Evaluation of Machine Learning and Deep Learning Models for Customer Journey Prediction in E-Commerce

An Executive Perspective

Contents

1. Introduction

- Introduction, motivation, (business) relevance
- Business problem: How can businesses predict customer behavior using clickstream data to timely react to customer behavior using personalization and couponing for example?
    - Which users are most likely to purchase (predict purchasing intent) (Sheil et al. 2018)
    - Which elements of the product catalogue do users prefer (rank content) (Sheil et al. 2018)
- Research question: How do different machine learning and deep learning models perform on the problem of customer journey prediction in e-commerce in comparison and particularly with regard to criteria that are relevant to marketing managers and executives?
- Hype about deep learning: Popularity of deep learning in computer vision, speech recognition and natural language processing – but does it really always have to be deep learning or do other models also suffice? → recent deep learning breakthroughs (e.g. ImageNet, BERT etc.)
- Algorithmic trust and trust in models in general, Grahl's literature suggestions: Dietvorst et al. (2016), Brynjolfsson and Mitchell (2017), Hall et al. (2017), Logg et al. (2018)
- Contributions to research and industry/practice
    - Research: profound meta-analysis of comparative machine learning literature
    - Industry/practice: Evaluation of traditional and novel models for customer journey prediction in e-commerce from the perspective of marketing managers and executives (e.g. beyond just predictive accuracy) based on a formal theory-backed evaluation framework

Section 2 introduces the methodology chosen 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 introduces the framework selected to evaluate the models used to predict customers' purchasing behavior; Section 5 explains the experimental setup, the data and the models in detail; Section 6 evaluates the experiments along the dimensions 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.

2. Methodology

The objective of the present thesis is to compare a literature-based selection of machine learning and deep learning models for the application of customer journey prediction in e-commerce, applying a framework that allows the evaluation of the chosen models with regard to criteria that marketing executives consider relevant for their decision making. To achieve this objective, the methodology for conducting a sound comparative study consists of a theoretical approach that is divided into two parts.

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, only the most relevant comparative studies have been considered based on a catalogue of specific selection criteria. 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- and marketing-centric part of the meta-analysis. Both parts of the meta-analysis examine different dimensions of the selected studies, such as the choice of models, the size 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 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 subsequent experiments. This approach allows to answer the question whether there are models or model families that tend to dominate others.

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; Lilien and Rangaswamy, 2004; 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) evaluation 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. 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.

3. Related Work

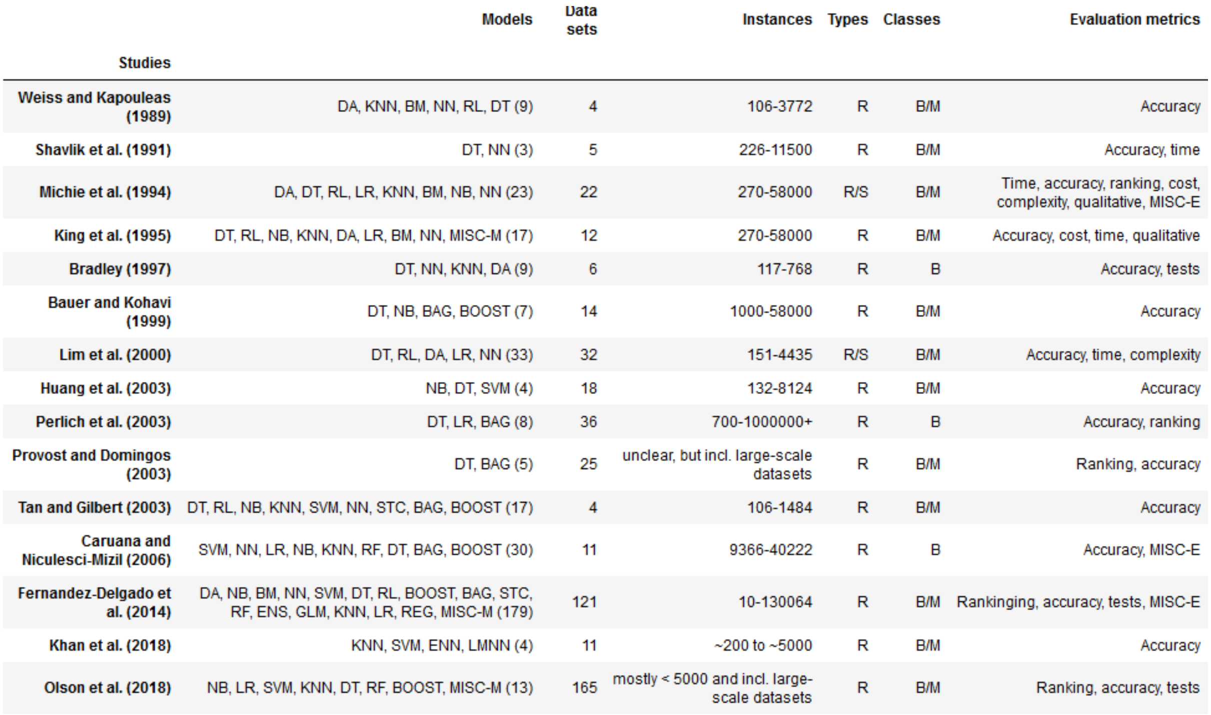
Section 3.1. presents a selection of related studies that compare different machine learning and deep learning models in general, mainly comparing their predictive performance and other metrics on classification tasks for a variety of different use cases and data sets. Section 3.2. explicitly focusses on studies that compare machine learning and deep learning models for the application of 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 briefly 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 compared in the subsequent 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.

3.1. 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 has been developed 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 model 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 can be condensed to the studies that are not only the most relevant to this thesis but also among the most notable overall. There are other studies that present and analyze previous comparative studies and leverage their learnings (e.g. Michie et al., 1994; Kind et al., 1995; Van den Poel and Buckinx, 2005; Baumann et al., 2018).

Table 1 presents 15 comparative studies that met the criteria above. The author(s) and the year of the study's publication are in the first column, the models and 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 have been mildly generalized and grouped. If a study developed a custom model, it has been tagged as such (CM). If generally unpopular or rather special models have been used that did not appear in many other studies, they have been combined into a miscellaneous group (MISC-M). Abbreviations have been used for all other models and families of models. A great variety of different metrics has been used in the considered studies for performance and model evaluation, which is why they have been 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 have been 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 used qualitative metrics to describe a model's ease of use for example and have been therefore tagged by ***qualitative***. Other studies used different ranking methods that have been combined in ***ranking***. Certain statistical tests have been also used to evaluate different models and have been grouped under ***test***. Finally, the time a model required for training, testing or classifying an unseen observation is captured within ***time***. The remaining metrics that could not be grouped within one of the categories above have been 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 1 and 2 in the Appendix, respectively.

Table 1



Studies	Models	Data sets	Instances	Types	Classes	Evaluation metrics
Weiss and Kapouleas (1989)	DA, KNN, BM, NN, RL, DT (9)	4	106-3772	R	B/M	Accuracy
Shavlik et al. (1991)	DT, NN (3)	5	226-11500	R	B/M	Accuracy, time
Michie et al. (1994)	DA, DT, RL, LR, KNN, BM, NB, NN (23)	22	270-58000	R/S	B/M	Time, accuracy, ranking, cost, complexity, qualitative, MISC-E
King et al. (1995)	DT, RL, NB, KNN, DA, LR, BM, NN, MISC-M (17)	12	270-58000	R	B/M	Accuracy, cost, time, qualitative
Bradley (1997)	DT, NN, KNN, DA (9)	6	117-768	R	B	Accuracy, tests
Bauer and Kohavi (1999)	DT, NB, BAG, BOOST (7)	14	1000-58000	R	B/M	Accuracy
Lim et al. (2000)	DT, RL, DA, LR, NN (33)	32	151-4435	R/S	B/M	Accuracy, time, complexity
Huang et al. (2003)	NB, DT, SVM (4)	18	132-8124	R	B/M	Accuracy
Perlich et al. (2003)	DT, LR, BAG (8)	36	700-1000000+	R	B	Accuracy, ranking
Provost and Domingos (2003)	DT, BAG (5)	25	unclear, but incl. large-scale datasets	R	B/M	Ranking, accuracy
Tan and Gilbert (2003)	DT, RL, NB, KNN, SVM, NN, STC, BAG, BOOST (17)	4	106-1484	R	B/M	Accuracy
Caruana and Niculesci-Mizil (2006)	SVM, NN, LR, NB, KNN, RF, DT, BAG, BOOST (30)	11	9366-40222	R	B	Accuracy, MISC-E
Fernandez-Delgado et al. (2014)	DA, NB, BM, NN, SVM, DT, RL, BOOST, BAG, STC, RF, ENS, GLM, KNN, LR, REG, MISC-M (179)	121	10-130064	R	B/M	Rankinging, accuracy, tests, MISC-E
Khan et al. (2018)	KNN, SVM, ENN, LMNN (4)	11	~200 to ~5000	R	B/M	Accuracy
Olson et al. (2018)	NB, LR, SVM, KNN, DT, RF, BOOST, MISC-M (13)	165	mostly < 5000 and incl. large-scale datasets	R	B/M	Ranking, accuracy, tests

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 used 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 focused 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) each with nine, 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 used dozens of data sets (Lim et al., 2000; Perlich et al., 2003; Provost and Domingos, 2003), some used even more than one hundred data sets (Fernandez-Delgado et al., 2014; Olson et al., 2018), but there are also studies that used 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 justified their selection and explained the characteristics of the data sets used, but some studies made it difficult to understand what the data they used was exactly like (Provost and Domingos, 2003; Khan et al., 2018). Bauer and Kohavi (1999) and Perlich et al. (2003) explicitly demanded data sets to contain at least 1000 and 700 observations, respectively. Only two of the considered studies used 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 have been used in multiple studies, typically taken from the University of California, Irvine Machine Learning Repository. Just three studies exclusively considered binary classification problems (Bradley, 1997; Perlich et al., 2003; Caruana and Niculescu-Mizil, 2006) while the remaining twelve studies additionally considered multi-class problems. All 15 studies used at least one evaluation metric related to accuracy, such as error or AUC. Nine out of 15 studies used metrics from multiple groups. Ranking metrics have been used by five, time has been used by four and tests have been used by three studies (Table 1, column seven). Only two studies evaluated models using metrics related to complexity (Michie et al., 1994; Lim et al., 2000). Likewise, only two studies used qualitative criteria for model evaluation and comparison, e.g. Michie et al. (1994) crafted a user guide for model evaluation and selection and King et al. (1995) evaluated 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 compared and evaluated 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 have been 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). Most studies explicitly confirmed 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) stated that there are no statistically significant differences between many models they evaluated, they claimed that there are huge differences in training time and interpretability though. Some studies that compared only a few models have been able to derive more differentiated conclusions. Bauer and Kohavi (1999) found that bagging (BAG) generally outperformed boosting (BOOST) while both performed better compared to DT and NB – however, at the cost of interpretability since BAG and BOOST are more complex and less interpretable. Perlich et al. (2003) stated that LR tended to perform better for smaller data sets while DT tended 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 subsequent experiments has been based on the number of total occurrences of a model across all studies. This 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 could not prove themselves. As a consequence, the frequently used models are likely to be the ones who could 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 presented in Table 1 in detail. Nevertheless, it is worth mentioning that most studies used a form of resampling technique, such as holdout or cross-validation, because some data sets that have been used had 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), have also been occasionally applied. A selection of these techniques is inspired by the meta-analysis and applied in the subsequent experiments as well.

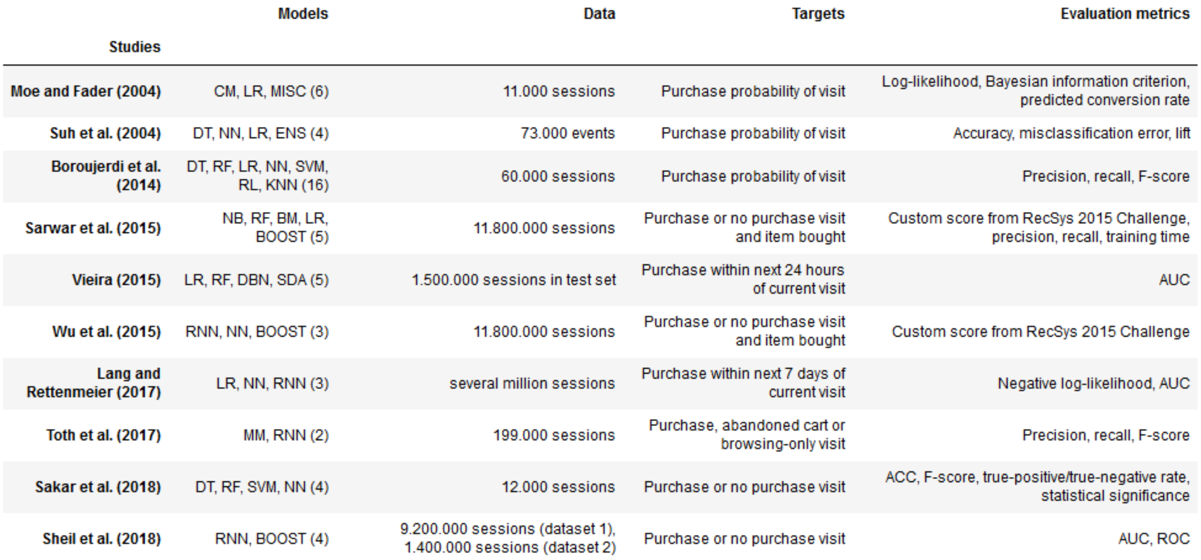
3.2. Comparative Machine Learning Studies Focused on Customer Journey Prediction in E-Commerce

While Section 3.1. presented 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 the specific application of purchase prediction in e-commerce is small compared to the body of literature concerned in Section 3.1., different criteria have been applied to the selection of relevant studies. Studies must be concerned with customer journey prediction, purchase prediction more precisely, 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 later experiments, which is desirable because methods and findings from these studies may be leveraged and transferred to the later experiments more easily.

It is worth noting that other models apart from machine learning and deep learning models have been 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 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 used 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. Logistic regression ranks on top with six occurrences, followed by neural networks with five, random forests and recurrent neural networks with each four and decision trees and boosting ensembles with each three occurrences. Nine out of ten studies use a single clickstream data set while only one study uses two clickstream data sets (Sheil et al., 2018). From some studies it is difficult to deduce the exact amount of data they used either due to business reasons (Lang and Rettenmeier, 2017) or because it is not clearly stated (Viera, 2016). The amount of data used in the selected studies varies widely from as few as a couple thousand sessions 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 visit leading to a purchase (Moe and Fader et al., 2004; Suh et al., 2004; Boroujerdi et al., 2014) while two studies predict whether a visit 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) additionally predict the item(s) likely to be bought in that visit. Vieira (2015) and Lang and Rettenmeier (2017) extend the time span within a visit is counted as a conversion 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 were not summarized. While most studies use some form of accuracy metric, such as accuracy, AUC, precision, recall or F-score (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 at least 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



Studies	Models	Data	Targets	Evaluation metrics
Moe and Fader (2004)	CM, LR, MISC (6)	11.000 sessions	Purchase probability of visit	Log-likelihood, Bayesian information criterion, predicted conversion rate
Suh et al. (2004)	DT, NN, LR, ENS (4)	73.000 events	Purchase probability of visit	Accuracy, misclassification error, lift
Boroujerdi et al. (2014)	DT, RF, LR, NN, SVM, RL, KNN (16)	60.000 sessions	Purchase probability of visit	Precision, recall, F-score
Sarwar et al. (2015)	NB, RF, BM, LR, BOOST (5)	11.800.000 sessions	Purchase or no purchase visit and item bought	Custom score from RecSys 2015 Challenge, precision, recall, training time
Vieira (2015)	LR, RF, DBN, SDA (5)	1.500.000 sessions in test set	Purchase within next 24 hours of current visit	AUC
Wu et al. (2015)	RNN, NN, BOOST (3)	11.800.000 sessions	Purchase or no purchase visit and item bought	Custom score from RecSys 2015 Challenge
Lang and Rettenmeier (2017)	LR, NN, RNN (3)	several million sessions	Purchase within next 7 days of current visit	Negative log-likelihood, AUC
Toth et al. (2017)	MM, RNN (2)	199.000 sessions	Purchase, abandoned cart or browsing-only visit	Precision, recall, F-score
Sakar et al. (2018)	DT, RF, SVM, NN (4)	12.000 sessions	Purchase or no purchase visit	ACC, F-score, true-positive/true-negative rate, statistical significance
Sheil et al. (2018)	RNN, BOOST (4)	9.200.000 sessions (dataset 1), 1.400.000 sessions (dataset 2)	Purchase or no purchase visit	AUC, ROC

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 of their other models outperforms all other 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 as well. Sheil et al. (2018) find that LSTM performs best for one data set while BOOST performs best for the other data set they used. Sarwar et al. (2015) find that BOOST outperforms all other models for the prediction of visits that lead to a purchase as well. Vieira (2015) builds another specialized NN, namely a stacked denoising auto-encoder, that he reports performed 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 a certain pattern, 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 subsequent experiments to include both frequently used models in general as well as models that tend to be frequently and successfully used for the application of 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 9, NB with 8 and finally RF and RNN with each 4 occurrences 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, and to allow for a more comprehensive general evaluation, both are considered in the subsequent 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 has been applied across the selected studies shown in Table 2. 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 will be done in Section 5 in detail.

4. Model Evaluation Framework

Anderl et al. (2014) condense, among other, the research on the application and acceptance of marketing models mentioned above into an evaluation framework 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 purchase events in an online shop (i.e. conversions) or revenue (Anderl et al., 2014, p. 7). Objectivity originates from Lilien’s claim for a model to allow for the computation of a variable’s relative impact and the objective evaluation of available decision options (Lilien, 2011, p. 198). ***Predictive accuracy*** is defined as a model’s ability to correctly predict conversions (Anderl et al., 2014, pp. 8), picking up Lodish's lesson of the importance of a model's credibility to persuade managers (Lodish, 2001, p. 54). ***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 requirement for a model to return useful results (Little, 1974, p. 470; Little, 2004, p. 1843). According to Little, models should be simple (Little, 1974, p. 470; Little, 2004, p. 1843) and easy to communicate with (Little, 1974, p. 470; Little, 2004, pp. 1844), 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 (pp. 8). ***Versatility*** incorporates Little's requirements that models should be easy to control (Little, 1974, p. 470; Little, 2004, pp. 1843-1844) and to adapt (Little, 1974, p. 470; Little, 2004, pp. 1844), 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 lesson that models should ideally deliver results on-demand (Lodish, 2001, p. 54), 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 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) 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?" (p. 54). 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) designed 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.

5. Experiments

Section 5.1. presents the setup of the experiments, covering the choice of models, target, features and data in general as well as naming 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 to be used for modeling. 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 along with explanations, stating and justifying important choices that have been 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, i.e. the reasoning behind the choices of models, target, features, training and test sets and methods applied to process and model the data.

5.1. Experimental Setup

A selection of twelve models has been 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 model.

The studies presented in Section 3.2. used different targets for predicting purchases, all of which are justifiable and each having different merits and drawbacks. One possibility is to predict whether a given session leads to a purchase (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 purchase (e.g. Moe and Fader, 2004; Suh et al., 2004; Boroujerdi et al., 2014). Another approach is to allow for a purchase 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 purchases 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 purchase, 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), several samples of different sizes are used in the experiments, ranging from several hundreds of thousands 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 have been 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 contain. For example, a 500.000 unique visitors sample contains all the unique visitors present in a 250.000 unique visitors sample and additionally 250.000 unique visitors that are different from those in the 250.000 unique visitors sample. 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. The number of unique visitors is used to specify the size of a sample instead of the actual number of sessions 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 either in the training or the test set but not split across both), thus capturing customer journeys and corresponding customer behavior in their entirety.

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

5.2. Data

The data used for this thesis stem from a large Swiss e-commerce website, comprising 63 GB and spanning six months from May to October 2016. The data can be understood as a sequence of events that capture visitors' clicking behavior and contain visitor-level information, such as device type, operating system and the marketing channel via which the visitor came. 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 have been 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. 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 were 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 was reduced by about 1.8 percent after cleaning the raw data, i.e. dropping rows with missing values or broken records. Missing values in the remaining rows were filled according to context, e.g. tagged as *Not Specified* if applicable. The amount of removed rows varied from 0.3 to 3.7 percent per month. There were 138 columns in the raw data of which 42 were accurately considered to contain useful information. After splitting the observations in certain columns, the number of columns increased to 48. Since certain columns were encoded, mappings of codes and strings that represent the actual information needed to be done using special mapping files. Some columns were already casted to the right data type and format in this processing step as well. There number of unique visitors was reduced from 4.7 to 4.5 million unique visitors in this processing step.

Aggregation. Lang and Rettenmeier (2017) find in their experiments that predictive performance is not substantially 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 were aggregated from event to session level. Datetime and numerical columns were aggregated using appropriate aggregators, such as the sum, the minimum or the maximum. Categorical columns were aggregated saving their first and last occurrence within a session. This led 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 was increased from 48 to 77, mainly due to saving the first and the last occurrence of each categorical column separately.

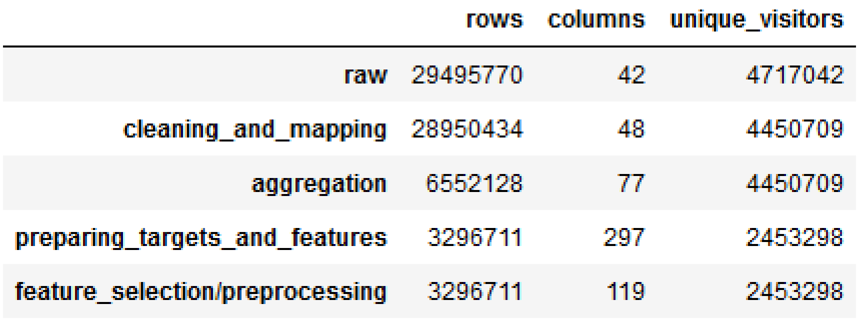
Preparing targets and features. Sessions containing only a single event (bounce sessions) were removed to reduce noise and because more engaged visitors are more interesting from a marketing executive's perspective, assuming that more active visitors have a higher propensity to purchase. Removing bounce sessions reduced the number of sessions to 3.3 million and the number of unique visitors to 2.5 million. This is a similar but less drastic step compared to Lang and Rettenmeier (2017) who only consider customers with a least 15 previous actions and Toth et al. (2017) who include only sessions with at least five events. The target, capturing a purchase within the next 24 hours of a given visit, was 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 was less than 0.1 percent of a categorical column’s most frequently occurring level were grouped in a level named *Other*. Then, the categorical columns were one-hot-encoded while each categorical column’s first dummy variable was dropped to avoid the dummy trap of multicollinearity. User age has been discretized into six bins, capturing typical age groups relevant for marketing (i.e. 14 to 25 years, 26 to 35 years etc.). Geographical information has been 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 have been created indicating month, day of month, day of week and hour of day. To reduce the number of resulting dummy variables, these time features have been mildly grouped to a more general representation (i.e. 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 in past visits seems to be at least partly indicative of whether a given visit leads to a purchase (e.g. Lang and Rettenmeier, 2017). Therefore, additional features have been created indicating whether a given visitor had a session or made a purchase within the last hours and days. Similar features have been created to capture the number of page and product views in the last visit as well. 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 visit leads to a purchase. Processing categorical columns and feature engineering increased the number of columns from 77 to 297. Those 297 columns include four identifiers that were removed before training and testing the models and are not considered in the following paragraph on actual features.

Feature selection. Three measures have been 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 targets are removed, e.g. a feature that indicates whether the customer reached the checkout step. Second, only the categorical columns aggregated by first occurrence were kept. This reduced the number of actual features from 293 to 172. Keeping categorical columns aggregated by both first and last occurrence would lead to a lot of redundancy in the data, justifying this step. Nevertheless, it is probable that there are differences between the two aggregation modes for certain columns, but testing for those would go beyond the scope of the present thesis and is therefore left to be explored in further experiments. Third, although certain models incorporate measures to assess the importance of individual features (e.g. coefficients in LR and SVM or measures of 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). In addition, feature selection should be done on training data only so that models can be tested on unseen data to avoid bias.

Therefore, feature selection is done on a training set containing 100.000 unique visitors because all other larger samples overlap with this sample's training and test sets in terms of unique visitors for the reasons specified in Section 5.1. 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 features in the 100.000 unique visitor training set. 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-values*. The *F-test*'s null hypothesis of a given feature not contributing to the prediction of the target has been rejected at the one percent significance level for 115 features (Appendix, Table 3). The selection includes 23 numerical, 11 boolean and 10 categorical features (Appendix, Table 4). 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 standardizing numerical and encoding categorical features. This is why dummy variables representing categorical features' individual levels were considered in the feature selection rather than categorical features capturing all levels in one feature. Consequently, some dummy variables representing levels of categorical features have been found to be insignificant and therefore not selected while dummy variables representing other levels of the same categorical features have been 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. Therefore, there is certainly more room for experimentation regarding feature selection.

Preprocessing. The preprocessing steps described in the paragraph above were applied to all other samples as well to prepare them for modeling. In summary, only categorical columns aggregated by first occurrences were kept and those aggregated by last occurrences were discarded, then the numerical features were standardized by removing the mean and scaling to unit variance and finally, all but those 115 features identified through feature selection were removed from training and test sets. Table 3 summarizes the number of rows, columns (including identifiers and actual features) and unique visitors after each of the processing stages explained above.

Table 3

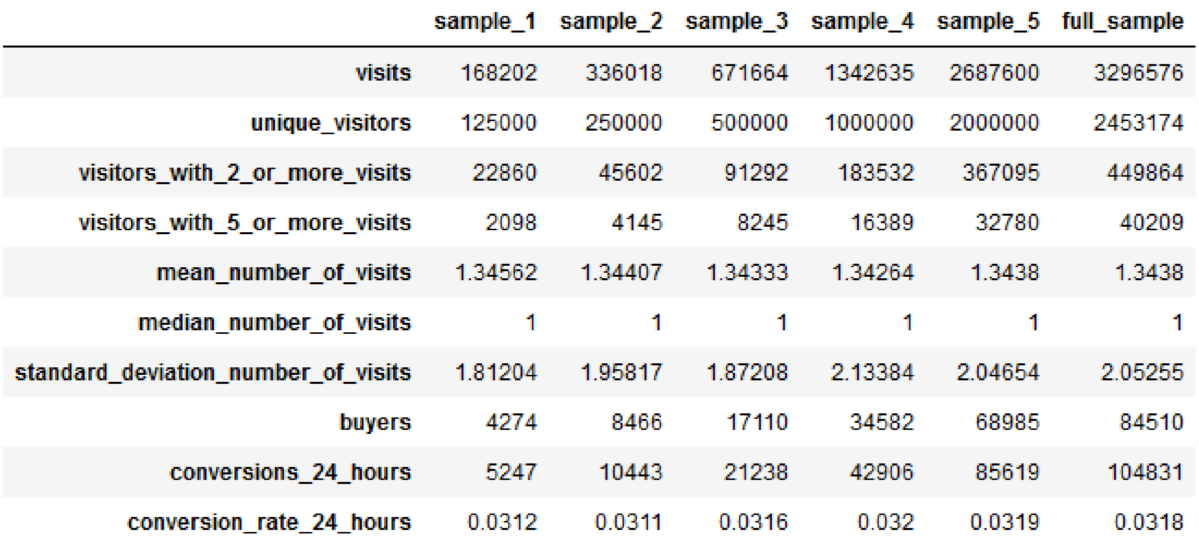


	rows	columns	unique_visitors
raw	29495770	42	4717042
cleaning_and_mapping	28950434	48	4450709
aggregation	6552128	77	4450709
preparing_targets_and_features	3296711	297	2453298
feature_selection/preprocessing	3296711	119	2453298

5.3. Descriptive Statistics

Table 4 presents descriptive statistics for the five 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 balancing the samples has been successful. 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 two percent have five or more sessions in the period under consideration. The average number of sessions per visitor is 1.3, the according median is 1 and the standard deviation is about two sessions per visitor. The share of buyers among all visitors is about 3.4 percent. About 20 percent of purchases are repeated conversions. The conversion rate across all samples and the entire data set ranges from 3.1 to 3.5 percent, dependent on sample and target. 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 topic to be investigated in future research.

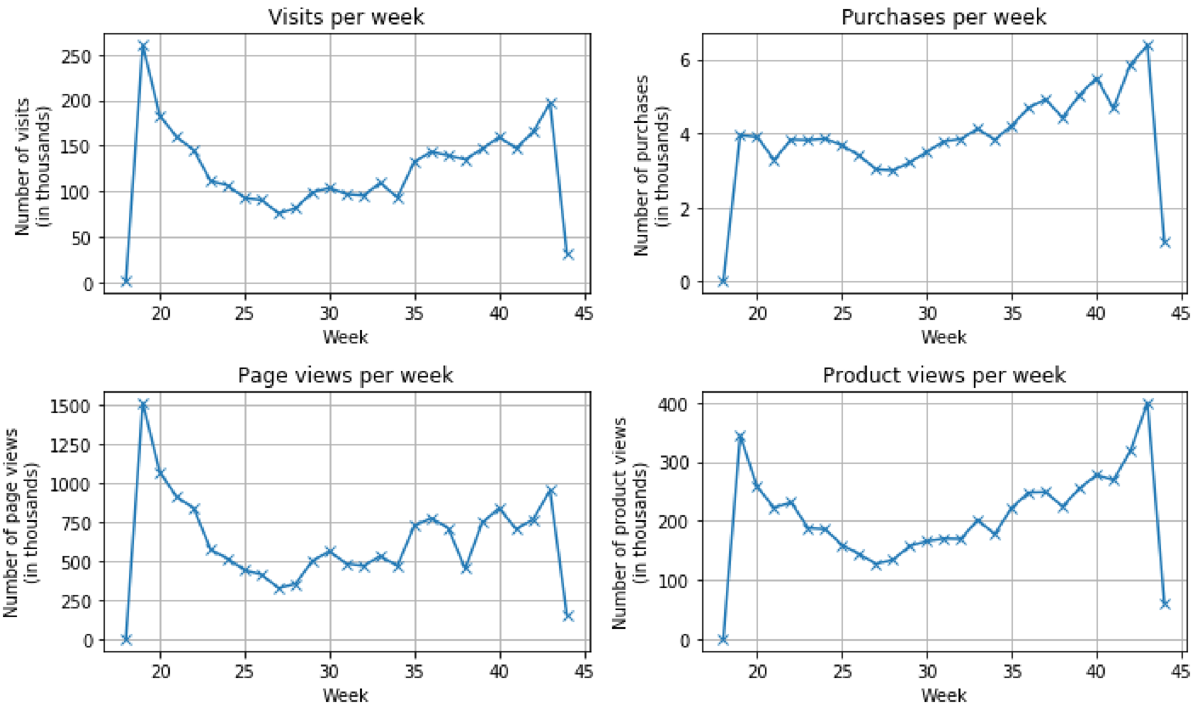
Table 4



	sample_1	sample_2	sample_3	sample_4	sample_5	full_sample
visits	168202	336018	671664	1342635	2687600	3296576
unique_visitors	125000	250000	500000	1000000	2000000	2453174
visitors_with_2_or_more_visits	22860	45602	91292	183532	367095	449864
visitors_with_5_or_more_visits	2098	4145	8245	16389	32780	40209
mean_number_of_visits	1.34562	1.34407	1.34333	1.34264	1.3438	1.3438
median_number_of_visits	1	1	1	1	1	1
standard_deviation_number_of_visits	1.81204	1.95817	1.87208	2.13384	2.04654	2.05255
buyers	4274	8466	17110	34582	68985	84510
conversions_24_hours	5247	10443	21238	42906	85619	104831
conversion_rate_24_hours	0.0312	0.0311	0.0316	0.032	0.0319	0.0318

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 increase of activity toward the autumn weeks and occasional peaks may be caused by temporal marketing campaigns or seasonality effects (e.g. holiday season during the summer weeks).

Figure 1



5.4. Models

Default hyperparameter settings have been chosen for most models (if possible) 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 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 have not been trained using GPUs, which would have probably increased their training speed, but instead CPUs were used for training like for all other models. The neural networks with one, three and five hidden layers, respectively, have been built in an analogous manner so that the following explanations in this paragraph apply to all three of them. 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. 115, 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 choice of activation functions, weight initialization and the optimizer follows Lang and Rettenmeier's (2017) choices for their NN and RNN. The number of epochs during training, i.e. the number of iterations over the entire training set, has been set to ten, following Toth et al. (2017). The batch size during training, i.e. the number of instances per gradient update, has been set to 256 instances, which is eight times the default batch size of 32. To speed up training if possible, early stopping has been configured so that training is ended early if validation loss has not been minimized in two subsequent epochs. The choice of batch size and early stopping follows Sheil et al. (2018). 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 has been 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 "(…) 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 and exploding 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 this thesis' experiments to show the potential superiority of LSTM over traditional RNN and the importance of storing long-term information in customer journey prediction. The RNN and LSTM used in this thesis 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 due to their rather complex architectures which is why both models are built using only one recurrent layer, following Lang and Rettenmeier (2017) and Toth et al. (2017). Each model's recurrent layer consist of 256 RNN and LSTM cells, respectively, since Sheil et al. (2018) find 256 cells per recurrent layer to be optimal in their 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 were explicitly designed to capture past customer behavior have been excluded from the training and test sets used for RNN and LSTM. This step allows to investigate their alleged superiority over vector-based models that rely on feature engineering. 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 were removed, reducing the number of features from 115 to 101 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 have been 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.

6. Evaluation of Models and Results

Section 6.1. evaluates the models in terms of the criterium of objectivity. Section 6.2. analyzes the experimental results considering the criterium of predictive accuracy and presents a variety of different metrics and methods to evaluate model performance. To investigate the robustness of the experimental results, Section 6.3. presents cross-validation results for each model. 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 why Section 6.6. analyzes the models' efficiency considering the time they required for training and testing on samples of different sizes. The evaluation of the models and the experimental results constitutes a central part of this thesis and forms the foundation for the subsequent discussion and the derivation of managerial implications in Section 7.

6.1. 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 they 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 these methods are features capturing the time passed since the last purchase, cart-related events and visitor-specific features like gender and age. 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 had on a prediction. Same applies to the Gaussian NB used in the experiment above. Other implementations of

NB, namely Multinomial and Bernoulli, however, have available ways to return coefficients that indicate a feature's importance (Pedregosa et al., 2011). Lang and Rettenmeier (2017) state that RNN and LSTM 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. 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 and LSTM, they only partly satisfy the objectivity criterion since it is not straight forward to create such visualizations and one has to predict conversion probabilities rather than a binary conversion outcome like in the experiments above.

6.2. 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). First, the effect of sample size on predictive accuracy is investigated through learning curves and second, performance metrics for different samples are examined in more detail.

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 dependent on the amount of training data (Shavlik et al., 1997; Perlich et al., 2003). Figure 2 shows for each model the relationship of AUC and the number of unique visitors in the training set. Test set AUC is on the y-axis and the number of unique visitors in the training set is on the x-axis. Computing learning curves using cross-validation would probably deliver more robust results but is computationally expensive and time-consuming, especially for large sample sizes, which is why the present learning curve analysis refrained from it. Besides, it is noteworthy that the feature selection that has been performed on the 100.000 unique visitors training set has been applied to the training sets with 3.125, 6.250, 12.500, 25.000 and 50.0000, respectively, even though these samples are contained in the data used for feature selection. This could cause a bias in the affected data but is considered negligible since the learning curve analysis first and foremost serves the purpose of showing that the minimum amount of training data to yield stable performance is different for different models. The learning curves of LR, DT, RF, SVM, BOOST, NN1, NN3 and NN5 move between AUC scores of 0.8 and 0.9, indicating that they are fairly robust to variations of sample size and most of these models are able to reach their peak performance with comparably small amounts of data, e.g. 50.000 or 100.000 unique visitors in the training set. NB performs poorly for the two smallest sample but reaches its peak performance from the 12.500 unique visitors training set on. KNN continuously improves with more data bit is unable to reach a performance level similar to most other models. The learning curves of KNN show how AUC slightly increases with growing sample size. The performance of RNN and LSTM steadily improves until the 100.000 unique visitors, then the performance improvements become smaller and they finally reach a level similar to other high-performing models. Figure 2 does not report an AUC score for RNN (like Table 8) because validation loss during training turned indefinite which is an indication of exploding gradients, a typical issue of recurrent networks that is addressed by LSTM (Hochreiter and Schmidhuber, 1997). Therefore, no performance metrics could be computed for RNN for the 1.600.000 unique visitors sample.

LR, KNN and SVM are strictly dominated by all other models from the 100.000 unique visitors training set on. NB reports the highest AUC from the 12.500 visitors training set on only exceeded by RNN in the 800.000 unique visitors training set. The other best performing models in the largest sample are NN3, LSTM, BOOST, NN1 and NN5, followed by DT and RF.

Figure 2 – undecided which one to keep

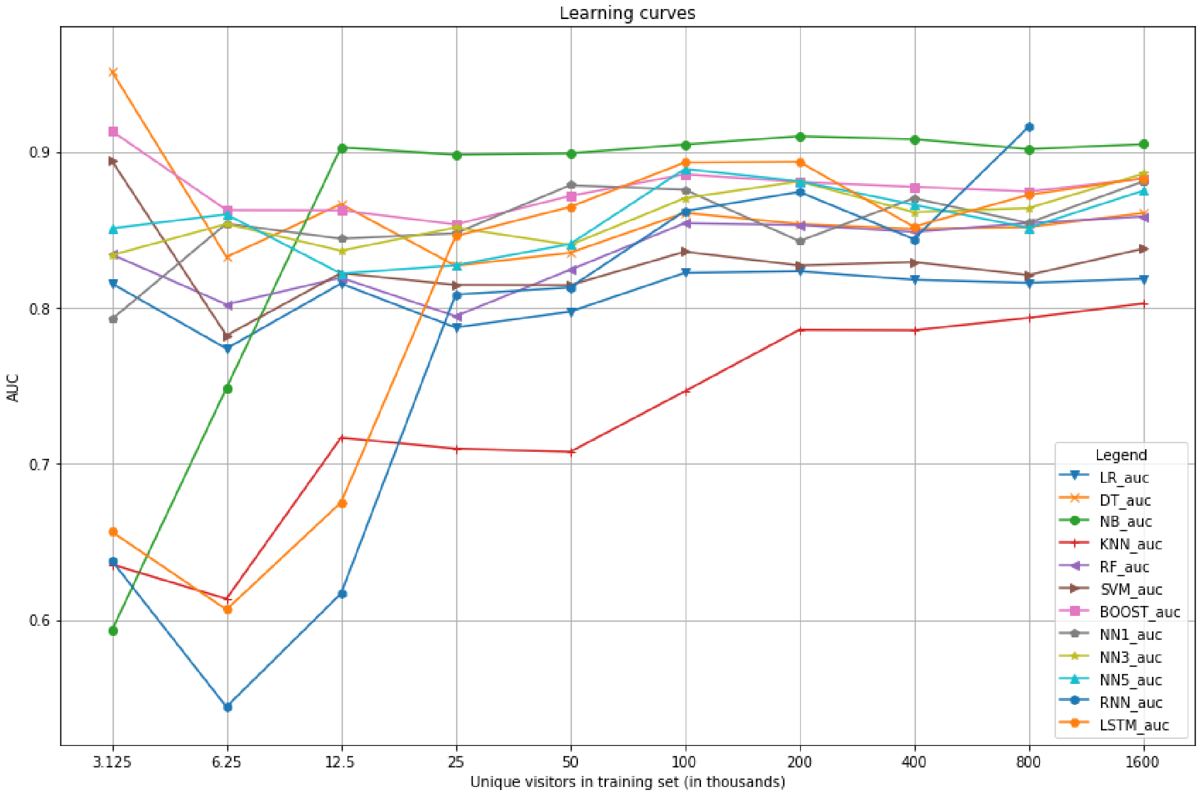
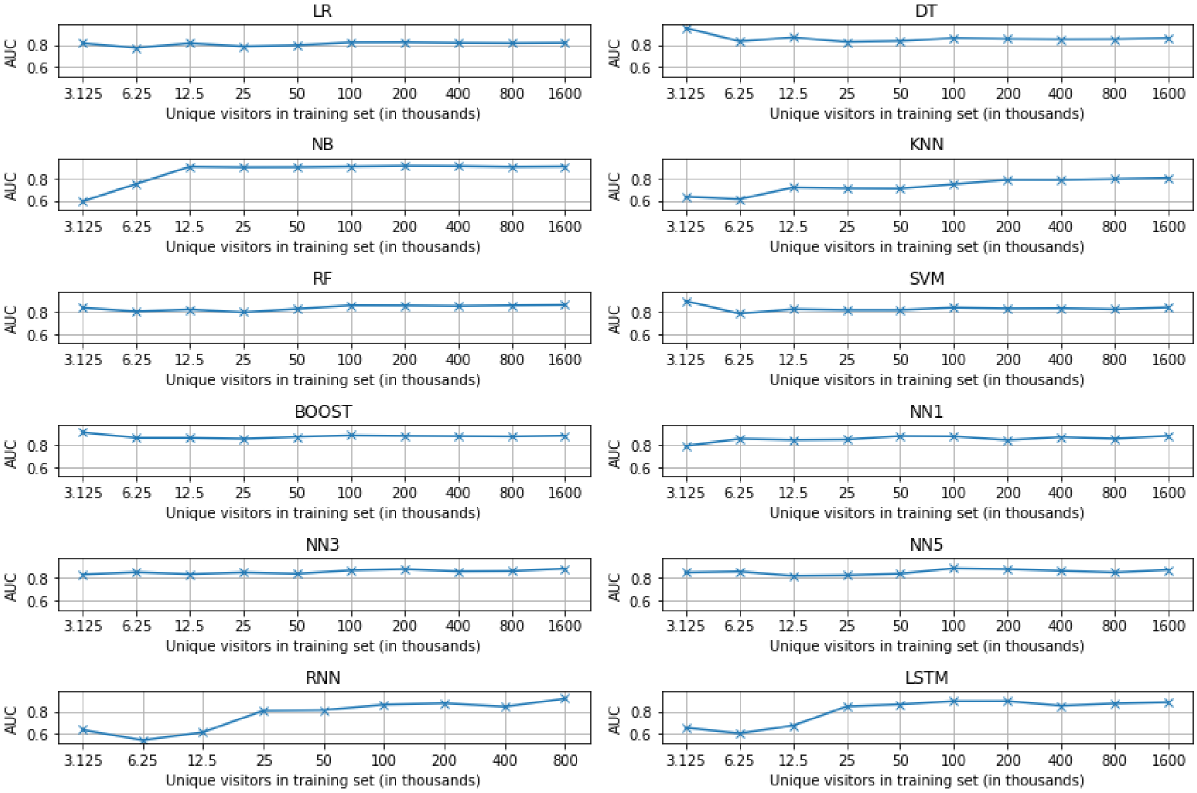
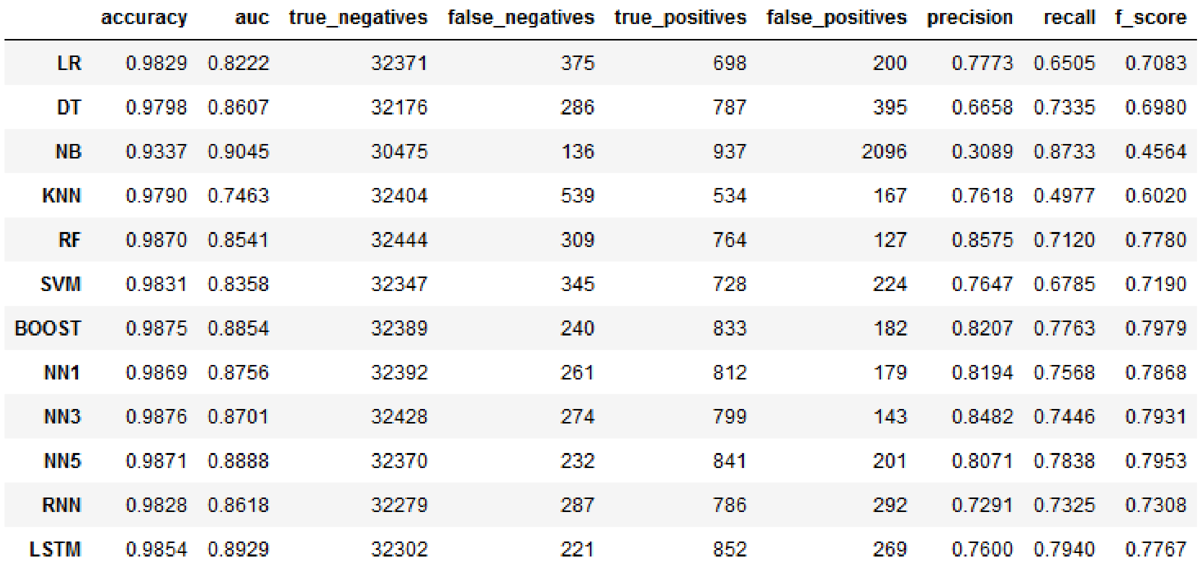


Figure 2 – undecided which one to keep



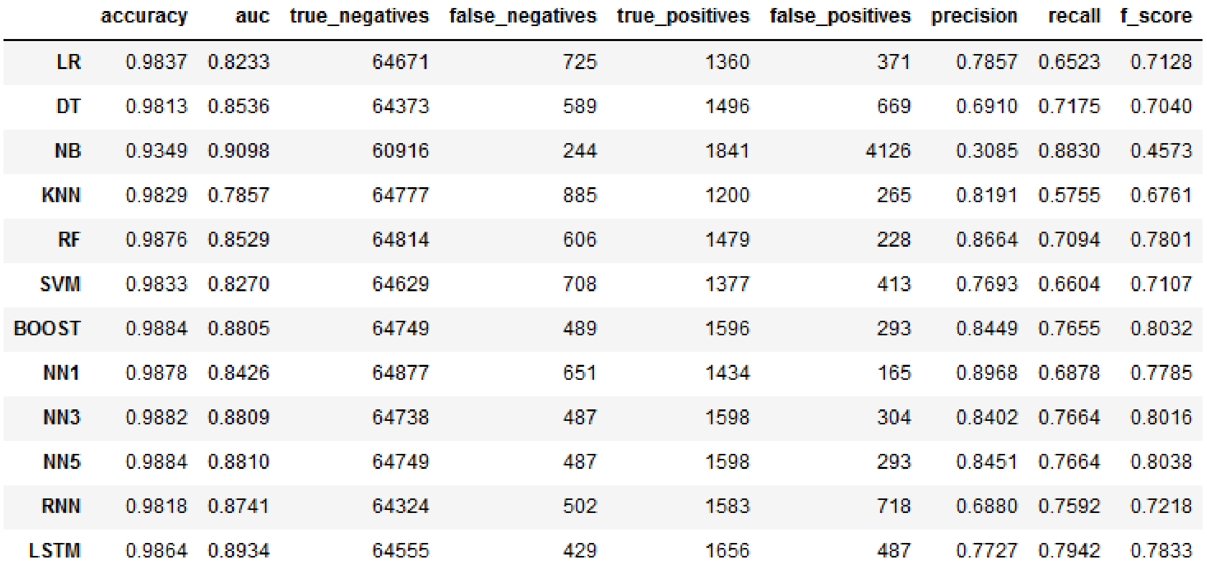
Tables 5 to 9 show the models' predictive performances in terms of accuracy, AUC, true negatives, false negatives, true positives and false positives for five samples of different sizes, respectively. Accuracy is close to 100 percent for all models across all samples but is not a suitable measure of predictive performance due to the highly imbalanced classes in the data (conversion rate of roughly three percent), resulting in a bias towards the majority class, i.e. visits that do not lead to a conversion. Alternative performance metrics that are able to better account for class imbalance are AUC, precision, recall and F-score.

Table 5 – 100k unique visitors



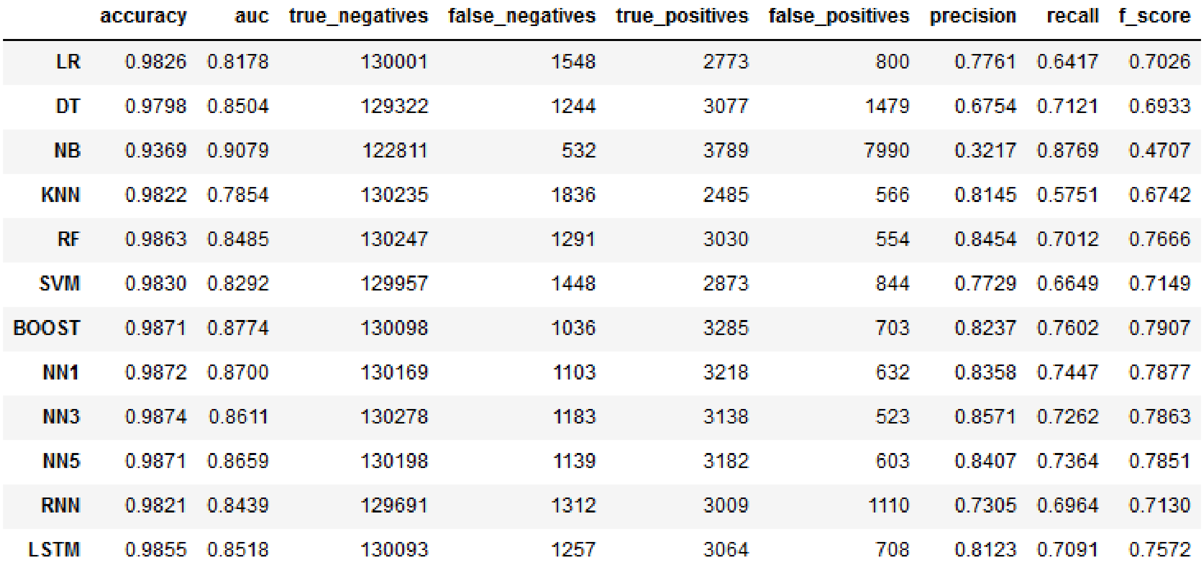
	accuracy	auc	true_negatives	false_negatives	true_positives	false_positives	precision	recall	f_score
LR	0.9829	0.8222	32371	375	698	200	0.7773	0.6505	0.7083
DT	0.9798	0.8607	32176	286	787	395	0.6658	0.7335	0.6980
NB	0.9337	0.9045	30475	136	937	2096	0.3089	0.8733	0.4564
KNN	0.9790	0.7463	32404	539	534	167	0.7618	0.4977	0.6020
RF	0.9870	0.8541	32444	309	764	127	0.8575	0.7120	0.7780
SVM	0.9831	0.8358	32347	345	728	224	0.7647	0.6785	0.7190
BOOST	0.9875	0.8854	32389	240	833	182	0.8207	0.7763	0.7979
NN1	0.9869	0.8756	32392	261	812	179	0.8194	0.7568	0.7868
NN3	0.9876	0.8701	32428	274	799	143	0.8482	0.7446	0.7931
NN5	0.9871	0.8888	32370	232	841	201	0.8071	0.7838	0.7953
RNN	0.9828	0.8618	32279	287	786	292	0.7291	0.7325	0.7308
LSTM	0.9854	0.8929	32302	221	852	269	0.7600	0.7940	0.7767

Table 6 – 200k unique visitors



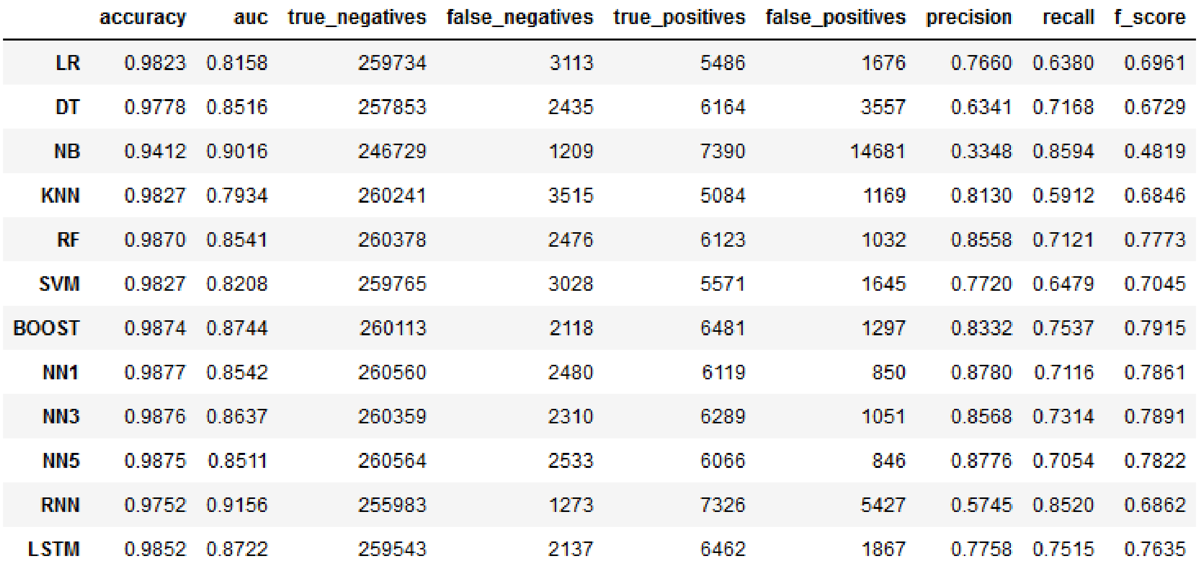
	accuracy	auc	true_negatives	false_negatives	true_positives	false_positives	precision	recall	f_score
LR	0.9837	0.8233	64671	725	1360	371	0.7857	0.6523	0.7128
DT	0.9813	0.8536	64373	589	1496	669	0.6910	0.7175	0.7040
NB	0.9349	0.9098	60916	244	1841	4126	0.3085	0.8830	0.4573
KNN	0.9829	0.7857	64777	885	1200	265	0.8191	0.5755	0.6761
RF	0.9876	0.8529	64814	606	1479	228	0.8664	0.7094	0.7801
SVM	0.9833	0.8270	64629	708	1377	413	0.7693	0.6604	0.7107
BOOST	0.9884	0.8805	64749	489	1596	293	0.8449	0.7655	0.8032
NN1	0.9878	0.8426	64877	651	1434	165	0.8968	0.6878	0.7785
NN3	0.9882	0.8809	64738	487	1598	304	0.8402	0.7664	0.8016
NN5	0.9884	0.8810	64749	487	1598	293	0.8451	0.7664	0.8038
RNN	0.9818	0.8741	64324	502	1583	718	0.6880	0.7592	0.7218
LSTM	0.9864	0.8934	64555	429	1656	487	0.7727	0.7942	0.7833

Table 7 – 400k unique visitors



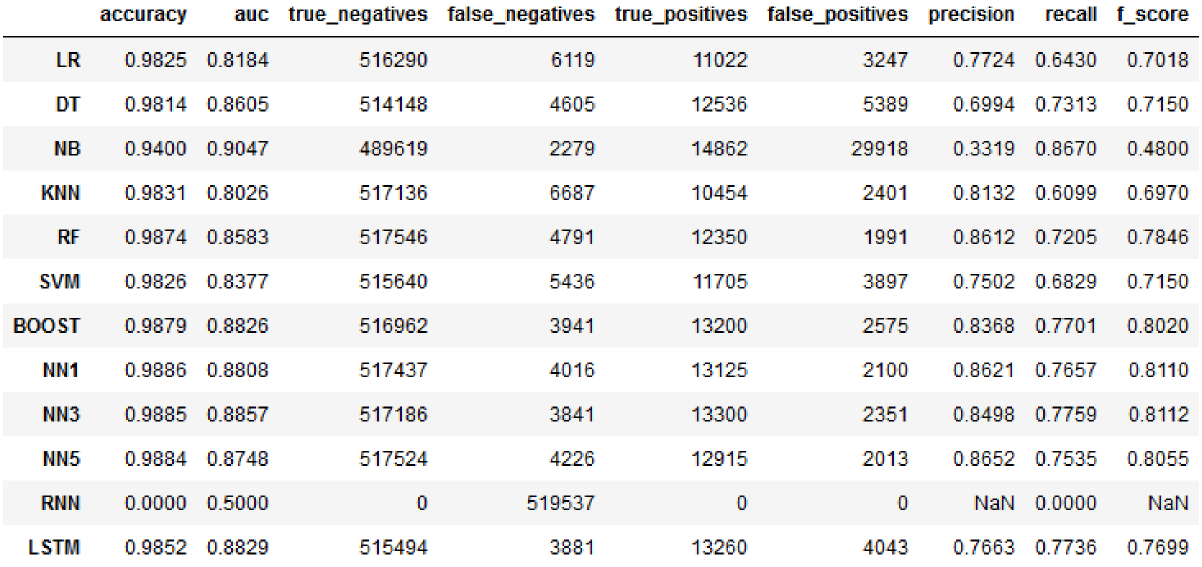
	accuracy	auc	true_negatives	false_negatives	true_positives	false_positives	precision	recall	f_score
LR	0.9826	0.8178	130001	1548	2773	800	0.7761	0.6417	0.7026
DT	0.9798	0.8504	129322	1244	3077	1479	0.6754	0.7121	0.6933
NB	0.9369	0.9079	122811	532	3789	7990	0.3217	0.8769	0.4707
KNN	0.9822	0.7854	130235	1836	2485	566	0.8145	0.5751	0.6742
RF	0.9863	0.8485	130247	1291	3030	554	0.8454	0.7012	0.7666
SVM	0.9830	0.8292	129957	1448	2873	844	0.7729	0.6649	0.7149
BOOST	0.9871	0.8774	130098	1036	3285	703	0.8237	0.7602	0.7907
NN1	0.9872	0.8700	130169	1103	3218	632	0.8358	0.7447	0.7877
NN3	0.9874	0.8611	130278	1183	3138	523	0.8571	0.7262	0.7863
NN5	0.9871	0.8659	130198	1139	3182	603	0.8407	0.7364	0.7851
RNN	0.9821	0.8439	129691	1312	3009	1110	0.7305	0.6964	0.7130
LSTM	0.9855	0.8518	130093	1257	3064	708	0.8123	0.7091	0.7572

Table 8 – 800k unique visitors



	accuracy	auc	true_negatives	false_negatives	true_positives	false_positives	precision	recall	f_score
LR	0.9823	0.8158	259734	3113	5486	1676	0.7660	0.6380	0.6961
DT	0.9778	0.8516	257853	2435	6164	3557	0.6341	0.7168	0.6729
NB	0.9412	0.9016	246729	1209	7390	14681	0.3348	0.8594	0.4819
KNN	0.9827	0.7934	260241	3515	5084	1169	0.8130	0.5912	0.6846
RF	0.9870	0.8541	260378	2476	6123	1032	0.8558	0.7121	0.7773
SVM	0.9827	0.8208	259765	3028	5571	1645	0.7720	0.6479	0.7045
BOOST	0.9874	0.8744	260113	2118	6481	1297	0.8332	0.7537	0.7915
NN1	0.9877	0.8542	260560	2480	6119	850	0.8780	0.7116	0.7861
NN3	0.9876	0.8637	260359	2310	6289	1051	0.8568	0.7314	0.7891
NN5	0.9875	0.8511	260564	2533	6066	846	0.8776	0.7054	0.7822
RNN	0.9752	0.9156	255983	1273	7326	5427	0.5745	0.8520	0.6862
LSTM	0.9852	0.8722	259543	2137	6462	1867	0.7758	0.7515	0.7635

Table 9 – 1600k unique visitors



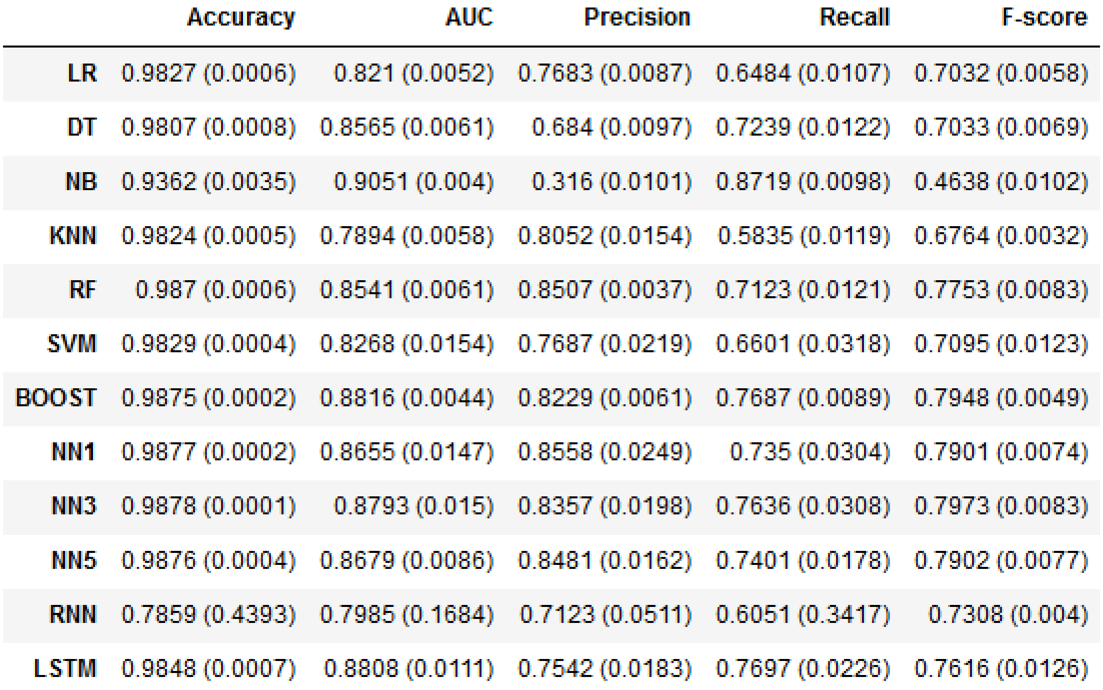
	accuracy	auc	true_negatives	false_negatives	true_positives	false_positives	precision	recall	f_score
LR	0.9825	0.8184	516290	6119	11022	3247	0.7724	0.6430	0.7018
DT	0.9814	0.8605	514148	4605	12536	5389	0.6994	0.7313	0.7150
NB	0.9400	0.9047	489619	2279	14862	29918	0.3319	0.8670	0.4800
KNN	0.9831	0.8026	517136	6687	10454	2401	0.8132	0.6099	0.6970
RF	0.9874	0.8583	517546	4791	12350	1991	0.8612	0.7205	0.7846
SVM	0.9826	0.8377	515640	5436	11705	3897	0.7502	0.6829	0.7150
BOOST	0.9879	0.8826	516962	3941	13200	2575	0.8368	0.7701	0.8020
NN1	0.9886	0.8808	517437	4016	13125	2100	0.8621	0.7657	0.8110
NN3	0.9885	0.8857	517186	3841	13300	2351	0.8498	0.7759	0.8112
NN5	0.9884	0.8748	517524	4226	12915	2013	0.8652	0.7535	0.8055
RNN	0.0000	0.5000	0	519537	0	0	NaN	0.0000	NaN
LSTM	0.9852	0.8829	515494	3881	13260	4043	0.7663	0.7736	0.7699

- TBD: briefly explain and reference AUC, precision, recall and F-score
- TBD: statistical (non-parametric) tests for model comparison, e.g. t-test, Wilcoxon test, Friedman test → Dietterich (1998), Alpaydin (1999), Demsar (2006), Lavesson and Davidsson (2007), Raschka (2018)

6.3. 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 has been tested using 5-fold cross-validation and a medium-sized sample with 500.000 unique visitors. Five folds and a sample of medium size have been chosen to speed up the otherwise computationally expensive and time-consuming cross-validation procedure. 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. Table 9 reports averaged accuracy, AUC, precision, recall and F-scores computed in the cross-validation procedure. Standard deviations are in parenthesis. Most models are robust as confirmed by the low standard deviations of the corresponding metrics. The number of hidden layers appears to have no effect on the robustness of NN1, NN3 and NN5, although the number of hidden layers in these networks is admittedly low and large differences not to be expected. For RNN, accuracy, AUC and recall, however, vary substantially. For LSTM, recall shows variation as well, making both models less robust.

Table 10



	Accuracy	AUC	Precision	Recall	F-score
LR	0.9827 (0.0006)	0.821 (0.0052)	0.7683 (0.0087)	0.6484 (0.0107)	0.7032 (0.0058)
DT	0.9807 (0.0008)	0.8565 (0.0061)	0.684 (0.0097)	0.7239 (0.0122)	0.7033 (0.0069)
NB	0.9362 (0.0035)	0.9051 (0.004)	0.316 (0.0101)	0.8719 (0.0098)	0.4638 (0.0102)
KNN	0.9824 (0.0005)	0.7894 (0.0058)	0.8052 (0.0154)	0.5835 (0.0119)	0.6764 (0.0032)
RF	0.987 (0.0006)	0.8541 (0.0061)	0.8507 (0.0037)	0.7123 (0.0121)	0.7753 (0.0083)
SVM	0.9829 (0.0004)	0.8268 (0.0154)	0.7687 (0.0219)	0.6601 (0.0318)	0.7095 (0.0123)
BOOST	0.9875 (0.0002)	0.8816 (0.0044)	0.8229 (0.0061)	0.7687 (0.0089)	0.7948 (0.0049)
NN1	0.9877 (0.0002)	0.8655 (0.0147)	0.8558 (0.0249)	0.735 (0.0304)	0.7901 (0.0074)
NN3	0.9878 (0.0001)	0.8793 (0.015)	0.8357 (0.0198)	0.7636 (0.0308)	0.7973 (0.0083)
NN5	0.9876 (0.0004)	0.8679 (0.0086)	0.8481 (0.0162)	0.7401 (0.0178)	0.7902 (0.0077)
RNN	0.7859 (0.4393)	0.7985 (0.1684)	0.7123 (0.0511)	0.6051 (0.3417)	0.7308 (0.004)
LSTM	0.9848 (0.0007)	0.8808 (0.0111)	0.7542 (0.0183)	0.7697 (0.0226)	0.7616 (0.0126)

6.4. Interpretability

Models and their results should be simple and easy to communicate to foster acceptance and application by marketing managers and 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 light onto the concept and relevance of interpretability in machine learning, recent developments regarding this topic are briefly outlined. Guidotti et al. (2018) provide a much more comprehensive overview and summary of the research efforts in this field. In recent years, several studies have been published, stressing and debating the notions and importance of interpretability and explicability of machine learning models, some of which are briefly mentioned in the following. Despite the substantial amount of research that has been recently conducted on the topic of interpretability and explicability of machine learning models 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 studies that compare different machine learning models in terms of complexity and monotonicity, 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 fashion. 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 with all its available methods to make machine learning models more transparent would exceed the scope of this thesis. Explicitly investigating the aspect of interpretability in the comparison of machine learning and deep learning models for a specific use case in addition to testing the suggested methods 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 instead. 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 it 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 (depending on the tree size) reducing complexity, 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. 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 the ensembles is possible with the implementations of RF and BOOST used in the experiments above but allows only limited insights if the ensembles consist of dozens of 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 provides information regarding which features the model considered important for its prediction. Although NN1, NN3, NN5, RNN and LSTM are comparably 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 to be made when building these models. Besides, there is no straight forward way to compute the impact individual features had on a neural network's predictions. There is the, however not straight forward, possibility to apply specialized methods to establish a relationship between a neural network's predictions and its input features (e.g. Sundararajan et al., 2017).

6.5. Versatility

The criterion of versatility requires models to be easy to control and adaptive when conditions change over time, 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. If, however, the level of aggregation of the data is to be changed or a different target is to be used, models' flexibility depends on the degree to which different aggregations or targets change the underlying structure of the data or 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 needed to be substantially adapted if they can only poorly or not at all predict probabilities in a straight forward fashion, e.g. DT, NB, SVM and BOOST (Caruana and Niculescu-Mizil, 2006, p. 163).

6.6. Algorithmic Efficiency

Models should allow to be updated in a reasonable amount of time and to compute results fairly quickly to provide them to managers and 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 (Lim et al., 2000). Figure 3 shows training times in seconds for all models. Some models required less than one second for training for some training sets which is why their curves are close to zero and flat. NB appears to be by far the fastest model with only a couple of seconds for training and testing for even the largest sample 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 20 minutes. More complex models like RNN and LSTM tend to be slower in general and require several minutes for training on smaller samples up to a couple of hours for training on larger samples. It seems intuitive that the models require more time for training and testing with increasing sample size. KNN, however, appears to not scale well to large amounts of data. Increasing the number of unique visitors in the training set by a factor of 16 increased training time by a factor of more than 350 while training time for other models increased by a factor of much less, namely roughly ten to 100. Figure 4 shows the times the models required for testing for different sizes of the test set. Most models required 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 required more time for testing than for training. In comparison to the other models, testing took so much longer for KNN that its corresponding curve had to be plotted in a separate graph due to the substantial differences in scale. 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 the use case and business environment (Anderl et al., 2014, p. 22).

Figure 3

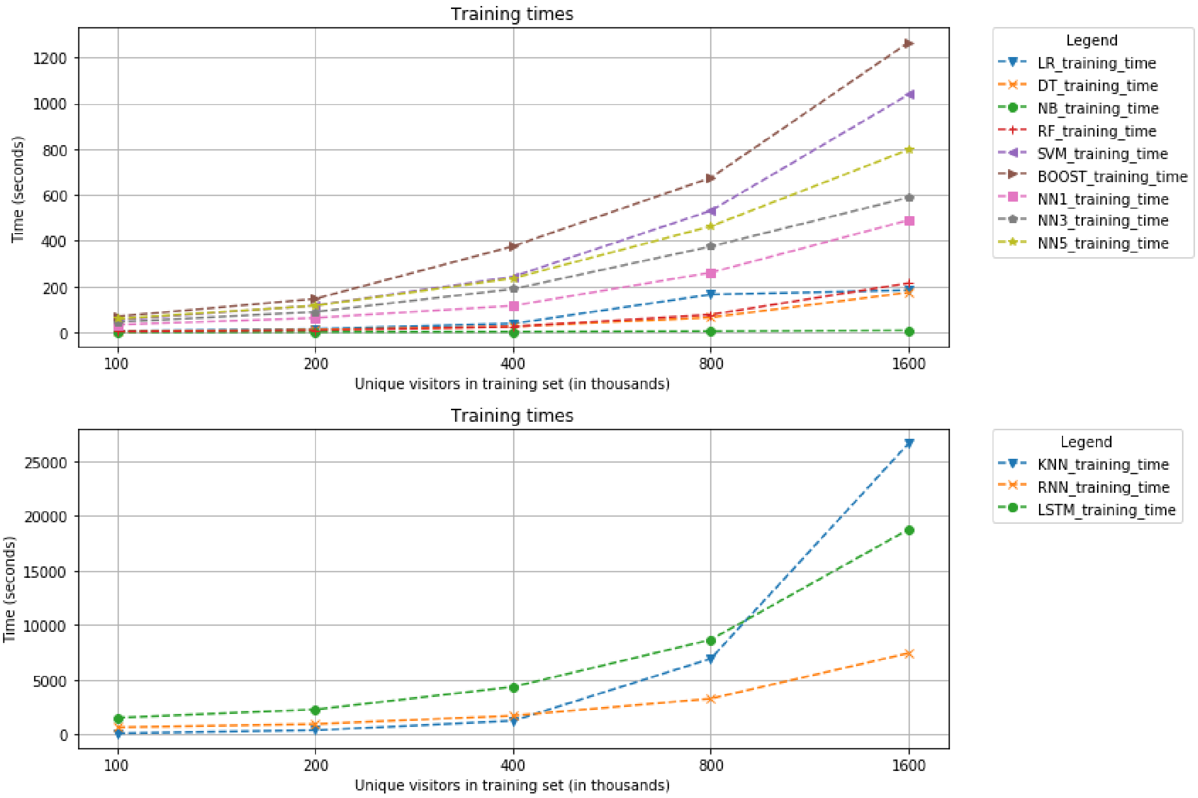
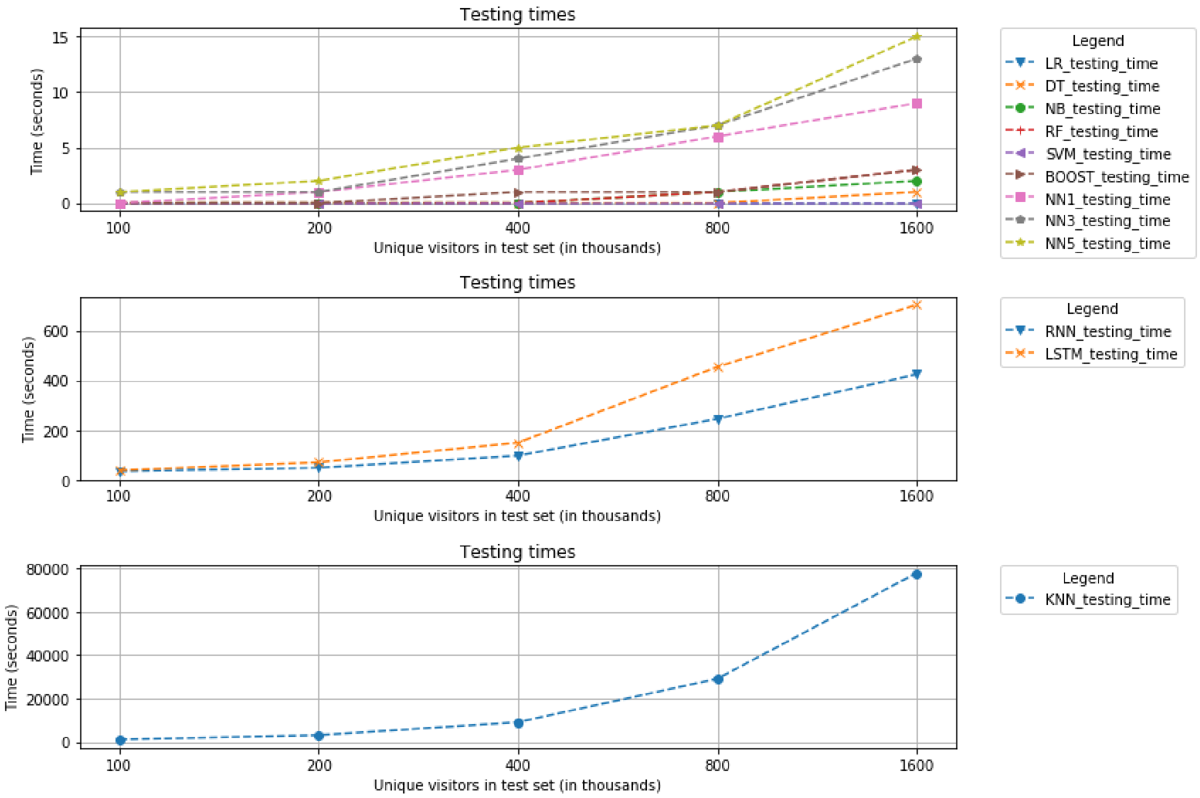


Figure 4



7. Discussion and Managerial Implications

- select one model and optimize it → Random Search (Bergstra and Bengio; 2012)
- feature importances if available for insight into which features appear to be relevant (e.g. top 10 most relevant features)
- inspiration: Sismeiro and Bucklin, 2004; Van den Poel and Bucklin, 2005 (features, literature review, managerial implications); Stange and Funk, 2015 (managerial implications)
- select model with regard to false positive and false negative: do you want to reach a large mass of potential buyers or do you want to reach visitors that are very likely to buy?
- Humans too oftentimes cannot explain themselves

8. Conclusion

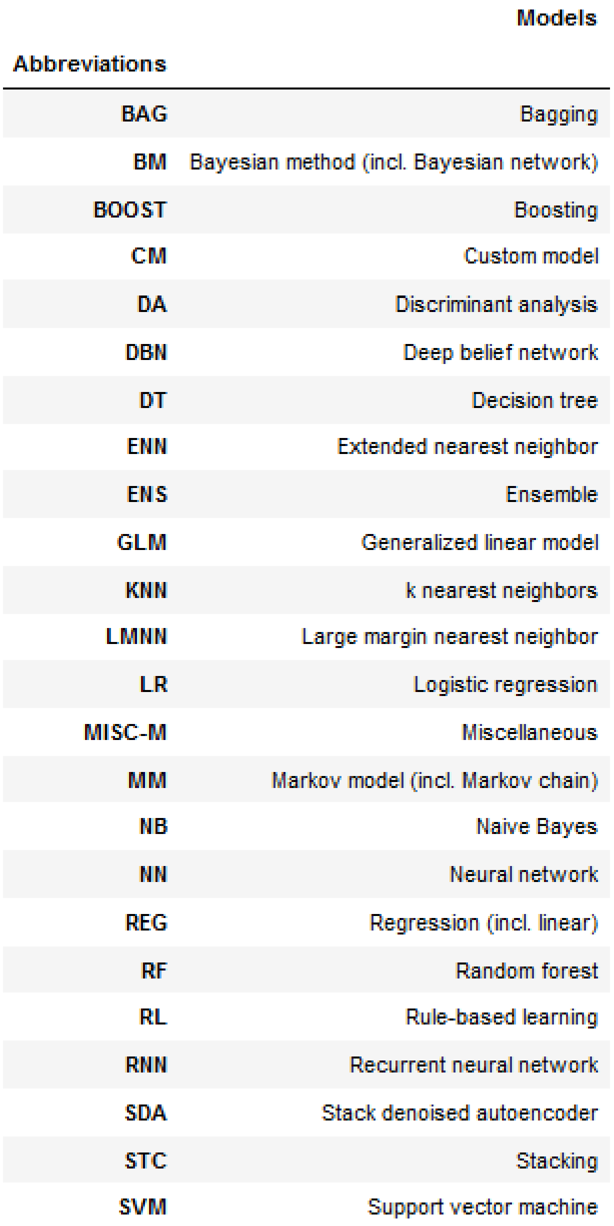
- Summary of research question, relevance, experiments and results
- Contributions to research and industry/practice
    - Research: profound meta-analysis of comparative machine learning literature
    - Industry/practice: Evaluation of traditional and novel models for customer journey prediction in e-commerce from the perspective of marketing managers and executives

    - (e.g. beyond just predictive accuracy) based on a formal theory-backed evaluation framework
- Limitations
    - Limited model selection → more models
    - Bias due to default hyperparameter settings → hyperparameter tuning for all models
    - Limited generalization due to one use case and data set only → more and different data and use cases + e.g. segmentation of visitors and separate models for individual visitor segments
- Additional ideas for future research
    - test effect of different targets/features → predict purchase probabilities rather than purchase or not purchase + extend to item prediction (e.g. recommender system)
    - expert interviews with marketing managers and executives on what really counts for them when it comes to ML and DL models
    - use tools such as LIME for model interpretability

References

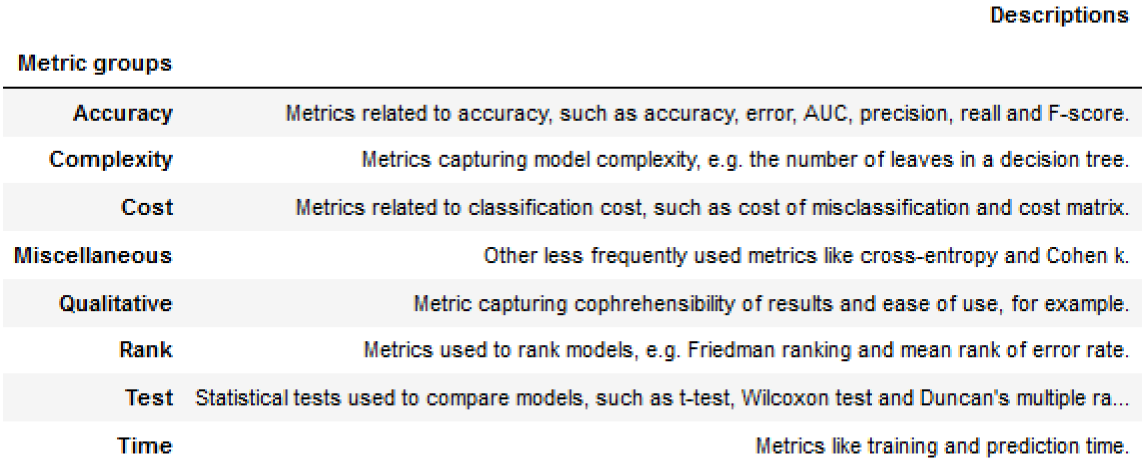
Appendix

Table 1



Abbreviations	Models
BAG	Bagging
BM	Bayesian method (incl. Bayesian network)
BOOST	Boosting
CM	Custom model
DA	Discriminant analysis
DBN	Deep belief network
DT	Decision tree
ENN	Extended nearest neighbor
ENS	Ensemble
GLM	Generalized linear model
KNN	k nearest neighbors
LMNN	Large margin nearest neighbor
LR	Logistic regression
MISC-M	Miscellaneous
MM	Markov model (incl. Markov chain)
NB	Naive Bayes
NN	Neural network
REG	Regression (incl. linear)
RF	Random forest
RL	Rule-based learning
RNN	Recurrent neural network
SDA	Stack denoised autoencoder
STC	Stacking
SVM	Support vector machine

Table 2



Metric groups	Descriptions
Accuracy	Metrics related to accuracy, such as accuracy, error, AUC, precision, reall and F-score.
Complexity	Metrics capturing model complexity, e.g. the number of leaves in a decision tree.
Cost	Metrics related to classification cost, such as cost of misclassification and cost matrix.
Miscellaneous	Other less frequently used metrics like cross-entropy and Cohen k.
Qualitative	Metric capturing cophrehensibility of results and ease of use, for example.
Rank	Metrics used to rank models, e.g. Friedman ranking and mean rank of error rate.
Test	Statistical tests used to compare models, such as t-test, Wilcoxon test and Duncan's multiple ra...
Time	Metrics like training and prediction time.