**Evaluation of Machine Learning and Deep Learning Models for Customer Journey Prediction in E-Commerce - An Executive Perspective**

Master Thesis

**Author**: Alexander Vladimir Merdian-Tarko (Student ID: 7325771)

**Study program**: M. Sc. Business Administration, Media and Technology Management

**Supervisor**: Univ.-Prof. Dr. Jörn Grahl

Department of Digital Transformation and Value Creation

Faculty of Management, Economics and Social Sciences

University of Cologne

May 02, 2019

**Contents**

1. Introduction
2. Methodology
3. Related Work
   1. General Comparative Studies
   2. Comparative Studies Focused on Customer Journey Prediction
4. Model Evaluation Framework
5. Experiments
   1. Experimental Setup
   2. Data
   3. Descriptive Statistics
   4. Models
6. Evaluation of Models and Experimental Results
   1. Objectivity
   2. Predictive Accuracy
   3. Robustness
   4. Interpretability
   5. Versatility
   6. Algorithmic Efficiency
7. Discussion and Managerial Implications
8. Conclusion

Appendix

References

List of Figures

Figure 1 – Visits, purchases, page views and product views aggregated by week

Figure 2 – AUC learning curves for all models and samples

Figure 3 – F-score learning curves for all models and samples

Figure 4 – Training times for all models and samples

Figure 5 – Testing times for all models and samples

List of Tables

Table 1 – Overview of general comparative studies

Table 2 – Overview of comparative studies focused on customer journey prediction

Table 3 – Descriptive statistics of data processing stages

Table 4 – Descriptive statistics of all samples

Table 5 – Average ranks of all models based on AUC and F-Score

Table 6 – Summary of model evaluation and comparison

Table 7 – Model abbreviations

Table 8 – Description of metric groups

Table 9 – *F-* and *p-values* of all features

Tables 10 to 14 – Top 10 features for LR, SVM, DT, RF and BOOST

Tables 15 to 24 – Predictive performance metrics for all models and samples

Tables 25 to 28 – Robustness metrics for all models and samples

1. Introduction

The abundance of different data and the analysis of these data are becoming increasingly critical factors for success in the field of marketing (Wedel and Kannan, 2016). Particularly clickstream data that capture users’ behavior on websites used to generate insights for marketing purposes gained traction in recent years and provide a multitude of opportunities for applications in marketing (Bucklin and Sismeiro, 2009). One such opportunity constitutes customer journey prediction where a customer’s purchasing intent is to be predicted based on her behavior captured by clickstream data, contributing to the business goal of raising sales and profits (Sheil et al., 2018, p. 1).

In addition to the growing amount and importance of data and analytics in marketing, new models keep getting added to the marketing practitioner’s toolbox over time (Wedel and Kannan, 2016, pp. 100-101). Machine learning, for example, can be used to predict conversions in e-commerce environments using clickstream data (e.g. Boroujerdi et al., 2014; Sarwar et al., 2015; Sakar et al., 2018). Deep learning, a sub-category of machine learning, attracts growing attention having achieved several breakthroughs across different domains in the recent past, such as long-short term memory recurrent neural networks for sequence prediction (Hochreiter and Schmidhuber, 1997), convolutional neural networks for image recognition (Krizhevsky et al., 2012) and transformer encoders for language understanding (e.g. Devlin et al., 2018). Recurrent neural networks in particular pose an interesting type of model to predict conversions in an e-commerce setting since they are sequence-based models and clickstream data essentially consist of sequences of user behavior (Lang and Rettenmeier, 2017, p. 1). Wu et al. (2015), Lang and Rettenmeier (2017), Toth et al. (2017) and Sheil et al. (2018) successfully apply recurrent neural networks to predict customers’ purchasing intentions and find them to be superior compared to other machine learning models in their experiments.

Evidently, the choice of available models is vast and their numbers growing and so the selection and application of the right model is not an easy endeavor for marketing executives and accompanied by numerous caveats (Little, 1970; Lodish, 2001; Little, 2004; Lilien, 2011). Therefore, the objective of this thesis is to compare a range of machine learning and deep learning models for customer journey prediction in e-commerce, considering the perspective of marketing executives who use models as support for their decision making. Thus, the research question is: How do different machine learning and deep learning models perform on the problem of predicting customers’ purchasing intentions in terms of criteria that are relevant to marketing executives? In addition, this thesis investigates whether typically more complex deep learning models like recurrent neural networks are superior compared to other machine learning models for predicting conversions.

To achieve this objective and to answer the research question, the methodology for conducting a sound comparative study consists of an approach that is divided into three parts. First, a multi-dimensional meta-analysis of related comparative machine learning studies is conducted to motivate the choice of models. Second, the models selected in the meta-analysis, namely logistic regression (LR), a decision tree classifier (DT), naïve Bayes (NB), k-nearest neighbors (KNN), random forest (RF), a support vector machine (SVM), a gradient tree boosting classifier (BOOST), neural networks with one (NN1), three (NN3) and five (NN5) hidden layers, respectively, a recurrent neural network (RNN) and a long-short term memory network (LSTM), are tested in large-scale experiments on real-world clickstream data. Third, the models and experimental results are evaluated using a theory-backed model evaluation framework that incorporates the following six criteria: *objectivity*, *predictive accuracy*, *interpretability*, *robustness*, *versatility* and *algorithmic efficiency* (Anderl et al., 2014, p. 7-10).

This thesis contributes to research by deriving general tendencies and insights from comparative research concerned with machine learning and deep learning from the past 30 years. This is accomplished through a comprehensive multi-dimensional meta-analysis considering multiple dimensions, such as models, data and evaluation metrics. The practical contribution constitutes large-scale experiments using twelve models, ten clickstream data samples and several evaluation metrics followed by a thorough evaluation and comparison of the models along a framework based on management and marketing research. This model evaluation framework is specifically designed to evaluate models considering criteria that are relevant to marketing executives who are typically the recipients of such models and goes beyond evaluating and comparing models using a single quantitative metric. Implications of this thesis for marketing executives are presented as well.

Section 2 describes the methodology applied to conduct this thesis; Section 3 presents and analyzes related studies and motivates the choices of models, metrics and methods to be implemented and investigated in the experiments; Section 4 explains the framework selected to evaluate the models used to predict customers' purchasing behavior; Section 5 introduces the experimental setup, the data and the models in detail; Section 6 evaluates the models and experiments considering the criteria of the previously introduced evaluation framework; Section 7 discusses the findings of the experiments and derives implications for marketing executives; Section 8 finally concludes with a summary and an outlook for future research.

1. Methodology

***Meta-analysis of related work***. First, related studies that compare different machine learning and deep learning models are analyzed. The first part of this meta-analysis comprises general studies comparing models for classifications tasks for a variety of different use cases and data sets while the second part specifically focusses on studies that compare models for the application of customer journey prediction. To keep the meta-analysis concise, the most relevant comparative studies have been selected based on a catalogue of specific criteria to be explained at the beginning of the meta-analysis. This procedure results in 15 studies being selected for the general part of the meta-analysis and ten studies being selected for the second, application-centric part of the meta-analysis. Both parts examine different dimensions of the selected studies, such as the choice of models, the size and type of the data sets and the choice of metrics to evaluate and compare model performance. The objective of the meta-analysis is to leverage findings of previous comparative studies on the one hand and to identify successful and frequently used models on the other hand to serve as the foundation for the model selection for the experiments. This approach allows to answer the question whether there are models or model families that tend to dominate others.

***Model evaluation framework***. Second, a model evaluation framework is introduced that builds on what management and marketing literature found to be important criteria for models to be successfully used in practice by marketing executives (e.g. Little, 1970; Lodish, 2001; Little, 2004; Lilien, 2011). Anderl et al. (2014) condense this research into a framework they use to evaluate online attribution models for mapping customer journeys. Their evaluation framework consists of six criteria: *objectivity*, *predictive* *accuracy*, *robustness*, *interpretability*, *versatility* and *algorithmic efficiency* (Anderl et al., 2014, p. 7-10). Although Anderl et al.’s (2014) framework has been designed to evaluate attribution models, it can be applied to evaluate machine learning and deep learning models for customer journey prediction as well since it is based on literature that deals with marketing models and criteria for their successful application in general. The objective of using this multi-level evaluation framework is to allow for a thorough comparison of different models beyond the calculation of single quantitative metrics but including qualitative criteria that are particularly relevant to marketing executives as well.

***Large-scale experiments on real-world data***. Third, twelve models, chosen as a result of the meta-analysis, are evaluated and compared on different samples of real-world clickstream data from a Swiss e-commerce website in terms of the previously introduced six criteria of the model evaluation framework. The objective of this large-scale comparison is to enable a comprehensive and thorough evaluation of machine learning and deep learning models for customer journey prediction considering criteria beyond just predictive accuracy that are relevant to marketing executives.

1. Related Work

Section 3.1 presents a selection of related studies that compare different machine learning and deep learning models in general, mainly comparing their predictive performance and other metrics on classification tasks for a variety of different use cases and data sets. Section 3.2 is explicitly focused on studies that compare machine learning and deep learning models for customer journey prediction (i.e. purchase prediction) using e-commerce clickstream data. Both sections first introduce the criteria applied to select the most relevant studies from a large body of comparative literature. Then, the selected studies and their general findings are presented while the most notable are particularly highlighted. Finally, both sections present the most frequently used models, respectively, which form the foundation for the choice of models to be implemented in the experiments. The objective of Section 3 is to conduct an analysis of existent comparative machine learning and deep learning studies on a meta-level to leverage previous research in this field and to form a sound foundation for the choice of models, evaluation metrics and other analytical methods.

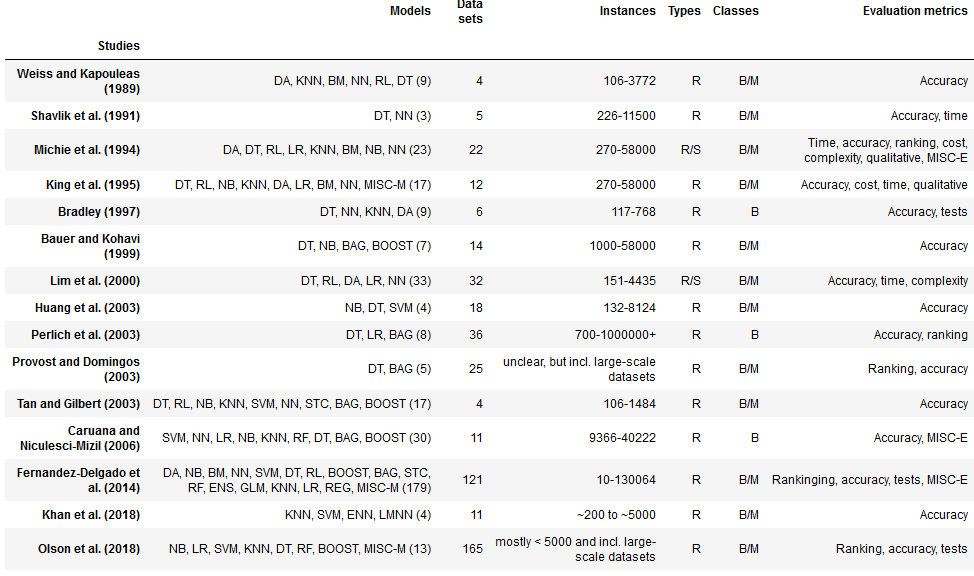
* 1. General Comparative Machine Learning Studies

The body of literature that compares machine learning models is vast, comprising many dozens of studies spanning a broad range of academic disciplines, such as finance, medicine and the natural sciences. For the sake of conciseness, a catalogue of quantitative and qualitative criteria is applied to identify the most relevant and notable studies conducted in the field of comparative machine learning research in the past 30 years. The subsequent meta-analysis of comparative machine learning studies is therefore focused on studies that explicitly apply three different supervised machine learning or deep learning model or families of models on at least two real-world data sets. The models of choice may stem from one family but must differ in the sense that they are not superficially modified variations of one and the same algorithm. Thus, they must entail differences that yield a substantially different model instead. Besides, the respective studies must not only explore classification problems with multiple classes but binary classification problems as well. Moreover, since Section 3.1 is meant to take a general stance toward comparative machine learning research, it mainly includes studies that themselves compare different models in a general setting regarding the problems and data at hand rather than being focused on specific applications, data or fields of research. By applying these criteria, the vast body of comparative machine learning literature is condensed to the studies that are not only the most relevant to this thesis but also among the most notable overall.

Table 1 presents 15 comparative studies that meet the criteria above. The author(s) and the year of the study’s publication are in the first column, the models and model families (with the total number of models used in parenthesis) are in the second column, the number of data sets, the range of the number of instances, the type of the data sets (R for real and S for synthetic data) and the classification problem at hand (B for binary and M for multi-class) are in the third, fourth, fifth and sixth column, respectively. The metrics and methods used to evaluate the models are in the seventh column.

To facilitate the comparison of the studies, the models and evaluation metrics are mildly generalized and grouped. If a study develops a custom model, it is tagged as such (CM). If rather esoteric models are used that do not appear in many other studies, they are combined into a miscellaneous group (MISC-M). Abbreviations are used for all other models and families of models. A great variety of different metrics is used in the considered studies for performance and model evaluation, which is why they are summarized to the following categories: *accuracy*, *complexity*, *cost*, *miscellaneous*, *qualitative*, *ranking*, *test* and *time*. Metrics such as accuracy, error, AUC, precision, recall and F-score are grouped under *accuracy*. *Complexity* captures a model’s complexity, for example considering the number of hidden layers or neurons in a neural network or the number of a leaves in a decision tree. If a certain cost is associated with misclassifying observations for instance, it is captured by *cost*. Some studies use qualitative metrics to describe a model’s ease of use for example and are therefore tagged as *qualitative*. Other studies use different ranking methods that are combined in *ranking*. Certain statistical tests are used to evaluate different models and are grouped under *test*. Finally, the time a model requires for training, testing or classifying an unseen observation is captured by *time*. The remaining metrics that cannot be grouped into one of the categories above are tagged *miscellaneous* (MISC-E). This approach allows to paint a general picture of models and metrics used rather than being left with an extensive list of a multitude of, occasionally esoteric, models and metrics, many being similar in fact but just differently named. The abbreviations and metric groups are listed in Tables 7 and 8 in the Appendix, respectively.

Table 1



Twelve of the considered 15 studies have been conducted in the period from 1989 to 2006. Then, following a gap of eight years, three studies have been conducted in the period from 2014 to 2018. Ten studies have been published in books or scientific journals (e.g. Machine Learning and the Journal of Machine Learning Research) while five have been published in the context of different conferences (e.g. the International Conference on Machine Learning and the IEEE International Conference on Data Mining).

The average number of models per study is 24 and the median of models per study is nine. The according standard deviation is 44 models. The relatively high standard deviation originates from the fact that there are studies that use dozens of models (Lim et al., 2000; Caruana and Niculescu-Mizil, 2006) or even more than one hundred models (Fernandez-Delgado et al., 2014), but there are also studies that focus on only a handful of models (e.g. Shavlik et al., 1991; Huang et al., 2003; Khan et al., 2018). Decision trees (DT) are the most frequently used models with 14 occurrences. They are followed by k-nearest neighbors (KNN) and neural networks (NN) with nine each, naïve Bayes (NB) with eight and logistic regression (LR) with seven occurrences.

A similar observation is made for the number of data sets used with an average of 32, a median of 14 and a standard deviation of 47 data sets. Again, there are studies that use dozens of data sets (Lim et al., 2000; Perlich et al., 2003; Provost and Domingos, 2003), some use even more than one hundred data sets (Fernandez-Delgado et al., 2014; Olson et al., 2018), but there are also studies that use a few data sets only (e.g. Weiss and Kapouleas, 1989; Shavlik et al., 1991; Bradley, 1997; Tan and Gilbert, 2003). The typical number of instances per data set ranges from as few as a couple of hundred to as many as a couple of thousand or tens of thousands, but there are only few large-scale data sets with hundreds of thousands or more than one million instances (Perlich et al., 2003; Provost and Domingo, 2003; Olson et al., 2018). Most studies explicitly justify their selection and explain the characteristics of the data sets used, but some studies make it difficult to understand what the data they use is exactly like (Provost and Domingos, 2003; Khan et al., 2018). Bauer and Kohavi (1999, p. 113) require their data sets to contain at least 1000 instances to make reliable assessments of models on sufficiently large test sets. Perlich et al. (2003, p. 214) use only data sets with at least 700 instances to be able to get learning curves of a reasonable length. Just two of the considered studies use synthetic data in addition to real-world data sets (Michie et al., 1994; Lim et al., 2000). The analysis shows that there are several data sets that are used in multiple studies, typically taken from the University of California, Irvine Machine Learning Repository. Just three studies exclusively consider binary classification problems (Bradley, 1997; Perlich et al., 2003; Caruana and Niculescu-Mizil, 2006) while the remaining twelve studies additionally consider multi-class problems.

All 15 studies use at least one evaluation metric related to accuracy. Nine out of 15 studies use metrics from multiple groups. Ranking metrics are used by five, time is used by four and tests are used by three studies (Table 1, column seven). Only two studies evaluate models using metrics related to complexity (Michie et al., 1994; Lim et al., 2000). Likewise, only two studies use qualitative criteria for model evaluation and comparison, e.g. Michie et al. (1994) craft a user guide for model evaluation and selection and King et al. (1995) evaluate models in terms of their ease of use.

Although the studies conducted by Michie et al. (1994) and King et al. (1995) may seem dated, they are still highly relevant. They not only compare and evaluate a multitude of different models on a variety of data sets using several evaluation metrics, but they use STATLOG, a project on the performance evaluation of machine learning, neural and statistical algorithms on real-world data sets funded by the European Commission in the 1990s (European Commission, 1994), as the foundation of their research. More recent noteworthy, thorough studies that are conducted on a large scale in terms of the number of models and/or data sets used are Lim et al. (2000), Caruana and Niculescu-Mizil (2006), Fernandez-Delgado et al. (2014) and Olson et al. (2018).

In general, there is no single model that outperforms all other models, but model performance is highly dependent on the given problem and data set (Salzberg, 1999, p. 11). Most studies explicitly confirm this statement (e.g. Michie et al., 1994; King et al., 1995; Bradley, 1997; Huang et al., 2003; Caruana and Niculescu-Mizil, 2006; Olson et al., 2018). Although Lim et al. (2000) state that there are no statistically significant differences between many models they evaluate, they claim that there are huge differences in training time and interpretability though. Some studies that compare only a few models derive more differentiated conclusions. Bauer and Kohavi (1999) find that bagging (BAG) generally outperforms boosting (BOOST) while both perform better compared to DT and NB – however, at the cost of interpretability since BAG and BOOST are more complex and less interpretable models. Perlich et al. (2003) state that LR tends to perform better for smaller data sets while DT tends to perform better for larger data sets. In a similar nuanced fashion, Tan and Gilbert (2003) find that support vector machine (SVM) and NN tend to perform much better over multi-dimensional and continuous features while DT and rule-based learning (RL) tend to perform better on discrete or categorical features. They also find that ensembles of models outperform individual models, but again, no model outperforms all others in every situation (Tan and Gilbert, 2003). According to Caruana and Niculescu-Mizil (2006), although random forest (RF), NN and specialized variants of BAG, BOOST and SVM perform best and NB, LR, and DT perform worst, there is no single model that outperforms all others on every problem and data set. Fernandez-Delgado et al. (2014) confirm these results, adding that RF is clearly the best model family followed by SVM, NN and BOOST. These results are at least partly confirmed by Olson et al. (2018) who find BOOST and RF to perform well while NB performs poorly in general. They additionally recommend SVM, LR and an Extra Tree Classifier, but state that certain variants or modifications of these and other models are not competitive at all, such as KNN, some NN, some BOOST and some DT (Olson et al., 2018).

There are clearly no models that strictly dominate all others. But there are certain tendencies (e.g. ensembles tend to outperform individual classifiers). Therefore, the selection of models for the experiments is based on the number of total occurrences of a model across all studies. This objective choice is supported by the assumption that over the course of time, research is likely to apply established and reliable models rather than those who cannot prove themselves. As a consequence, the frequently used models are likely to be the ones who can hold their ground against other less prominent and therefore less successful models. Thus, to capture the general notion of the selected studies, the pre-selected models entail DT, KNN, NN, NB and LR. Section 3.2 extends this pre-selection by taking into account the marketing perspective and the application of customer journey prediction.

Since the objective of the meta-analysis is to derive a general picture of the comparative studies selected, it goes beyond the scope of this thesis to describe and analyze all selected studies shown in Table 1 in detail. Nevertheless, it is worth mentioning that most studies use some kind of resampling technique, such as holdout or cross-validation, because some data sets that are used have only a couple of hundred or thousand instances. Several preprocessing, feature engineering and selection techniques as well as techniques to measure the effect of sample size, such as learning curves (e.g. Lim et al.; 2000; Perlich et al., 2003), are also occasionally applied. A selection of these techniques applied in the experiments is inspired by the meta-analysis.

* 1. Comparative Machine Learning Studies Focused on Customer Journey Prediction in E-Commerce

While Section 3.1 presents comparative machine learning research from a general perspective, Section 3.2 explores studies directly related to the application of customer journey prediction, comparing machine learning and deep learning models for purchase prediction using e-commerce clickstream data. Since the amount of studies for this specific application is small compared to the body of literature concerned in Section 3.1, slightly different criteria are applied to the selection of relevant studies. Studies must be concerned with customer journey prediction (i.e. purchase prediction) using clickstream data from an e-commerce website. Further, studies must still focus on applying supervised machine learning and deep learning models, but it suffices if they use two different models or model families. This approach allows for the selection of studies that are closely related to the use case implemented in the experiments, which is desirable because methods and findings from these studies can be leveraged and transferred to the experiments more easily.

It is worth noting that other models apart from machine learning and deep learning are successfully used for purchase prediction on clickstream data. The following mentions a selection of such studies. Moe et al. (2002) develop a Bayesian tree model that groups customers based on their behavior and examines their purchasing decision based on in-store experiences at the same time, which they find to be superior to a latent class logit model. Montgomery et al. (2004) find that a dynamic multinomial probit model better predicts path information than traditional multinomial probit and first-order Markov models, leading to an increase in the accuracy of predicting conversions compared to the benchmark that does not include path information. Sismeiro and Bucklin (2004) model conversions via a task competition approach by linking what customers do and what they are exposed to on an e-commerce website. Baumann et al. (2018) use clickstream data to build graphs of visitor sessions and use corresponding graph metrics to show their importance for predicting purchase events.

Table 2 presents ten studies that meet the criteria above. The author(s) and the year of the study’s publication are in the first column, the models and model families (with the total number of models used in parenthesis) are in the second column, a description of the data sets, the target and the metrics used to evaluate the models are in the third, fourth and fifth column, respectively.

Two studies have been published in 2004 while the other eight have been published in the period from 2014 to 2018, signaling a recent rise in the popularity of comparative machine learning and deep learning research concerned with the application of customer journey prediction. Three studies have been published in scientific journals (e.g. Management Science and Neural Computing and Applications), four studies have been published in the context of different machine learning challenges and workshops (e.g. the International ACM Recommender Systems Challenge and the ACM SIGIR Workshop on eCommerce) and three studies have been published on the Cornell University’s preprint document server arXiv.org.

Most studies use two to six different models while only a single study uses 16 models (Boroujerdi et al., 2014). The average number of models per study is five, the median is four and the standard deviation is equal to about 4 as well. LR ranks on top with six occurrences, followed by NN with five, RF and recurrent neural networks (RNN) with four each and DT and BOOST with three occurrences each.

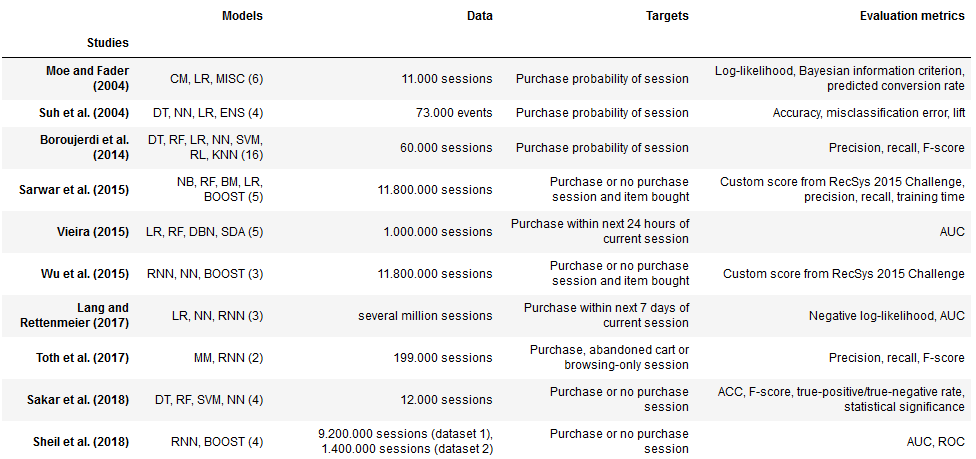
Nine out of ten studies use a single data set while only one study uses two data sets (Sheil et al., 2018). From some studies it is difficult to deduce the exact amount of data they use, e.g. for business reasons (Lang and Rettenmeier, 2017, p. 3) or because it is not entirely clearly stated (Vieira, 2015). The amount of data used in the selected studies varies widely from as few as a couple thousand sessions (i.e. clearly defined visits to an online shop) over a timespan of several weeks or months (Moe and Fader, 2004; Suh et al., 2004; Boroujerdi et al., 2014; Toth et al., 2017; Sakar et al., 2018) to as much as several million sessions in a few weeks up to several months (Sarwar et al., 2015; Vieira, 2015; Wu et al., 2015; Lang and Rettenmeier, 2017; Sheil et al., 2018).

In the cases where the conversion rate is reported, it ranges from as low as less than one percent (e.g. Sheil et al., 2018) to as high as more than ten percent (e.g. Moe and Fader, 2004; Sakar et al., 2018). In other cases, the reported conversion rate lies between two and six percent (e.g. Suh et al., 2004; Sarwar et al., 2015; Vieira, 2015).

Three studies predict the probability of an individual session leading to a purchase (Moe and Fader et al., 2004; Suh et al., 2004; Boroujerdi et al., 2014) while two studies predict whether a session leads to a purchase or not (Sheil et al., 2018; Sakar et al., 2018). In addition to the latter type of prediction, Sarwar et al. (2015) and Wu et al. (2015) predict the item(s) likely to be bought in a session. Vieira (2015) and Lang and Rettenmeier (2017) extend the time span within which a session is counted as a conversion (i.e. purchase event) for their prediction to 24 hours and seven days, respectively. Toth et al. (2017) are the only study modeling customer journey prediction as a multi-class problem, splitting their target into three classes, namely *purchase*, *abandoned cart* and *browsing-only*.

Since the number of different evaluation metrics used in the studies in Table 2 is comparably small they are not summarized. While most studies use some form of accuracy metric (see Table 2, column five), other studies use different metrics to evaluate their models’ performance as well. For example, Moe and Fader (2004) and Lang and Rettenmeier (2017) use negative log-likelihood and Sarwar et al. (2015) and Wu et al. (2015) use a custom metric specifically tailored to the RecSys Challenge 2015 (Ben-Shimon et al., 2015). Sarwar et al. (2015) are the only ones to report the time their models required for training. Only Sakar et al. (2018) explicitly report *p-values* of statistical significance tests they run to evaluate model performance.

Table 2



Contrary to the studies under consideration in Section 3.1, the studies considered in Section 3.2 present more definite results, which seems intuitive given that they use less models on average and typically focus on a single problem and data set. Moe and Fader (2004) report that their custom model outperforms all benchmark models under consideration, among which is LR. Suh et al. (2004) and Boroujerdi et al. (2014) find that an ensemble created from their other models outperforms all individual models. Wu et al. (2015) and Toth et al. (2017) explicitly explore recurrent neural networks (RNN) for purchase prediction and find that they perform best in their experiments. Lang and Rettenmeier (2017) use an RNN with long-short term memory (LSTM) that is better at capturing long-term dependencies and conclude that LSTM reduce the need for extensive feature engineering, yield increased predictive performance and improve interpretability of predictions. Sheil et al. (2018) find that LSTM performs best for one data set while BOOST performs best for the other data set they use. Sarwar et al. (2015) find that BOOST outperforms all other models for the prediction of sessions that lead to a purchase. Vieira (2015) builds another specialized NN, namely a stacked denoising auto-encoder, that he reports performs best in his experiments. Finally, Sakar et al. (2018) find NN to yield better performance than RF and SVM. Although one might believe to recognize certain patterns, it might be too rash to conclude that complex models, such as different variants of NN or BOOST, generally yield better results. The claim that model performance heavily depends on the specific problem and data at hand, is likely to hold true in this specialized context as well, given the diversity in data used and preprocessing techniques applied across the studies in Table 2. Besides, the studies under consideration in Section 3.2 might be biased toward models that have gained popularity just recently (e.g. RNN) given that most of them have been conducted in the period from 2014 to 2018.

The most frequently used models, jointly considering Sections 3.1. and 3.2., are selected to be used in the experiments to include both frequently used models in general as well as models that tend to be frequently and successfully used for customer journey prediction. Besides, this procedure helps to mitigate the potential bias toward models that have gained popularity particularly in recent years. The selected models are DT with 17, NN with 14, LR with 13, KNN with nine, NB with eight and finally RF and RNN with four occurrences each in total across both Sections 3.1. and 3.2. Although BOOST and SVM each appear at the lower end of the frequency rankings in both sections, the sum of their occurrences across both sections is equal to eight. Therefore, to allow for a more comprehensive and generalized evaluation, both are considered in the experiments.

As mentioned in Section 3.1, a variety of preprocessing and feature engineering and selection techniques along with techniques for measuring the effect of sample size is applied across the selected studies shown in Table 2 as well. For example, it is noteworthy that Lang and Rettenmeier (2017) claim that RNN is capable of reducing the need for manual feature engineering because this type of model is able to use information contained in past sessions automatically. Given the similarity of the problem explored in the selected studies and this thesis and consequently the relevance for the experiments, it is worth referring to some techniques which is done in Section 5 in detail.

1. Model Evaluation Framework

Anderl et al. (2014) condense, among other, the research on the application and acceptance of marketing models into an evaluation framework they use to assess online attribution models in a comprehensive and concise manner. Their evaluation framework comprises six criteria: *objectivity*, *predictive accuracy*, *robustness*, *interpretability*, *versatility* and *algorithmic efficiency* (Anderl et al., 2014, pp. 7-10). The criterion of ***objectivity*** is defined as a model’s ability to assign credit to specific features in the data that factually contribute to the objective of the application the model is applied to, e.g. increasing the number of conversions or revenue (Anderl et al., 2014, p. 7). Objectivity originates from Lilien’s (2011, p. 198) claim for a model to allow for the computation of a variable’s relative impact and the objective evaluation of available decision options. ***Predictive accuracy*** is defined as a model’s ability to correctly predict conversions (Anderl et al., 2014, pp. 8), picking up Lodish’s (2001, p. 54) lesson of the importance of a model’s credibility to persuade managers. ***Robustness*** is defined as a model’s ability to deliver “(…) stable and reproducible results (…)” after multiple runs of the model (Anderl et al., 2014, pp. 8), covering Little’s (1974, p. 470; 2004, p. 1843) requirement for a model to return useful results. According to Little (1974, p. 470; 2004, pp. 1843-1844), models should be simple and easy to communicate with, which Anderl et al. (2014) translate to the criterion of ***interpretability***, defined as the fact that a model’s structure and results should be transparent and understandable to all stakeholders involved with reasonable effort (p. 8). ***Versatility*** incorporates Little’s (1974, p. 470; 2004, pp. 1843-1844) requirements that models should be easy to control and to adapt, i.e. models should allow for the inclusion of novel information and data in rapidly and frequently changing environments through a high degree of flexibility (Anderl et al., 2014, p. 10). ***Algorithmic efficiency*** builds upon Lodish’s (2001, p. 54) lesson that models should ideally deliver results on-demand, i.e. when managers need them, which is particularly important when dealing with large amounts of data (Anderl et al., 2014, p. 10).

There appears to be a divide between the models developed for marketing decision support in academia and their actual application by practitioners in the field (Lilien, 2011) and in addition, the most complex model does not necessarily turn out to be the one that has the largest impact on a company (Anderl et al., 2014, p. 7). Lodish (2001, p. 54) puts it like this: "The criterion for a good, productive model is not whether it is theoretically or empirically perfect. It is, will the manager's decision, based on the model, improve productivity enough to justify the costs and resources devoted to developing and using the model?". Anderl et al.’s (2014) model evaluation framework builds upon these insights by incorporating not only quantitative metrics but also emphasizes the importance of criteria that capture dimensions that are relevant for marketing executives to actually apply models in practice (Anderl et al., 2014, p. 8).

Although Anderl et al. (2014) design their framework to evaluate online attribution models, it generalizes well given that it builds upon research that explores the application and requirements of marketing models in general. Therefore, their evaluation framework can be transferred to the evaluation of machine learning and deep learning models for the application of customer journey prediction. The framework’s six criteria are applied in Section 6 to evaluate the experiments on predicting customers’ purchasing behavior in detail, using a multitude of machine learning and deep learning models and respective performance metrics.

1. Experiments

Section 5.1 presents the setup of the experiments and names the tools and software packages used to conduct the experiments. Section 5.2 provides detailed information on the data used in the experiments, the choices made during processing and the techniques applied to transform the raw data into training and test sets. Section 5.3 first defines important concepts related to clickstream data and then analyzes the data in a descriptive manner to generate first insights for customer journey prediction. The target and features used in the experiments are examined in more detail as well. Section 5.4 finally introduces the models used in the experiments, stating and justifying important choices that are made regarding implementation, parameter choice, training and testing. For the sake of brevity, the models are not explained in great depth, but secondary references are provided for the interested reader. The objective of Section 5 is to explain the experiments’ setup and the reasoning behind the choices of models, target, features, training and test sets and methods applied to process and model the data.

* 1. Experimental Setup

A selection of twelve models is derived from the meta-analysis in Section 3: LR, DT, NB, KNN, RF, SVM, BOOST, NN with one (NN1), three (NN3) and five (NN5) hidden layers, respectively, RNN and LSTM. Increasing the number of hidden layers in NN increases complexity and allows to observe whether predictive performance increases with complexity for this particular model.

The studies presented in Section 3.2 use different targets for predicting conversions, all of which are justifiable and each having different merits and drawbacks. One possibility is to predict whether a given session leads to a conversion (e.g. Sheil et al., 2018; Sakar et al., 2019). A natural extension to this approach would be to predict the probability of a given session leading to a conversion (e.g. Moe and Fader, 2004; Suh et al., 2004; Boroujerdi et al., 2014). Another approach is to allow for a conversion to happen within a defined time window, say within the next 24 hours or seven days of a given session (Vieira, 2015; Lang and Rettenmeier, 2017). The latter approach is selected, creating a target that captures conversions within the next 24 hours of a given visit. This approach builds on the assumption that multiple sessions within a specified time window can contribute to a conversion, i.e. assuming that customers tend to purchase more frequently after multiple sessions and seldomly after single, isolated sessions.

To measure the effect of sample size, inspired by Shavlik et al. (1991), Lim et al. (2000), Perlich et al. (2003), Moe and Fader (2004) and Vieira (2015), ten clickstream data samples of different sizes are used in the experiments, ranging from a couple of thousand to more than one million unique visitors per sample. Only few studies considered in Section 3.2 explicitly state how they sample their data or create their training and test sets. For example, Wu et al. (2015) decide for a train test split ratio of three to one and add every fourth session to the test set while Lang and Rettenmeier (2017) instead use the first three weeks in their data for training, the following week for validation and the subsequent two weeks for testing. For the following experiments, unique visitors are randomly selected from the entire data set to ensure a stratified class distribution and a generally balanced distribution of attributes over the entire period represented in the data. To allow for a smoother comparison and measurement of the sample size’s effect on predictive performance, the samples partly overlap regarding the unique visitors they entail. For example, the 500,000 unique visitors sample contains all the unique visitors contained in the 250,000 unique visitors sample and additionally 250,000 different unique visitors. Each sample is split in the fashion that the resulting training and test sets contain distinct unique visitors. Thus, the training set contains 80 and the test set contains 20 percent of a sample’s unique visitors, resulting in a train test split ratio of four to one. The number of unique visitors is used to specify the size of a sample instead of the actual number of sessions (i.e. instances) in a sample. This seems practical and reasonable given that the number of unique visitors per sample is roughly proportional to the number of sessions per sample (Section 5.3, Table 4). Creating training and test sets in this manner allows to leverage the entire timespan of the data without cutting individual customer journeys (i.e. a visitor’s entire sessions are contained either in the training or the test set but not split across both), thus capturing customer journeys and corresponding behavior in their entirety.

The experiments are conducted on a Linux workstation with 32 CPUs and 125 GB of memory, running on Ubuntu 18.04. Python 3.6.7 is used for working with the data and models in general. The Python machine learning library Scikit-learn 0.20.2 (Pedregosa et al., 2011) is used to build LR, DT, NB, KNN, RF, SVM and BOOST. The Python deep learning library Keras 2.2.4 (Chollet and others, 2015) with a TensorFlow (Abadi et al., 2015) backend is used to build NN, RNN and LSTM. Further details on the experiments, the data and the models are provided in the remainder of Section 5.

* 1. Data

The data used for this thesis stem from a Swiss e-commerce website, comprising 63 GB and spanning six months from May to October 2016. The data can be seen as a sequence of events that capture visitors’ clicking behavior and additionally contain visitor-level information, such as device type, operating system and the marketing channel via which the visitor came to the online shop. Each row in the raw data represents an event (e.g. page view, product view, addition of a product to the shopping cart, purchase etc.) tagged with a timestamp and additional information that are registered by the tracking software implemented on the website. A visitor or customer is an individual that visits an e-commerce website, to browse the online shop’s product catalogue or to purchase a specific product for example. An event or hit constitutes a visitor’s specific action during her visit, e.g. a page or product view, a product’s addition to the shopping cart or the purchase of a specific product. Every event is tagged with a timestamp and contains further event-specific information, such as a product’s price or the product category it belongs to as well as visitor-specific information, such as login status, gender or age. A session or visit is a well-defined sequence of subsequent events that lay no further apart than 30 minutes while the maximum amount of time between the first and the last event in a session is twelve hours. The terms session and visit are used interchangeably. A purchase is also called a conversion and the conversion rate is therefore defined as the ratio of the number of conversions and the number of sessions in a given period.

***Cleaning and mapping***. There are 29.5 million rows in the raw data, ranging from 3.4 to 6.7 million rows per month. The number of rows in the raw data is reduced by about 1.8 percent after cleaning the raw data (e.g. dropping rows with missing values or broken records). Missing values in the remaining rows are filled according to context (e.g. tagged as *Not Specified* if applicable). The amount of removed rows varies from 0.3 to 3.7 percent per month. There are 138 columns in the raw data of which 42 are considered to contain useful information. After splitting the observations in certain columns, the number of columns increases to 48. Since certain columns are encoded, mappings of codes and strings that represent the actual information are done using special mapping files. Some columns are already casted to the right data type and format in this processing step as well. The number of unique visitors is reduced from 4.7 to 4.5 million in this processing step.

***Aggregation***. Lang and Rettenmeier (2017, p. 5) find in their experiments that predictive performance is only marginally impacted by the data’s level of aggregation. Therefore, and to make the data more manageable in terms of size and granularity, the raw data are aggregated from event to session level. Datetime and numerical columns are aggregated using appropriate aggregators, such as the sum, the count, the minimum or the maximum. Categorical columns are aggregated saving their first occurrence within a session. Examining the difference between aggregating categorical columns by first or by last occurrence could be investigated in additional experiments. Aggregating the data from event to session level leads to the reduction of 29 million rows (i.e. events) to 6.6 million rows (i.e. sessions), still entailing 4.5 million unique visitors. The number of columns increases from 48 to 50.

***Preparing target and features***. Sessions containing only a single event (i.e. bounce sessions) are removed to reduce noise and because more engaged visitors are more interesting from a marketing executive’s perspective, assuming that more active visitors have a higher propensity to purchase. Removing bounce sessions reduces the number of sessions from 6.6 to 3.3 million and the number of unique visitors from 4.5 to 2.5 million. This is a similar but less drastic step compared to Lang and Rettenmeier (2017, p. 4) who only consider customers with a least 15 previous actions and Toth et al. (2017, p. 2) who include only sessions with at least five events.

The target, capturing a purchase within the next 24 hours of a given visit, is generated using the purchase event in the data that indicates whether a session contained a purchase. Categorical columns that capture information like a customer’s operating system or search engine typically contain dozens of levels. Encoding such categorical columns in their raw state would lead to the creation of dozens of dummy variables, drastically inflating the data. Most visitors are generally represented within just several levels of a categorical column while the majority of the long tail of levels does not occur very frequently in the data. Therefore, those levels whose share in the data is less than 0.1 percent of a categorical column’s most frequently occurring level are grouped in a level named *Other*. Then, the categorical columns are one-hot-encoded while each categorical column’s first dummy variable is dropped to avoid the dummy trap of multicollinearity. User age is discretized into six bins (i.e. 14 to 25, 26 to 35, 36 to 45, 46 to 55, 56 to 65 and over 65 years, respectively). Geographical information is largely ignored for the sake of simplicity. It is quite intuitive that it might play a role whether a session takes place on a weekday or weekend and likewise whether a session takes place in the morning or at night. To capture time effects and to control for seasonality, different time features are created indicating month, day of month, day of week and hour of day. To reduce the number of resulting dummy variables, these time features are mildly grouped to a more general representation (e.g.. dummy variables for weekday and weekend instead of dummy variables for each day of the week and dummy variables for morning, afternoon, evening and night rather than dummy variables for every hour of the day).

Behavior during past sessions seems to be at least partly indicative of whether a given session leads to a conversion (e.g. Lang and Rettenmeier, 2017). Therefore, additional features are created, indicating whether a given visitor visited the online shop or purchased a product within the last several hours or days. Similar features are created to capture the number of page views and product views during the last visit. The hypothesis goes that the more engaged (i.e. active) a visitor is not just in a given session but also in previous sessions, the higher the likelihood that a given session leads to a conversion.

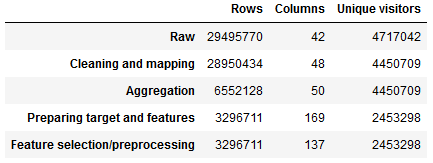
Processing categorical columns and feature engineering increase the number of columns from 50 to 169. Those 169 columns include two identifiers and the target and are not considered in the following paragraph on actual features.

***Feature selection***. Two measures are taken to reduce the large number of features created in the previous steps and to filter out the most relevant ones. First, features that are too closely correlated with the target are removed (e.g. a feature that indicates whether the customer reached the checkout step during a session). Second, although certain models incorporate measures to assess the importance of individual features (e.g. coefficients in LR and SVM or feature importances in tree-based models), relying on those is not practical since they are model-specific. A model-agnostic measure of feature importance is more desirable because many different models are used in the experiments and all should use the same features (with an exception for RNN and LSTM to be explained later). To avoid introducing bias in the modeling stage, feature selection is done on an independent sample that is neither used for training nor for testing. The sample used for feature selection is fairly large, containing roughly 450,000 unique visitors. Its conversion rate is similar to those of the other samples which ensures that its general characteristics are similar to those of the other samples as well. After standardizing numerical features by removing the mean and scaling to unit variance, analysis of variance (ANOVA) *F-values* and according *p-values* are computed for the target and features in the aforementioned sample. To avoid setting an arbitrary threshold to select the *k* best features, only those features are selected whose *F-value* was significant at the one percent significance level, indicated by the according *p-value*. In other words, the *F-test*’s null hypothesis of a given feature not contributing to the prediction of the target is rejected at the one percent significance level for 134 features. Table 9 in the Appendix lists all features and their corresponding *F-* and *p-values*. More in depth explanations of the features are provided in the corresponding tracking software’s reference document (Adobe Systems Incorporated, 2019).

It is important to mention that feature selection is done after encoding categorical features. Consequently, dummy variables representing categorical features’ individual levels are considered rather than categorical features capturing all levels in one feature. As a result, some dummy variables representing levels of categorical features are found to be insignificant and therefore not selected while dummy variables representing other levels of the same categorical features are found to be significant. Another approach could be to in- or exclude categorical features in their entirety instead of measuring the contribution of dummy variables representing individual levels. Thus, there is certainly more room for experimentation regarding feature selection.

***Preprocessing***. The preprocessing steps described in the paragraph above are applied to all other samples as well. In summary, the numerical features are standardized by removing the mean and scaling to unit variance and all but those 134 features identified through feature selection are removed from training and test sets. Table 3 summarizes the number of rows, columns (including identifiers, the target and actual features) and unique visitors after each of the processing stages explained above.

Table 3



* 1. Descriptive Statistics

Table 4 presents descriptive statistics for the ten clickstream data samples used in the subsequent experiments and the entire data set. The evenly spread figures result from the random sampling of unique visitors and indicate that the samples’ attributes are balanced. The number of visits per sample is roughly proportional to the number of unique visitors per sample – the ratio is about three to four. Across all samples, 18 percent of visitors have two or more while only about two percent have five or more visits in the period under consideration. The average number of visits per visitor is 1.3 to 1.4, the according median is 1 and the standard deviation is about 1.8 to 3.8 visits per visitor. The share of buyers among all visitors is about 3.3 to 3.5 percent. About 20 percent of conversions are repeated conversions. The conversion rate across all samples and the entire data set ranges from 2.9 to 3.2 percent. Overall, the smaller samples tend to be slightly less balanced.

The low conversion rate indicates a severe class imbalance which could prove challenging in the experiments. Balancing the class ratio using sampling techniques, such as SMOTE (Chawla et al., 2002), tend to be computationally expensive, especially with large amounts of data. Exploring their usefulness and whether they can significantly improve predictive performance could be an interesting task to be investigated in future research.

Table 4

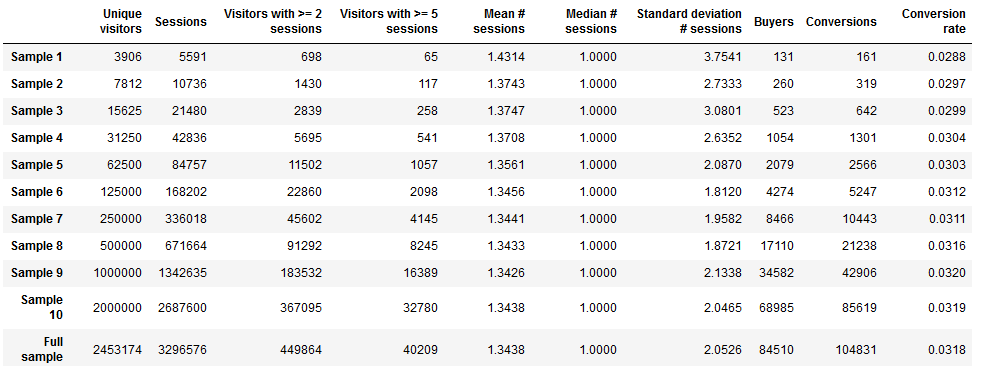
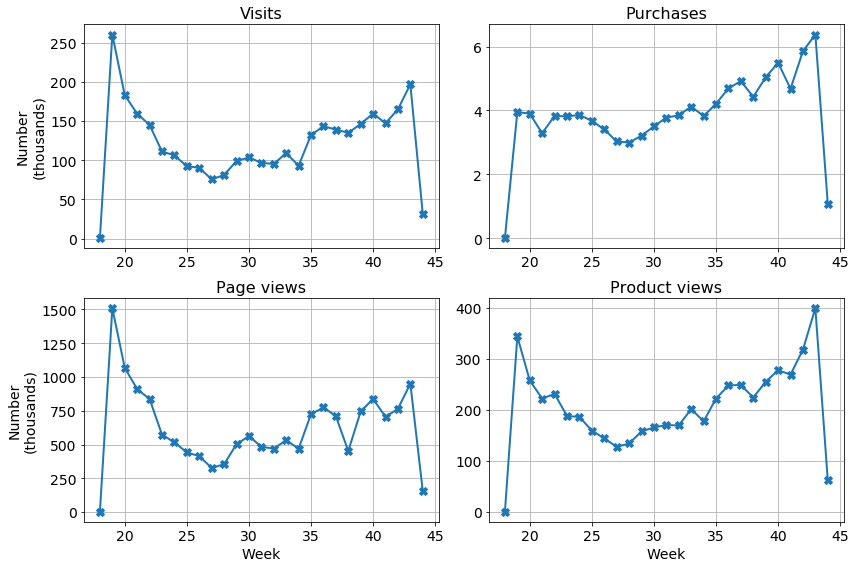


Figure 1 shows the development of activity in the online shop represented by visits, purchases, page views and product views over time. The numbers are aggregated by week and are based on the cleaned, aggregated and processed data. There is a steep drop at the beginning and the end of each line in all four graphs which is caused by the fact that the first and the last week in the data are cut and therefore contain less data points, resulting in lower numbers for visits, purchases, page views and product views for those weeks. Visits, page views and product views first decrease, resulting in a dent in the summer weeks, followed by an increase in the autumn weeks. Purchases instead seem to grow more constantly over the entire period, yet with a small dent in the summer weeks as well. Several peaks are spread over the plots as well. The occasional peaks and the increase of activity toward the autumn weeks may be caused by temporal marketing campaigns or seasonality effects (e.g. holiday season during the summer weeks), respectively.

Figure 1



* 1. Models

Default hyperparameter settings are chosen for most models to maintain a certain degree of comparability. One could argue, however, that some models’ default hyperparameters make them unreasonably superior over other models’ defaults, resulting in a biased comparison. Since the objective of this thesis is to compare a broad range of models, the default hyperparameters are selected as a starting point because optimizing all models would require a considerable amount of resources, which could be a task worth investigating in additional experiments. Therefore, Scikit-learn’s default hyperparameter settings are used for LR, DT, NB, KNN, RF, SVM and BOOST.

LR is a ***logistic regression classifier*** with *l2* penalty, the regularization parameter *C* equal to one and a *liblinear* solver for solving the optimization problem. DT is a ***decision tree classifier*** with a *gini* function to measure the quality of a split, a splitting strategy that choses the best split at each node, an arbitrary maximum tree depth, a minimum of two instances per split, a minimum of one instance per leaf, equal weighting of instances, an unlimited number of leaf nodes and considering all available features. NB is a ***Gaussian naïve Bayes classifier*** without prior probabilities of the classes. KNN is a ***k-nearest neighbors voting classifier*** with the number of neighbors being equal to five, uniform weighting of all neighbors in a neighborhood, automatic selection of the appropriate algorithm to compute the nearest neighbors and the distance metric being Euclidean as specified by the Minkowski metric’s power parameter *p* being equal to two. RF is a ***random forest classifier*** with a forest of ten trees, a *gini* function to measure the quality of a split, an arbitrary maximum tree depth, a minimum of two instances per split, a minimum of one instance per leaf, equal weighting of instances, an unlimited number of leaf nodes and considering all available features. RF is an ensemble that fits several decision trees on various sub-samples of the training set and uses averaging to improve predictive accuracy and to mitigate overfitting. The sub-sample size is equal to the size of the training set, but bootstrap sub-samples are drawn with replacement. SVM is a ***linear*** ***support vector classifier*** with *l2* penalty, squared hinge loss and the regularization parameter *C* being equal to one. These settings tend to scale better to large amounts of data than other SVM implementation with *rbf* kernels for example. BOOST is a ***gradient boosting classifier*** with deviance loss function, a learning rate of 0.1 controlling the contribution of each tree, 100 boosting stages, the entire training set being used for fitting the base learners, mean squared error with improvement score by Friedman to measure the quality of a split, a minimum of two instances per split, a minimum of one instance per leaf, equal weighting of instances, a maximum tree depth of three nodes, an unlimited number of leaf nodes and considering all available features. BOOST builds an additive model by fitting a single regression tree on the negative gradient of the binomial deviance loss function in each of the 100 boosting stages. Detailed information on the respective hyperparameters, implementational details and references for further study are found in the Scikit-learn documentation (Pedregosa et al., 2011).

Building neural networks using Keras requires more decision-making regarding model architecture and hyperparameter settings. Besides, for the sake of comparability, the (recurrent) neural networks are not trained using GPUs, which would have probably increased their training speed, but instead CPUs are used for training all models. However, a binary customized to the workstation’s CPUs enabling AVX2 FMA support is used to speed up training, which might slightly favor neural networks though.

The neural networks with one, three and five hidden layers, respectively, are built in an analogous manner so that the following explanations in this paragraph apply to all three. The input layer and all hidden layers are fully connected and consist of as many neurons as there are features in the training set (i.e. 134) and use a Rectified Linear Unit (ReLU) activation function. The output layer consists of one neuron and uses a Sigmoid activation function for binary classification. The default Xavier uniform initializer (Glorot and Bengio, 2010) is used to initialize the layers’ weight matrices. Binary cross-entropy loss is minimized using the Adam optimizer. The choices of activation functions, weight initialization and the optimizer follow Lang and Rettenmeier’s (2017, p. 4) choices for their NN and RNN. The number of epochs during training (i.e. the number of iterations over the entire training set) is set to ten, following Toth et al. (2017, p. 3). The batch size during training (i.e. the number of instances per gradient update) is set to 256 instances, which is eight times the default batch size of 32. To speed up training if possible, early stopping is configured so that training is ended early if validation loss is not minimized in two subsequent epochs. The choices of batch size and early stopping follow Sheil et al. (2018, pp. 5-6). Using random dropout, 20 percent of units per layer are reset to zero during training to prevent overfitting. Srivastava et al. (2014) suggest a dropout rate of 0.5 for large neural networks, but since the neural networks considered here are comparably small a substantially lower dropout rate is chosen instead. Detailed information on the respective hyperparameters, implementational details and references for further study are found in the Keras documentation (Chollet and others, 2015).

The previously discussed models are vector-based in the sense that they require feature vectors to be of fixed length (Chapelle et al., 2014). Because e-commerce clickstream data represent sequences of varying length of customer behavior over time, these sequences need to be converted through extensive feature engineering into sets of features of fixed length for predicting future customer behavior using vector-based models (Lang and Rettenmeier, 2017, p. 1). RNN instead are able to directly “(…) operate on sequences of varying length and therefore (…)” present a natural fit for purchase prediction in e-commerce (Lang and Rettenmeier, 2017, p. 1). RNN can be extended by more powerful computational LSTM cells that have been originally developed by Hochreiter and Schmidhuber (1997) to address the problem of vanishing gradients and to make the storing of long-term information in RNN architectures possible. Wu et al. (2015), Lang and Rettenmeier (2017), Toth et al. (2017) and Sheil et al. (2018) use such LSTM cells in their RNN. Both RNN and LSTM are explored in the experiments to investigate the potential superiority of LSTM over traditional RNN and the importance of storing long-term information in customer journey prediction. The RNN and LSTM implemented in the experiments are built in analogous manners with the only difference being the type of computational cell used in the recurrent layer. RNN and LSTM tend to be more computationally expensive than other models due to their rather complex architectures which is why both RNN and LSTM are built using only one recurrent layer, following Lang and Rettenmeier (2017, p. 4) and Toth et al. (2017, p. 3). Each model’s recurrent layer consists of 256 RNN and LSTM cells, respectively since Sheil et al. (2018, p. 6) find 256 cells per recurrent layer to be optimal in their experiments, which are conceptually similar to the subsequent experiments. Deeper and more complex architectures could be explored in additional experiments. Moreover, both models use similar hyperparameters like the NN above, namely a Sigmoid activation in the output layer, the default Xavier uniform initialization, 20 percent dropout and recurrent dropout, respectively, and the Adam optimizer to minimize binary cross-entropy loss. In contrast to a ReLU activation being used in the input and hidden layers in the NN above, the default recurrent layer activation is a hyperbolic tangent (tanh) activation function.

There are two specialties regarding RNN and LSTM. First, RNN make the need for feature engineering largely obsolete by preserving past customer behavior in a “(…) latent state that corresponds to a representation of learned features (…)” (Lang and Rettenmeier, 2017, pp. 1-2). Therefore, features that are explicitly designed to capture past customer behavior are excluded from the training and test sets used for RNN and LSTM. This step allows to investigate their alleged superiority over vector-based models. Thus, features explicitly capturing purchases in the last hours or days previous to a given visit, visits in the last hours or days previous to a given visit and page views and product views in the last visit before a given visit are removed, reducing the number of features from 134 to 115 in the training and test sets for RNN and LSTM. Second, since RNN and LSTM operate on sequences of sessions, the training and test sets used for RNN and LSTM are transformed from being two-dimensional to being three-dimensional instead. Vector-based models use two-dimensional input data where an instance is a visitor’s session *s* that is described by *f* features. For sequence models, however, three-dimensional input data is required where an instance is represented by a visitor’s session *s* and all that visitor’s previous sessions *p*, all those sessions *s* and *p* each being described by *f* features.

1. Evaluation of Models and Experimental Results

Section 6 presents the experimental results and evaluates the models in terms of the six criteria that constitute the model evaluation framework. Section 6.1 evaluates the models with regard to the criterium of objectivity. Section 6.2 analyzes the experimental results considering the criterium of predictive accuracy and presents different metrics and methods to evaluate and compare model performance. To investigate the robustness of the experimental results, Section 6.3 presents cross-validation results for each model and different samples. The models used in the experiments vary widely in terms of structure and complexity. Therefore, Section 6.4 relates the notion of interpretability in machine learning to the models under examination and the role of their structure and complexity. Section 6.5 considers the models from the viewpoint of versatility. Complexity and model-specific idiosyncrasies determine algorithmic efficiency which is a relevant criterion for marketing executives. Thus, Section 6.6 presents the models’ efficiency considering the time they required for training and testing on samples of different sizes. The evaluation of the experimental results and comparison of the models constitutes a central part of this thesis and forms the foundation for the subsequent discussion and the derivation of managerial implications in Section 7.

* 1. Objectivity

Models should allow for the computation of the relative impact a feature has on the prediction of a given target (Lilien, 2011; Anderl et al., 2014), e.g. a conversion. LR and SVM satisfy the objectivity criterion because they enable the computation of coefficients that indicate a feature’s relative impact and whether the impact is positive or negative. The objectivity criterion is satisfied by DT, RF and BOOST as well since these models are able to return feature importances (i.e. the higher the importance of a feature, the more important the feature for the prediction of a conversion). Among the most important features identified by the models above are features that capture the time passed since the last purchase, cart-related events, product and page views and visitor-specific features like gender and age. Tables 10 to 14 in the Appendix show the top ten features most indicative of a conversion for LR, SVM, DT, RF and BOOST and their corresponding coefficients and feature importances, respectively.

For KNN and NN, however, the objectivity criterion is not fulfilled given that there is no straight forward way to compute the relative impact a feature has on a prediction. Same applies to the Gaussian NB used in the experiments above. Other implementations of NB, namely Multinomial and Bernoulli NB, however, have available ways to return coefficients that indicate a feature’s importance (Pedregosa et al., 2011).

Lang and Rettenmeier (2017, p. 6) state that their RNN with LSTM cells not only improve predictive accuracy and limit the need for extensive feature engineering, but these models are more explainable than vector-based models as well. They show that their RNN with LSTM cells are able to establish links between events in customers’ behavioral sequences and predictions of conversion probabilities that are saved in the recurrent units’ hidden states that are in turn updated every time an event happens (Lang and Rettenmeier, 2017, pp. 5-8). In their Figure 3, they visualize a customer’s fluctuating conversion probability over the course of several sessions, including the days passed since the previous session, the sum and type of events that happened in a session and the session duration (Lang and Rettenmeier, 2017, p. 8). Although, Lang and Rettenmeier (2017) show how events and conversion probabilities can be linked using RNN with LSTM cells, they only partly satisfy the objectivity criterion since it is not straight forward to create such visualizations and derive such explanations. Besides, one has to predict conversion probabilities rather than a binary conversion outcome which is not the case in this thesis’ experiments.

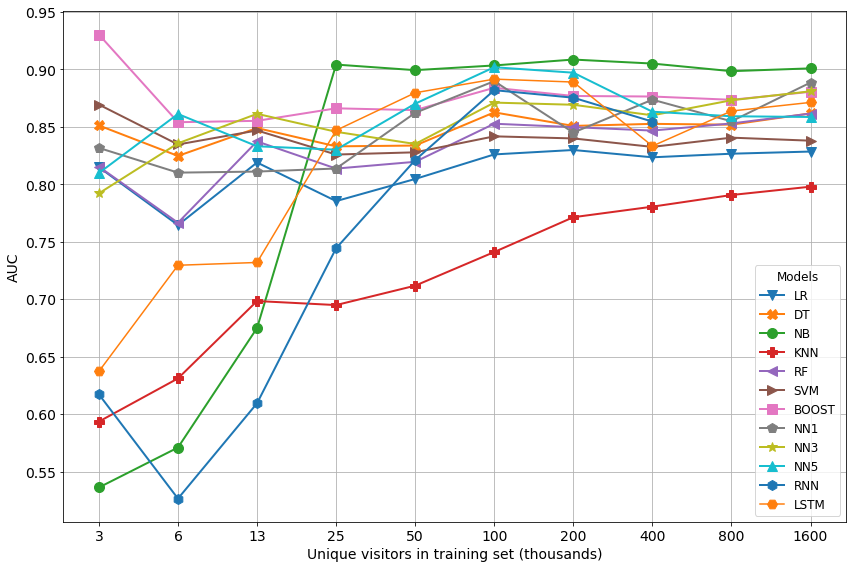
* 1. Predictive Accuracy

Models should be able to correctly predict conversions to ensure credibility in the models and their predictions, which is why the criterion of predictive accuracy is important (Lodish et al., 2001; Anderl et al., 2014). Learning curves are not only a way to investigate the relationship of sample size and predictive performance but also allow for the comparison of models, which is relevant because models tend to perform differently depending on the amount of training data they are fed (Shavlik et al., 1997; Perlich et al., 2003). Computing learning curves using cross-validation would probably deliver more robust results and standard deviations but is computationally expensive and time-consuming, especially for large sample sizes, which is why the present learning curve analysis refrains from it.

Accuracy is close to 100 percent for all models across all samples but is not a suitable measure of predictive accuracy due to the highly imbalanced classes in the data, resulting in a bias towards the majority class (i.e. sessions that do not lead conversions), and the inability of accuracy to distinguish between correctly classified predictions of different classes (Sokolova et al., 2006, p. 1016). Alternative performance metrics that are able to better account for class imbalance are AUC, precision, recall and F-score because they do not rely on a single performance indicator but are computed using several components indicative of predictive accuracy, explained in the following. AUC is defined as the area under the Receiver Operating Characteristic (ROC) curve, which is created by plotting the values of the true positive rate on the y-axis and the false positive rate on the x-axis, taken from the confusion matrix created from a classifier’s class predictions (Bradley, 1997, pp. 1145-1147). Precision answers the question of what proportion of predicted conversions was actually correct and is defined as the quotient of true positives, i.e. correctly predicted conversions, and the sum of true positives and false positives, i.e. all predicted conversions (Sokolova et al., 2006, p. 1016). Recall (also sensitivity) answers the question of what proportion of actual conversions was identified correctly and is defined as the quotient of true positives and the sum of true positives and false negatives, i.e. all actual conversions (Sokolova et al., 2006, p. 1016). The F-score (also F1-score or F-measure) computes the harmonic mean from precision and recall and is defined as two times the product of precision and recall divided by their sum (Sokolova et al., 2006, p. 1016).

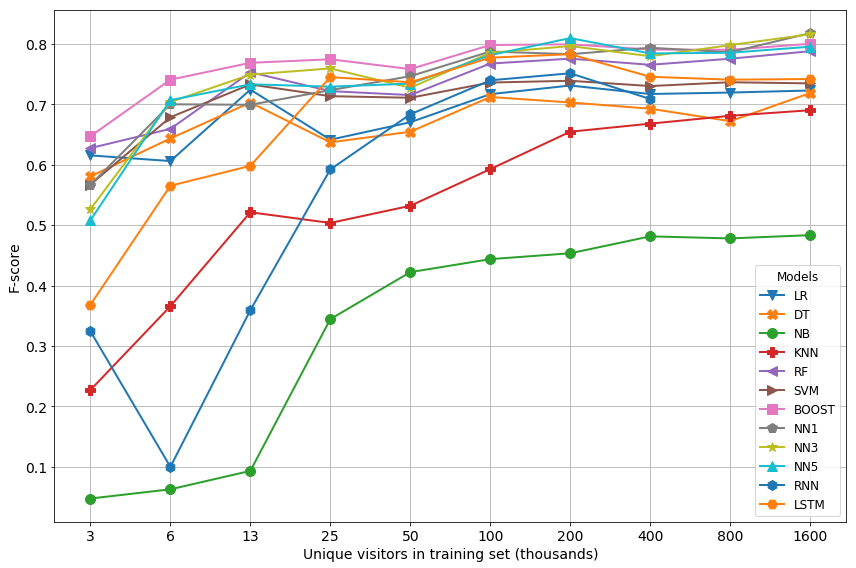
Figure 2 shows the relationship of AUC and the number of unique visitors in the training set for each model. Test set AUC is on the y-axis and the number of unique visitors in the training set is on the x-axis. The learning curves of LR, DT, RF, SVM, BOOST, NN1, NN3 and NN5 seem to be unstable for smaller training sets, indicated by the movements of the curves. Reaching a training set size of 100,000 unique visitors, the curves of these models stabilize and slightly flatten for the larger training sets. The difference between the models’ lowest and highest AUC score, respectively, is not very large as these models achieve relatively high AUC scores for small training sets already, indicating that they are overall fairly robust to variations of the number of training examples. NB first performs poorly for the three smallest training sets but reaches its level of peak performance from the 25,000 unique visitors training set on, outperforming all other models in terms of AUC. KNN almost continuously improves its AUC with more training data but is unable to reach a competitive level of performance. AUC of RNN and LSTM steadily improves, peaking at the 100,000 unique visitors training set. Then, AUC for both models does not improve further and the curves flatten until they finally reach a level similar to the one of other high-performing models with AUC scores between 0.83 and 0.9. LR, KNN and SVM are strictly dominated in terms of AUC by all other models from the 100,000 unique visitors training set on. NB reports the overall highest AUC from the 25,000 unique visitors training set on. The other models with similar performance for the largest training set are NN1, NN3 and BOOST followed by LSTM, DT and RF and finally NN5.

Figure 2



To derive more generalizable conclusions, Figure 3 additionally shows the relationship of F-score and the number of unique visitors in the training set for each model. Test set F-score is on the y-axis and the number of unique visitors in the training set is on the x-axis. Unlike in Figure 2, in Figure 3 no model reaches peak performance from the small training sets on, but all models require a certain amount of training data to yield satisfying F-scores. NB is the worst performing model, never exceeding an F-score of 0.5. This originates from the fact that NB predicts around ten times as many false positives across all samples compared to most other models, drastically deteriorating precision and consequently negatively impacting the corresponding F-scores. A reason for this might be that NB is too simplistic of a model to recognize complex patterns in the training data. Although KNN almost continuously increases performance in terms of F-score, it again does not tend to reach a competitive level of predictive performance. RNN reaches a fairly competitive F-score from the 50,000 unique visitors training set on while LSTM reaches a competitive F-score from the 25,000 unique visitors training set on already. LR, DT and SVM find themselves at the lower end of the overall performance spectrum. The highest F-scores tend to be reached by NN1 and NN3, followed by BOOST, NN5 and RF – although differences between these models are quite small since their F-scores are within a small range of values.

Figure 3



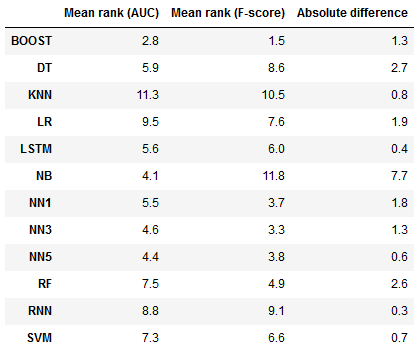
Both Figures 2 and 3 do not report AUC and F-scores for RNN for training sets with 800,000 and 1.600,000 unique visitors, respectively. This is because validation loss during training turned indefinite for these training sets, which is an indication of vanishing gradients. This is a typical issue of RNN that is addressed by Hochreiter and Schmidhuber (1997) who develop LSTM cells that mitigate this and other issues of RNN. As a result, LSTM does not suffer from a similar problem so that AUC and F-score can be computed for LSTM for all samples.

Tables 15 to 24 in the Appendix report the models’ detailed predictive performance in terms of accuracy, AUC, true negatives, false negatives, true positives and false positives, precision, recall and F-score for ten samples of different sizes, respectively.

Statistical tests can be used to investigate whether there are significant differences between multiple models’ predictive accuracy on multiple data sets (Demsar, 2006). One such test is the non-parametric Friedman test that ranks models’ performance for each data set separately under the null hypothesis that that there are no significant differences between the models, i.e. their ranks should be equal (Friedman, 1937; Friedman, 1940; Demsar, 2006, p. 11).

Table 5 shows the average ranks of all models computed using their AUC and F-scores for all samples, respectively, and the absolute differences between each model’s average ranks for AUC and F-score. BOOST has the best average rank for AUC and F-score and a low absolute difference between its ranks. NN3, NN5 and LSTM have good average ranks for both AUC and F-score and fairly small absolute differences between their respective average ranks. The average ranks of DT, LR, NB, NN1 and RF vary depending on the choice of performance metrics, indicated by the large absolute difference between the models’ respective average ranks. KNN, RNN and SVM report relatively bad average ranks considering AUC and F-score, respectively. This is an indication that the evaluation and comparison of models tends to depend, among other factors, on the evaluation metric of choice.

Table 5



The Friedman test can be used in this case because multiple models are to be compared across multiple samples and because only the average performance is of interest and not its variation, which would be problematic since the samples are not independent (Demsar, 2006, p. 5). According to the Friedman test, considering twelve models (11 degrees of freedom) and ten samples (9 degrees of freedom), the models’ respective ranks are significantly different from the average ranks of 6.4417 (AUC, F-value of 6.415) and 6.45 (F-score, F-value of 22.2834) given the critical F-value of 1.8867. Therefore, the null hypothesis that the models are equal can be rejected at the 0.05 significance level. Since the null hypothesis of the Friedman test can be rejected, the Nemenyi test can be used for a pairwise comparison of models’ performance (Nemenyi, 1963; Demsar, 2006, pp. 11-12). A model outperforms another model at the 0.05 significance level if the difference between their average ranks is at least equal to the critical distance of 5.2695 (based on the number of models and samples under consideration and the resulting critical value of 3.268 of the Nemenyi test). The formulas to calculate the Friedman and Neymeni tests and how to retrieve the tests’ respective critical values can be found in Section 3.2.2 of Demsar (2006).

For example, BOOST is significantly better than KNN, LR and RNN in terms of the average rank based on AUC and significantly better than DT, KNN, LR, NB and RNN in terms the average rank based on F-score. LSTM is only significantly better than KNN in terms of the average rank based on AUC and only significantly better than NB in terms of the average rank based on F-score. KNN, LR, RF, RNN and SVM are not significantly better than any model in terms of the average rank based on AUC. Likewise, DT, KNN, LR, NB, RNN and SVM are not significantly better than any model in terms of the average rank based on F-score. These observations confirm the hypotheses from the meta-analysis in Section 3 that there is no single model that outperforms all others, but there are certain tendencies, such as ensembles (e.g. BOOST) outperforming individual models (e.g. DT).

* 1. Robustness

Models should be robust in the sense that they yield stable and reproducible results over multiple runs, i.e. the variance of performance over several model runs should be low, which is covered by the criterion of robustness (Little, 1970; Little 2004; Anderl et al., 2014). The models’ robustness is tested using ten-fold cross-validation and four samples of small to medium size to make the generally computationally expensive and time-consuming cross-validation procedure more efficient. Using ten folds allows to investigate models’ in-sample robustness while using four different samples enables insights into cross-sample robustness and the effect of sample size. The folds have been created in a way that guarantees they contain randomly selected and equal amounts of distinct unique visitors. This procedure has been applied to balance classes and data characteristics and to avoid cutting customer journeys (i.e. to ensure that a customer’s sessions are not split across training and test sets). The four samples contain roughly 8,000, 31,000, 125,000 and 500,000 unique visitors, respectively.

In general, variation in precision, recall and consequently F-score tends to be higher than variation in accuracy and AUC. Standard deviations in accuracy and AUC typically range between less than 0.01 and up to 0.07 for the two small samples and between less than one and up to two for the two medium-sized samples. Standard deviations in precision, recall and F-sore typically vary between less than 0.01 up to 0.1 for the two small samples and between less than 0.01 and typically up to 0.06.

There are a several outliers (narrowly defined here as case where standard deviation exceeds 0.1), though, typically in the two smaller samples and across several models. But overall, models’ robustness tends to improve with increasing sample size, indicated by typically lower standard deviations and thus less numerous outliers. The sole exception constitutes RNN with outliers for different metrics in all samples but the one containing 125,000 unique visitors. LR, DT, NB, BOOST and NN1 are the only models that do not report any standard deviations exceeding 0.1 across all metrics and samples. The reason why for NN1 no outliers but for NN3 and NN5 several outliers are reported may be due to the more complex architecture of the latter two models, increasing their tendency to overfit individual folds.

Tables 25 to 28 in the Appendix report averaged accuracy, AUC, precision, recall and F-scores with standard deviations in parenthesis for the four aforementioned samples computed in the respective cross-validation procedures.

* 1. Interpretability

Models and their results should be simple and easy to communicate to foster acceptance and application by marketing executives, which is captured by the criterion of interpretability (Little, 1970; Little, 2004; Anderl et al., 2014). For the sake of completeness and to shed additional light onto the concept and relevance of interpretability in machine learning, recent developments regarding this topic are briefly outlined in the following. Guidotti et al. (2018) provide a much more comprehensive overview and summary of the research efforts in this field for the interested reader. In recent years, several studies have been published, stressing and debating the importance and the notions of interpretability and explicability of machine learning models. Despite the substantial amount of research that has been recently conducted in this field and the general consensus that these are important concepts to foster adoption of machine learning applications, a unified and holistic view of what these concepts actually imply and how they can be satisfied and evaluated appears to be hard to agree upon (Doshi-Velez and Kim, 2017; Lipton, 2017). There are single studies that compare different machine learning models in terms of complexity and monotonicity, including models such as decision trees, nearest neighbors and Bayesian networks (Freitas, 2014). But there is also a range of specialized methods that have been developed to explicitly allow for more transparency of machine learning models and their outputs. For example, Ribeiro et al. (2016) state the importance of trust in machine learning models to facilitate their widespread adoption and therefore present LIME (Local Interpretable Model-agnostic Explanations) which is an explanation technique able to explain the predictions of any classifier by learning an interpretable model around a prediction in a local environment. Lundberg and Lee (2017) propose a unified framework for interpreting predictions called SHAP (SHapley Additive exPlanations), addressing the tensions between accuracy and interpretability that are caused by deploying increasingly complex models in practice. Sundararajan et al. (2017) identify the fundamental axioms of sensitivity and implementation variance which they use to design an attribution method called Integrated gradients that is supposed to enable the attribution of a deep neural network’s predictions to its input features and thereby make it more understandable for its users. Considering the current state of research in the field of interpretability of machine learning models with all its available methods to make these models more transparent would exceed the scope of this thesis. Explicitly investigating the aspects of interpretability and explicability in the comparison of machine learning and deep learning models for a specific use case in addition to testing some of the suggested methods above in practice could be an interesting task for future research.

In addition to model complexity, Freitas (2014) suggests monotonicity constraints as a possible means to evaluate a model’s degree of interpretability, but the following evaluation stays close to Anderl et al.’s (2014) marketing-centric reading of model interpretability. Therefore, the following evaluation is mainly focused on model complexity and whether a model is able to explain its predictions by assigning meaning and weights to the features used for predicting conversions. LR is a linear model and the sign and magnitude of coefficients indicate which features the model considered important for its prediction. DT possess a graphical structure, typically contains only a subset of all available features reducing complexity (depending on the tree size), the hierarchical structure of trees and the possibility to compute the relative importance of features provide insight into which attributes of the data are the most relevant for the model’s prediction (Freitas, 2014, p. 2). Gaussian NB used in the experiments above is a comparably simple yet not linear model since it relies on products of probabilities and does not provide information on the relevance of features (as mentioned above, other implementations of NB are able to provide such information), making it generally more difficult to comprehend its predictions. For KNN, according to Freitas (2014, p. 4), the feature values of the nearest training instances are usually different for every new test instance to be classified and in data sets with many features even neighboring instances can differ substantially, making KNN less explainable overall. Using prototypes (i.e. instances that represent typical data points) instead of the entire training data and computing attribute weights that are proportional to their predictive power are two approaches to improve the explicability of KNN’s predictions, but they come with additional effort (Freitas, 2014, p. 4). Ensembles like RF and BOOST consist of multiple decision trees that are relatively easy to understand individually, but the ways in which these ensembles combine individual trees and their predictions make them less intuitive. The visualization of individual trees from within these ensembles is possible with the implementations of RF and BOOST used in the experiments above but allows only limited insight because these ensembles consist not of a single but of many trees. The calculation of the importance of individual features for the ensembles’ predictions additionally helps making RF and BOOST more comprehensible. SVM is a linear model because it produces a two-dimensional hyperplane in binary classification problems and the computation of coefficients and their signs provides information regarding which features the model considered important for its prediction. Although NN1, NN3, NN5, RNN and LSTM are generally shallow rather than deep in terms of the number of hidden layers, they are the most complex models under consideration given the multitude of computational units in the input and hidden layers and the architectural choices made when building these models. Besides, there is no straight forward way to compute the impact individual features have on a neural network’s predictions. There is the, however not straight forward, possibility to apply specialized methods to try to establish a relationship between a neural network’s predictions and its input features (e.g. Sundararajan et al., 2017).

The degree to which machine learning applications need to be transparent and interpretable certainly varies from application to application. For example, the requirements in terms of a user’s in-depth understanding of model structure, how predictions are derived and which features are indicative of a prediction are probably different for an email spam classifier and a machine learning application in healthcare. Ultimately, a marketing executive needs to specify for herself and her specific application the importance of comprehending increasingly complex and diverse models, their mechanics and their output for her individual decision-making.

* 1. Versatility

The criterion of versatility requires models to be easy to control and adaptive when conditions change over time, particularly in fast-paced environments like e-commerce, e.g. when new data or features become available (Little, 1970; Little, 2004; Anderl et al., 2014). All models considered in the experiments above are versatile in the sense that they are easily adapted if new data or features become available. After processing new data and features in the same or a similar manner as explained in Section 5.2, models can be simply retrained using training and test sets extended by fresh data and features. The time models require for being retraining on new data, however, varies substantially, as explained in Section 6.6. If, in addition, the data’s level of aggregation or granularity is to be changed or a different target is to be used, models’ flexibility depends on the degree to which these changes impact the underlying structure of the data and prediction problem. For example, if instead of modeling purchase prediction as a binary classification problem, conversion probabilities were to be predicted (which would be a natural extension of the experiments above), some models and their predictions would need to be substantially adapted if they can only poorly or not at all predict probabilities in a straight forward fashion, e.g. NB and SVM (Caruana and Niculescu-Mizil, 2006, p. 163).

* 1. Algorithmic Efficiency

Models should allow to be updated in a reasonable amount of time and should be able to compute results fairly quickly to provide them to marketing executives when they need them, which is addressed by the criterion of algorithmic efficiency (Lodish, 2001; Anderl et al., 2014). In addition, scalability can be explicitly used to compare models (e.g. Lim et al., 2000).

Figure 4 shows training times in seconds for all models and samples. Some models required less than one second for training for some training sets which is why their curves are close to zero and flat particularly for small sample sizes. NB appears to be the fastest model with only a couple of seconds for training and testing for even the largest samples in the experiments. LR, DT and RF require a couple of seconds up to several minutes for training for even large amounts of data. More complex models like SVM, BOOST, NN1, NN3 and NN5 tend to be still comparably fast with training times of several minutes up to about 15 to 20 minutes for large samples. More complex models like RNN and LSTM tend to be slower in general and require several minutes for training on smaller up to a couple of hours for training on larger samples. It seems intuitive that models generally require more time for training and testing with increasing sample size.

Figure 5 shows the times the models required for testing for different sizes of the test set. Most models require substantially less time for testing than for training which seems legit given the train test split ratio of four to one. Interestingly, however, KNN is the only model that requires more time for testing than for training, the increase in testing time being even disproportionally. In comparison to the other models, testing takes so much longer for KNN that its corresponding curve is plotted in a separate graph due to the substantial differences in scale. This phenomenon could be ascribed to the characteristics of KNN’s specific algorithm structure.

Overall, all models except for KNN achieve reasonable run times while less complex models tend to be generally faster, even substantially on some occasions. What is reasonable, however, as well as the trade-off between predictive accuracy, robustness and algorithmic efficiency heavily depends on use case and business environment at hand (Anderl et al., 2014, p. 22).

Figure 4

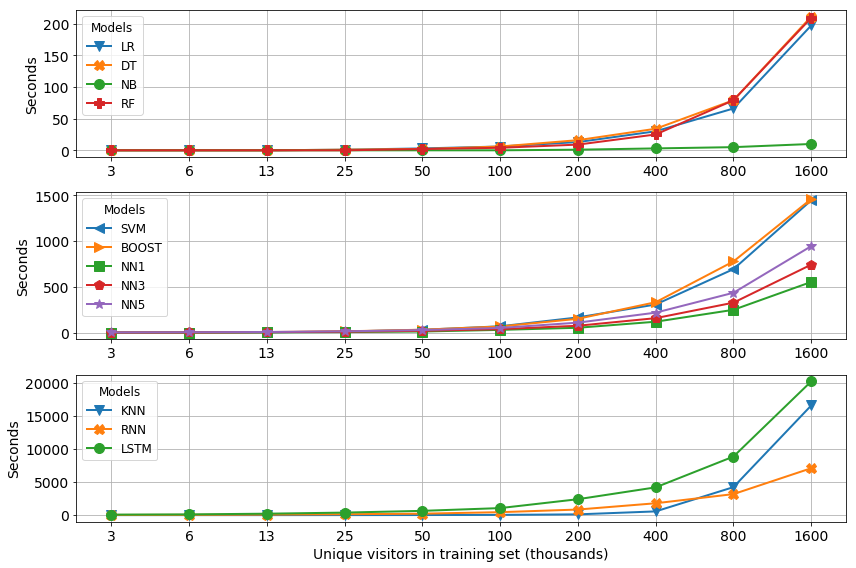
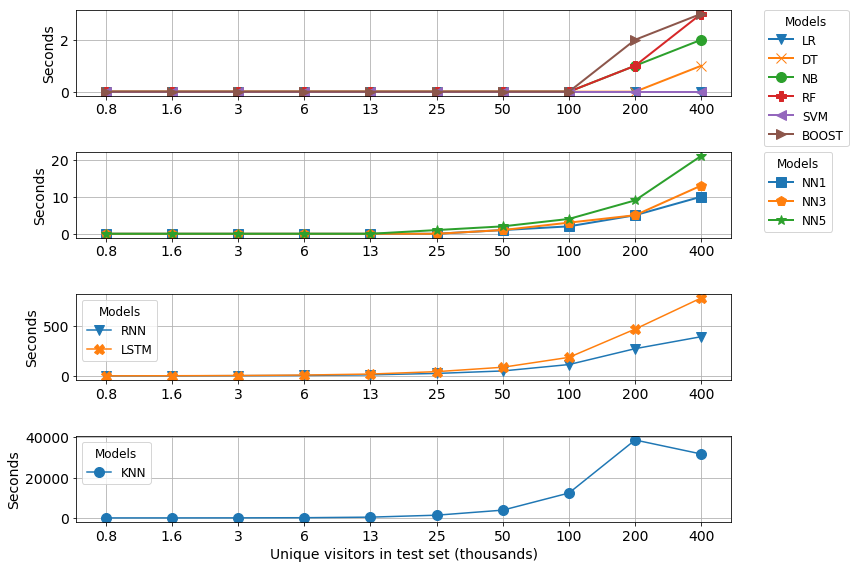


Figure 5



1. Discussion and Managerial Implications

***Complexity of neural networks***. The variation in the number of hidden layers in the neural networks is supposed to uncover the effect of a more complex model structure predominantly on predictive accuracy and algorithmic efficiency since all other criteria of the applied model evaluation framework are largely unaffected in this case. For example, less hidden layers make a neural network’s structure certainly more transparent but its predictions not more explainable. Section 6.2 shows that the predictive performance of the neural networks with one, three and five hidden layers, respectively, is similar and that they are not statistically significantly different from each other. Section 6.6 reveals that the training and test times of the considered neural networks are not substantially different as well. Admittedly, the neural networks considered in this thesis are quite shallow and thus, the comparison of deeper architectures could yield further and more profound insights.

***RNN versus LSTM***. RNN and LSTM are similar models in the sense that they just differ in the type of their recurrent layers, i.e. LSTM is an RNN with more powerful LSTM cells in its recurrent layer instead of regular RNN cells (Lang and Rettenmeier, 2017, pp. 2-3). As a consequence, their evaluation turns out to be similar for most criteria of the applied model evaluation framework. However, in terms of predictive accuracy measured by AUC and F-score, LSTM tends to outperform RNN for smaller training sets while their predictive performances converge with increasing sample size. Besides, RNN appears to suffer from vanishing gradients for the two largest training sets and consequently fails to deliver results while LSTM, due to the fact that is was explicitly designed to cope with issues like vanishing gradients, manages to product competitive results for these two samples (Section 6.2). Cross-validation in Section 6.4 additionally shows that RNN tends to be less robust, producing high standard deviations several times. Although both models achieve similar run times for most samples, LSTM tends to require substantially more time for training and testing for the largest sample under consideration (Section 6.6).

***Vector- versus sequence-based models***. Lang and Rettenmeier (2017, p. 1) claim that sequence-based models like RNN and LSTM constitute a better fit to sequential clickstream data used in customer journey prediction compared to vector-based models. This is since these models are able to achieve better predictive accuracy, require less manual feature engineering and allow for explainable visualizations of their predictions (Lang and Rettenmeier, 2017, p. 6). Predicting conversions and not conversion probabilities, however, this thesis’ experiments cannot confirm the proposed superiority of sequence- over vector-based models. Although RNN and LSTM achieve competitive predictive performance and LSTM is even significantly better than some models, they both overall do not tend to yield significantly better predictive performance compared to the considered vector-based models (Section 6.2). Besides, RNN tends to be less robust (Section 6.4) and both RNN and LSTM require substantially more time for training and test (Section 6.6) compared to most other vector-based models – without compensating for longer run times with superior predictive accuracy. Besides, recurrent neural networks tend to be more complex and less intuitive than other, linear and less complex models (Section 6.3) and might even require additional data transformations to function properly (Section 5.4).

***Model evaluation and comparison***. Table 6 summarizes the evaluation and comparison of the models in terms of the six criteria of the model evaluation framework. A check mark indicates that a model satisfies a criterion, a checkmark in parenthesis indicates that a model at least partly satisfies a criterion and a cross mark implies that a model does not sufficiently satisfy a criterion. This summary is not meant to issue an ultimate judgement on the models under consideration but rather intends to relate them regarding this thesis’ particular use case and to illustrate specific tendencies of the evaluation and comparison.

The comprehensive model selection and extensive experiments assessed by a multi-faceted evaluation framework, specifically tailored to the requirements of marketing executives, allow for the evaluation and comparison of the considered models in a comprehensive manner and ultimately the recommendation of a preferred model.

Section 6.1 finds LR, SVM, DT, RF and BOOST to satisfy the objectivity criterion since these models allow for the computation of a feature’s relative impact on a model’s prediction while NB, KNN and the (recurrent) neural networks do not provide such insights without additional effort or modifications.

Although many models achieve satisfying predictive accuracy, only a few tend to dominate others, such as BOOST, NN3, NN5 and LSTM, while other models’ performance is either unbalanced across different performance metrics (e.g. DT, LR, NB, NN1 and RF) or their performance is simply not competitive enough (e.g. KNN, RNN and SVM) as shown in Section 6.2.

Most models, except for occasional outliers for some models on small samples and the overall not very robust RNN, are fairly robust in comparison to each other and across samples of different sizes (Section 6.3).

As pointed out in Section 6.4, more complex models and their predictions tend to be more difficult to comprehend and the tendential lack of general superior predictive performance of more complex models (Section 6.2), leads to the conclusion that more complex models are not necessarily superior. Besides, insight into which features are particularly decisive for a model’s predictions helps make them more explainable (Section 6.1).

Overall, the considered models are versatile to a certain degree, depending on the exact nature of a given modification as some models tend to cope better with certain modifications than others. If instead of conversions one were to predict conversion probabilities for example, some models could probably be more easily adapted than others (Section 6.5).

Most models achieve reasonable run times even for large amounts of data while the hypothesis that more complex models require more run time generally seems to hold true. A particular exception constitutes KNN that appears to not scale well to increasing amounts of data and requires more time for testing than for training (Section 6.6).

Considering all of the above, Gradient Tree Boosting (BOOST) seems to present for this thesis’ use case a promising compromise between predictive performance, interpretability and model complexity as it satisfies many of the above requirements either completely or at least to a reasonable extent. Although BOOST is a quite complex model and being an ensemble not very intuitive per se, the possibility to compute feature importances and individual decision trees from within the ensemble ensures at least some degree of explicability and transparency with regard to its inner workings and predictions. It is not only robust and theoretically able to predict (conversion) probabilities (Pedregosa et al., 2011) but outperforms several other models in terms of predictive accuracy measured by AUC and F-score. BOOST is not the fastest model, but its run time is still moderate. Unfortunately, optimizing BOOST tends to be complicated as there are many hyperparameters to tune. Therefore, other models potentially worth considering with less hyperparameters to tune could be LR, DT and RF as they are relatively simple, robust, quick and allow to assign relevance to features – despite their shortcomings regarding predictive accuracy compared to BOOST as shown in Section 6.2.

***Managerial implications***. The experiments show that valuable insights from clickstream data can be extracted using machine learning and deep learning. For example, it is noteworthy that many models’ predictive performance saturates for relatively small amounts of data already, indicating that not necessarily the entirety of available data needs to be used to produce useful results. Besides, some models allow for the computation of a feature’s relative impact on predicting conversions. These insights can be used by marketing executives to improve their decision-making considering different business aspects, such as customer retention management or conversion rate optimization. Potential measures could be to use these insights for more customer-centric personalization or couponing, tailored to the respective customer’s needs learned from the data and models. But the experiments also show that using such models can be difficult given the plethora of models and performance metrics to choose from and associated caveats to consider. A thorough model evaluation using different criteria and also several performance metrics is therefore crucial to avoid false conclusions. A marketing executive needs to decide which compromises and trade-offs she is willing to accept, e.g. considering predictive accuracy, interpretability of models and output and computational resources and time – Should a model be able to recognize as many (potential) buyers as possible, i.e. high recall but including many false positives and thus low precision, or should a model rather be very confident in its predictions risking overlooking some (potential) buyers, i.e. high precision but low recall? The application of machine learning and deep learning models for customer journey prediction can help marketing executives shed light onto these and other questions, improving their decision-making and in turn making their own decisions more explainable and justifiable. There are several promising alternatives for marketing executives to choose from, depending on the specific requirements of the problem, data and business environment at hand.

1. Conclusion

The objective of this thesis is to investigate a selection of machine learning and deep learning models to predict conversions using e-commerce clickstream data and to evaluate them considering criteria that are relevant to marketing practitioners who are supposed to use them as decision support in their work. Data is a natural component of e-commerce and in combination with increasingly powerful and widespread machine learning and deep learning applications offers abundant potential to improve customer satisfaction and different business aspects, making this topic highly relevant (e.g. Wedel and Kannan, 2016; Sheil et al., 2018).

Twelve models derived from a meta-analysis of comparative machine learning studies are compared on ten clickstream data samples of different size, created from an e-commerce website’s clickstream data. Logistic regression (LR), a decision tree classifier (DT), naïve Bayes (NB), k-nearest neighbors (KNN), random forest (RF), a support vector machine (SVM), a gradient tree boosting classifier (BOOST), neural networks with one (NN1), three (NN3) and five (NN5) hidden layers, respectively, a recurrent neural network (RNN) and a long-short term memory network (LSTM) are evaluated according to a model evaluation framework consisting of six criteria that management and marketing literature find a model desirable to satisfy: *objectivity*, *predictive accuracy*, *interpretability*, *robustness*, *versatility* and *algorithmic* e*fficien*cy (Anderl et al., 2014, p. 7-10).

The experiments do not uncover a substantial effect of the number of hidden layers in the considered neural networks on predictive accuracy and algorithmic efficiency, but this is probably since all three are shallow and therefore quite similar in terms of complexity. LSTM are found to yield better predictive performance and are more robust compared to RNN, both requiring a certain amount of training data to achieve competitive predictive performance and longer run times compared to less complex models. Although being sequential models and therefore a natural fit to predict conversions from sequences of customer sessions (Lang and Rettenmeier, 2017, p. 1), they do not tend to significantly outperform the other vector-based models but rather achieve similar predictive performance – tendentially LSTM more so than RNN however.

Overall, BOOST, LR, DT and RF constitute the models that appear to best satisfy the evaluation framework’s criteria since they are fairly accurate, robust, quick and allow for the computation of features’ relative impacts on predictions, the latter particularly helping to better understand the more complex model structures of RF and BOOST. It is worth noting that BOOST ranks higher on average than the other three mentioned models in terms of predictive performance, even significantly outperforming LR and DT depending on the evaluation metric. No model, however, significantly outperforms all others, confirming similar findings and hypotheses derived from the meta-analysis.

The contribution of this thesis has a theoretical and a practical aspect. First, it contributes to research by conducting a multi-dimensional meta-analysis of comparative machine learning studies from the past three decades, identifying general tendencies and particularly relevant works. Second, it consolidates and extends existing work in the sphere of customer journey prediction using machine learning and deep learning. It does this by increasing the scope of models and metrics used based on the meta-analysis and, most importantly, evaluates the models under consideration with regard to what management and marketing science have found to be the most important criteria to the addressees of these models, namely marketing executives. The experiments provide marketing executives with a comprehensive evaluation and comparison of a variety of models that can be used as a starting point to then make necessary adjustments and choices individual use cases and business environments require.

There are several limitations to this thesis that offer potential for future research. First, the selected models are applied and evaluated using only one data set and use case which might restrict generalizability. Extending the scope of this thesis with additional data sets from other e-commerce domains and use cases from the field of marketing (e.g. churn prediction and customer segmentation) could improve the general validity of the results. Another approach could be to apply adapted models to individual customer segments that are generated algorithmically to better map segment-specific idiosyncrasies and customer segments (e.g. high- and low-activity customers).

Second, the models are implemented and compared mostly using their default hyperparameter settings. Although this considerably facilitates the conduction of the experiments, it might bias the comparison and evaluation in the sense that some models’ defaults are inherently superior. Fine-tuning the hyperparameters of all considered models could enhance the significance of their comparison.

Finally, the scope of customer journey prediction is broad, but only one aspect is explored in this thesis. A natural extension to predicting conversions in a binary setting would be to predict conversion probabilities instead since they might represent a customer’s propensity to purchase more accurately (e.g. Moe and Fader, 2004; Boroujerdi et al., 2014). In addition, the product(s) likely to be purchased in a given session could be predicted as well, moving the problem closer to the domain of recommender systems (e.g. Sarwar et al., 2015; Wu et al., 2015).

References

Appendix