	0.91	0.70	0.91	0.70
12	0.91	0.67	0.91	0.69	0.91	0.70	0.91	0.70
13	0.92	0.68	0.92	0.70	0.92	0.71	0.92	0.70
14	0.91	0.67	0.91	0.69	0.91	0.70	0.91	0.70
15	0.92	0.68	0.92	0.70	0.92	0.71	0.92	0.71
16	0.92	0.68	0.92	0.70	0.92	0.72	0.92	0.72
17	0.93	0.68	0.93	0.70	0.93	0.72	0.93	0.72
18	0.93	0.68	0.93	0.70	0.93	0.72	0.93	0.72
19	0.93	0.68	0.93	0.70	0.93	0.72	0.93	0.72
20	0.93	0.68	0.93	0.70	0.93	0.72	0.93	0.72

Figure 7.4: The Exit model results

The Exit model results table comperes overall resultsin terms of Accuracy (Acc) and Area under curve (AUC) of the model on the OoS and IS basis. Moreover there are results present per hits for ex ante model.

Chapter 8

Practical implementation

"The reason it seems that price is all your customers care about is that you haven't given them anything else to care about." (Seth Godin)

The most important part of the empirical research should be practical implementation. The proposed models for predicting purchasing intent and predicting exit are further described in following chapter in terms of possible practical implementations.

Ex ante

The ex ante model represents the scenario where both algorithms predicting exit and buying intention work with synergy providing the best possible customer experience. The whole solution is present on the Diagram 8.1. As the visitor starts browsing, he or she clicks trough the e-commerce website. Should he or she be predicted to leave the website in next few clicks the website predicts his or her possible buying intentions. Should he or she be willing to buy something, but at same time there is a huge probability of him or her leaving the website, the website should engage with user with custom content. This can include discount, helpful pop up, free shipping or suggestion of other items. This could prevent visitor exit and hopefully the visitor will interact with the website further resulting into a buying event. This solution is very simple a straight forward. It does not require website to have direct contact on the user (email or phone number). The most important is the e-commerce website does not hares its shoppers with constant suggestions but the suggestions of new content happen only in specific cases (user is going to exit and is willing to buy

something). This should lead to better experience for the customer and higher profits for the e-commerce shop.

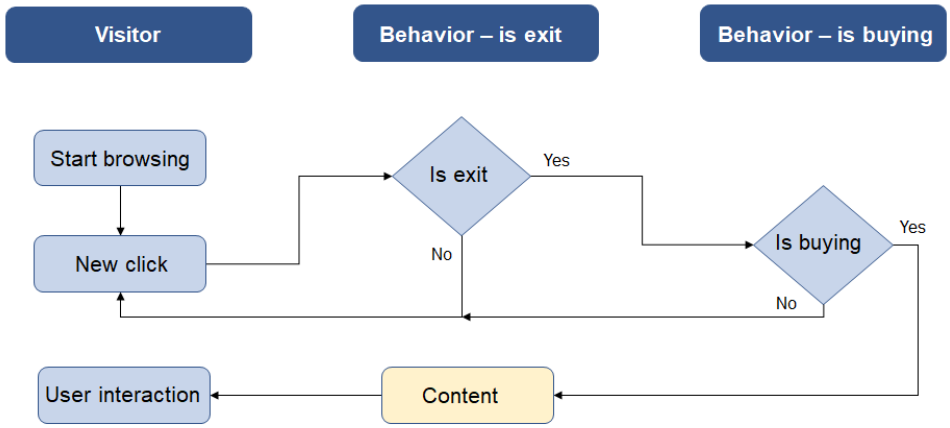

Figure 8.1: Proposed solution diagram

The Proposed solution diagram shows how the ex ante model predicting if user is going to buy something and the second model which predicts if the user will exit in next few clicks will work together. The user visits an e-commerce website. If the visitor does not seem to exit system is idle. If the system predicts that user is going to exit the website and is willing to buy something it will provide user with addition custom content. This can be some suggestion of goods, discount friendly pop up, free shipping etc. Visitor than interacts with new content, hopefully resulting into successful purchase.

Ex post

The ex post model would present interesting solution for website which require user to be registered and therefore the website should have at least his or her email address or phone number. After visitor has ended his or her session on a e-commerce website the algorithm calculate probability of him or her wanting to buy something on the e-shop. If the probability is high but he did not purchase anything on the website the algorithm could send him an email with content (suggestion, discount, free shipping etc.). With the use of the algorithm website does not spam all its visitors with unwanted content in case when email communication is used. In case of phone call (for more expensive goods and services) it is beneficent for both customer and the e-commerce website. The shoppers do not get unwanted phone calls and the e-commerce store saves costs for their call center which can be quit substantial.

Chapter 9

Conclusion

In this thesis, the data set from Google's Merchandise store was explored in terms of overall statistics, prediction of purchasing intent of the customers and prediction of the exit of the customers. In addition, practical applications of the model are presented.

The first part of the analysis consisted of the exploration of the data set, provided by the Google in order to promote their services for websites around the globe (Google Analytics). The data set includes one year of data collected from Google's merchandise store. The data set has three main dimension, primary ID of the visitor, click number and features. There is more than 300 features available that the Google tracks about their visitors. The 3 dimensional type of the data set is a standard shape of the click-stream data that show behavior of visitors on the websites on the Internet. The data set was analyzed as it was, with focus on efficient computing due the sheer size of the data set.

The second part of the analysis took into account knowledge about the data set, repaired the shortcomings of the data set. Furthermore the features commonly present in the literature were calculated to achieve best possible prediction capabilities. The most important once were type of the webpage seen by the user (product page, category page or information page), product and category variance and number of distinct product and category pages seen by the shopper.

The third part of the analysis consisted of creation of the main model which predicts purchasing intention of the customer on the e-commerce website, using the best practices in machine learning. The base model was logit. This model was further developed into vanilla artificial neural network and later in Recurrent neural network the Long short term memory model (LSTM). The

LSTM outperformed other models mainly because of possibility to take as an input 3 dimension data set. The model was evaluated by the standard evaluation metric - area under curve metric. In addition model was further evaluated with respect to the two possible practical applications in new and novel way.

In the forth part, the secondary model was developed using the LSTM with different architecture in order to predict whether the shopper on the e-commerce website will leave the website in next few clicks in real time. The performance of the model was checked in terms of accuracy and area under curve.

Finally the possible implantation of the both models was described in greater detail. Two possible applications were described. The ex ante model and ex post model. The ex ante model is calculated in real time while the shopper is still browsing. If the user is predicted to be leaving the website in next few clicks and he or she is willing to buy something, the website could provide user with additional content keeping him or her on the website hopefully finishing order. The ex post model analyzes the behavior of the shopper after he or she has already left. Should the shopper be predicted with high purchasing intention the website could send him or her automatic email with special content (discount, suggestion of goods, free shipping etc.). Furthermore, if the e-commerce store has more information about its visitors (phone number) the website could automatically send lists of users that likely want to buy something to the call center connecting the virtual and real shopping experience, saving money for the e-commerce store and valuable time of the customers.

In conclusion I would like to say that the analysis was successful developing interesting models with good overall performance.

Bibliography

Ackley, D. H., G. E. Hinton, & T. J. Sejnowski (1985): "A learning algorithm for boltzmann machines." *Cognitive science* **9(1)**: pp. 147–169.

Anonymous (1958): "New navy device learns by doing psychologist shows embryo of computer designed to read and grow wiser." *New York Times* .

Bag, S., M. K. Tiwari, & F. T. Chan (2019): "Predicting the consumer's purchase intention of durable goods: An attribute-level analysis." *Journal of Business Research* **94**: pp. 408–419.

Bengio, Y. (2012): "Practical recommendations for gradient-based training of deep architectures." In "Neural networks: Tricks of the trade," pp. 437–478. Springer.

Bengio, Y., N. Boulanger-Lewandowski, & R. Pascanu (2013): "Advances in optimizing recurrent networks." In "2013 IEEE International Conference on Acoustics, Speech and Signal Processing," pp. 8624–8628. IEEE.

Bengio, Y., P. Simard, P. Frasconi *et al.* (1994): "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks* **5(2)**: pp. 157–166.

Bishop, C. M. *et al.* (1995): *Neural networks for pattern recognition*. Oxford university press.

Bucklin, R. E. & C. Sismeiro (2003): "A model of web site browsing behavior estimated on clickstream data." *Journal of marketing research* **40(3)**: pp. 249–267.

Carmona, C. J., S. Ramírez-Gallego, F. Torres, E. Bernal, M. J. del Jesús, & S. García (2012): "Web usage mining to improve the design of an e-commerce website: Orolivesur. com." *Expert Systems with Applications* **39(12)**: pp. 11243–11249.

Dauphin, Y. N., R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, & Y. Bengio (2014): "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization." In "Advances in neural information processing systems," pp. 2933–2941.

Gao, P., H. Li, S. Li, P. Lu, Y. Li, S. C. Hoi, & X. Wang (2018): "Question-guided hybrid convolution for visual question answering." In "Proceedings of the European Conference on Computer Vision (ECCV)," pp. 469–485.

Gers, F. A., J. Schmidhuber, & F. Cummins (1999): "Learning to forget: Continual prediction with lstm." .

Glorot, X., A. Bordes, & Y. Bengio (2011): "Deep sparse rectifier neural networks." In "Proceedings of the fourteenth international conference on artificial intelligence and statistics," pp. 315–323.

Graves, A., M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, & J. Schmidhuber (2008): "A novel connectionist system for unconstrained handwriting recognition." *IEEE transactions on pattern analysis and machine intelligence* **31(5)**: pp. 855–868.

Group, M. M. (2018): "Oecd member countries data."

Harris, Z. S. (1954): "Distributional structure." *Word* **10(2-3)**: pp. 146–162.

Hochreiter, S. & J. Schmidhuber (1997): "Long short-term memory." *Neural computation* **9(8)**: pp. 1735–1780.

Hu, R., H. Xu, M. Rohrbach, J. Feng, K. Saenko, & T. Darrell (2016): "Natural language object retrieval." In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition," pp. 4555–4564.

Janiszewski, C. (1998): "The influence of display characteristics on visual exploratory search behavior." *Journal of consumer research* **25(3)**: pp. 290–301.

Karlik, B. & A. V. Olgac (2011): "Performance analysis of various activation functions in generalized mlp architectures of neural networks." *International Journal of Artificial Intelligence and Expert Systems* **1(4)**: pp. 111–122.

Lazcorreta, E., F. Botella, & A. Fernández-Caballero (2008): "Towards personalized recommendation by two-step modified apriori data mining algorithm." *Expert Systems with Applications* **35(3)**: pp. 1422–1429.

Le, Q. V., N. Jaitly, & G. E. Hinton (2015): "A simple way to initialize recurrent networks of rectified linear units." *arXiv preprint arXiv:1504.00941* .

LeCun, Y., Y. Bengio, & G. Hinton (2015): "Deep learning." *nature* **521(7553)**: p. 436.

Liu, B. & I. Lane (2015): "Recurrent neural network structured output prediction for spoken language understanding." In "Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions," .

Mandel, N. & E. J. Johnson (2002): "When web pages influence choice: Effects of visual primes on experts and novices." *Journal of consumer research* **29(2)**: pp. 235–245.

Masters, D. & C. Luschi (2018): "Revisiting small batch training for deep neural networks." *arXiv preprint arXiv:1804.07612* .

Mikolov, T., K. Chen, G. Corrado, J. Dean, L. Sutskever, & G. Zweig (2013a): "word2vec." *URL https://code. google. com/p/word2vec* .

Mikolov, T., W.-t. Yih, & G. Zweig (2013b): "Linguistic regularities in continuous space word representations." In "Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies," pp. 746–751.

Minsky, M. & S. A. Papert (1969): *Perceptrons: An introduction to computational geometry*. MIT press.

Moe, W. W. (2003): "Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream." *Journal of consumer psychology* **13(1-2)**: pp. 29–39.

Moe, W. W. & P. S. Fader (2000): "Which visits lead to purchases? dynamic conversion behavior at e-commerce sites." *University of Taxas* .

Moe, W. W. & P. S. Fader (2001): "Uncovering patterns in cybershopping." *California Management Review* **43(4)**: pp. 106–117.

Montgomery, A. L., S. Li, K. Srinivasan, & J. C. Liechty (2004): "Modeling online browsing and path analysis using clickstream data." *Marketing science* **23(4)**: pp. 579–595.

OECD (2008): "The future of the internet economy: A statistical profile." *Statistical profile prepared for the OECD Ministerial meeting on the Future of the Internet Economy taking place in Seoul* .

Olazaran, M. (1996): "A sociological study of the official history of the perceptrons controversy." *Social Studies of Science* **26(3)**: pp. 611–659.

Pai, D., A. Sharang, M. M. Yadagiri, & S. Agrawal (2014): "Modelling visit similarity using click-stream data: A supervised approach." In "International Conference on Web Information Systems Engineering," pp. 135–145. Springer.

Pascanu, R., T. Mikolov, & Y. Bengio (2013): "On the difficulty of training recurrent neural networks." In "International conference on machine learning," pp. 1310–1318.

Pratchett, T. (2008): *Hogfather:(Discworld Novel 20)*, volume 20. Random House.

Quick, R. (1998): "Gawkers or shoppers? selling bras on the web." *The Wall Street Journal* p. B1.

Rajamma, R. K., A. K. Paswan, & M. M. Hossain (2009): "Why do shoppers abandon shopping cart? perceived waiting time, risk, and transaction inconvenience." *Journal of Product & Brand Management* **18(3)**: pp. 188–197.

Reed, R. & R. J. MarksII (1999): *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press.

Rohm, A. J. & V. Swaminathan (2004): "A typology of online shoppers based on shopping motivations." *Journal of business research* **57(7)**: pp. 748–757.

Romov, P. & E. Sokolov (2015): "Recsys challenge 2015: ensemble learning with categorical features." In "Proceedings of the 2015 International ACM Recommender Systems Challenge," p. 1. ACM.

Rosenblatt, F. (1958): "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* **65(6)**: p. 386.

Ruder, S. (2016): "An overview of gradient descent optimization algorithms." *arXiv preprint arXiv:1609.04747* .

Sakar, C. O., S. O. Polat, M. Katircioglu, & Y. Kastro (2019): "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks." *Neural Computing and Applications* **31(10)**: pp. 6893–6908.

Sheil, H., O. Rana, & R. Reilly (2018): "Predicting purchasing intent: automatic feature learning using recurrent neural networks." *arXiv preprint arXiv:1807.08207* .

Sibi, P., S. A. Jones, & P. Siddarth (2013): "Analysis of different activation functions using back propagation neural networks." *Journal of Theoretical and Applied Information Technology* **47(3)**: pp. 1264–1268.

SimilarTech (2019): "Google analytics."

Stathakis, D. (2009): "How many hidden layers and nodes?" *International Journal of Remote Sensing* **30(8)**: pp. 2133–2147.

Toth, A., L. Tan, G. Di Fabbrizio, & A. Datta (2017): "Predicting shopping behavior with mixture of rnns." In "eCOM@ SIGIR," .

Wilson, J. W. & C. P. Jones (2002): "An analysis of the s&p 500 index and cowles's extensions: Price indexes and stock returns, 1870–1999." *The Journal of Business* **75(3)**: pp. 505–533.

Xie, S., R. Girshick, P. Dollár, Z. Tu, & K. He (2017): "Aggregated residual transformations for deep neural networks." In "Proceedings of the IEEE conference on computer vision and pattern recognition," pp. 1492–1500.

Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, & Y. Bengio (2015): "Show, attend and tell: Neural image caption generation with visual attention." In "International conference on machine learning," pp. 2048–2057.

Zhang, Y. & J. R. Jiao (2007): "An associative classification-based recommendation system for personalization in b2c e-commerce applications." *Expert Systems with Applications* **33(2)**: pp. 357–367.

Appendix A

Logit results

Table 1 Results: Logit	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
visitNumber_max	-0.0074	0.0010	-7.5091	0.0000	-0.0093	-0.0054
newVisits_max	-0.8005	0.0232	-34.5019	0.0000	-0.8460	-0.7550
isMobile_max	-6.7722	0.0887	-76.3881	0.0000	-6.9459	-6.5984
hitNumber	0.2560	0.0028	90.2458	0.0000	0.2505	0.2616
time	0.0000	0.0000	2.1619	0.0306	0.0000	0.0000
hour	-0.0182	0.0022	-8.2442	0.0000	-0.0225	-0.0138
pct_diff_open	7.0510	2.4996	2.8209	0.0048	2.1519	11.9500
productPrice_avg	-0.0000	0.0000	-25.2879	0.0000	-0.0000	-0.0000
productPrice_var	-0.0000	0.0000	-4.1517	0.0000	-0.0000	-0.0000
calc_category_dist_c	-0.4007	0.0094	-42.8285	0.0000	-0.4190	-0.3824
calc_category_dist_p	-0.2589	0.0130	-19.9145	0.0000	-0.2844	-0.2335
calc_category_perc	0.0710	0.0456	1.5567	0.1195	-0.0184	0.1604
calc_product_perc	-0.1047	0.0707	-1.4810	0.1386	-0.2433	0.0339
calc_category_var	0.3070	0.0593	5.1767	0.0000	0.1908	0.4232
calc_product_var	1.9048	0.0561	33.9587	0.0000	1.7949	2.0147
calc_is_weekend	-0.0921	0.0309	-2.9851	0.0028	-0.1526	-0.0316
Holiday	-0.1927	0.0864	-2.2303	0.0257	-0.3621	-0.0234
contentGroupUniqueViews1	-0.2025	0.0351	-5.7634	0.0000	-0.2713	-0.1336
contentGroupUniqueViews3	-0.2762	0.0250	-11.0539	0.0000	-0.3252	-0.2272
browser_chrome	0.1782	0.0550	3.2401	0.0012	0.0704	0.2861
browser_safari	-0.3472	0.0755	-4.5992	0.0000	-0.4952	-0.1993
operatingSystem_ios	0.3681	0.0806	4.5684	0.0000	0.2102	0.5260
operatingSystem_macintosh	0.3809	0.0243	15.6970	0.0000	0.3333	0.4285
deviceCategory_desktop	-5.8079	0.0732	-79.2896	0.0000	-5.9514	-5.6643

Appendix B

Used variables

Variable name	Description
unique_session_id	unique id
y	is buying, or is exit based on context
visitNumber	the session number for this user.
newVisits	if it is the first visit, this value is 1 otherwise null
hitNumber	the sequenced hit number
isMobile	if the user is on a mobile device it is 1 else 0
hour	the hour in which the hit occurred
time	the total time (in milliseconds) spent
productPrice	product price is USD
calc_category_dist_c	number of distinct categories
calc_category_dist_p	number of distinct products
calc_category_perc	ratio of category pages
calc_product_perc	ratio of product pages
calc_product_var	pseudo product seen variance
calc_category_var	pseudo category seen variance
contentGroupUniqueViews1	The number of unique content group views.
contentGroupUniqueViews2	Content group views in different sessions are
contentGroupUniqueViews3	counted as unique content group views.
time_on_page	time spent on previous page
type_event	dummy for type of hit event
type_page	dummy for type of hit page
calc_type_page_category	dummy for category page
calc_type_page_information	dummy for information page
calc_type_page_product	dummy for product page
browser_chrome	dummy is browser chrome
browser_safari	dummy is browser safari
operatingSystem_ios	dummy is operating system iOS
operatingSystem_macintosh	dummy is operating system Mac
deviceCategory_desktop	dummy is desktop
holiday	is holiday
pct_diff_open	percentage difference in stock index price
calc_is_weekend	is weekend