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

Master Thesis

Author: Alexander Vladimir Merdian-Tarko (Student ID: 7325771)
Study program: M. Sc. Business Administration, Media and Technology Management
Supervisor: Univ.-Prof. Dr. Jörn Grahl

Department of Digital Transformation and Value Creation
Faculty of Management, Economics and Social Sciences
University of Cologne

May 02, 2019

Contents

List of Figures

List of Tables

1 Introduction

A growing number of increasingly complex models is available for different marketing use cases, but wide-spread application in practice appears to remain challenging. This is abetted by the lack of a comprehensive evaluation and comparison of proven and novel models that consider criteria that go beyond merely assessing predictive accuracy and instead make the perspective of marketing executives the center of attention. Therefore, this thesis is set to formally evaluate and compare a selection of machine learning and deep learning models for the prediction of customers' purchasing intentions in large-scale experiments, using real-world data from an e-commerce website. This task is accomplished by (1) identifying the most relevant models for predicting customers' purchasing intentions from an extensive meta-analysis of comparative machine learning research, (2) implementing and testing them in a large-scale real-world scenario and (3) applying a formal framework grounded on management and marketing science to evaluate and compare them in a comprehensive fashion.

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

In addition to the growing amount of data and pronounced importance of analytics in marketing, new models keep getting added to the marketing practitioner's toolbox over time (Wedel & Kannan, 2016, pp. 100-101). Machine learning, for example, can be used to predict conversions in e-commerce environments using clickstream data (e.g., Boroujerdi et al., 2014; Sarwar, Hasan, & Ignatov, 2015; Sakar, Polat, Katircioglu, & Kastro, 2018). Deep learning, a subcategory of machine learning, attracts growing attention having achieved several breakthroughs across different domains in the recent past. Notable examples are long-short term memory networks for sequence prediction (Hochreiter & Schmidhuber, 1997), convolutional neural networks for image recognition (Krizhevsky, Sutskever, & Hinton, 2012) and transformer encoders for language understanding (Devlin, Chang, Lee, & Toutanova, 2018). Especially recurrent neural networks like LSTM pose an interesting type of model for the prediction of conversions in an e-commerce setting. This is since they are sequence-based and clickstream data

essentially consist of sequences of user behavior over time (Lang & Rettenmeier, 2017, p. 1).

Evidently, the choice of available models is vast and their numbers growing and so the selection and application of the right model is not an easy endeavor for marketing executives. It is accompanied by numerous caveats, including the observations that managers tend to refuse to apply models they do not understand and good models are generally hard to find (Little, 1970, 2004), that it is a complex art to build models that are able to improve productivity (Lodish, 2001) and that 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 – despite the potential benefits such decision support systems might entail (Lilien, 2011).

Therefore, the objective of this thesis is to compare a selection of machine learning and deep learning models for customer journey prediction in e-commerce, considering the perspective of marketing executives who use models as support for their decision making. Thus, the research question of this thesis is:

How do different machine learning and deep learning models perform on the problem of predicting customers' purchasing intentions in terms of criteria that are relevant to marketing executives?

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

This thesis contributes to research by deriving general tendencies and insights from comparative research concerned with machine learning and deep learning from the past 30 years. This is accomplished through a comprehensive meta-analysis that considers several different dimensions, such as models, data and

evaluation metrics. The practical contribution of this thesis constitutes the conduction of large-scale experiments using twelve models, ten clickstream data samples and several evaluation metrics, followed by a thorough evaluation and comparison of the models along a multi-faceted framework based on management and marketing research. The framework is specifically designed to evaluate models with regard to criteria that are relevant to marketing executives, who are typically the recipients of such models and their output. Implications for marketing executives are derived as well.

The remainder of this thesis is structured as follows: Section 2 describes the methodology applied to conduct this thesis in more detail; Section 3 presents and analyzes related studies and motivates the choices of models, metrics and methods to be implemented in the experiments; Section 4 explains the model evaluation framework; Section 5 introduces the experimental setup, the data and the models in detail; Section 6 evaluates the models and experimental results considering the criteria of the previously introduced evaluation framework; Section 7 discusses the findings of the model evaluation and comparison and derives implications for marketing executives; Section 8 finally concludes with a summary and an outlook for future research.

2 Methodology

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

Model evaluation framework. Second, a model evaluation framework is introduced that builds on what management and marketing research found to be important criteria for models to be successfully used in practice by marketing executives (e.g., Little, 1970, 2004; Lodish, 2001; 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, pp. 7-10). Although Anderl et al.'s (2014) framework has been designed to evaluate attribution models, it can be applied to evaluate machine learning and deep learning models for customer journey prediction as well. This is possible because it is based on literature that deals with marketing models and criteria for their successful application in general. The objective of using this multi-level evaluation framework is to allow for a thorough comparison of different models beyond the calculation of single quantitative metrics but including qualitative criteria that are particularly relevant to marketing executives as well.

Large-scale experiments on real-world data. Third, twelve models, chosen as a result of the meta-analysis, are evaluated and compared on ten different samples of real-world clickstream data from a Swiss e-commerce website in terms of the previously introduced six criteria of the model evaluation framework. The

objective of this large-scale comparison is to enable a comprehensive evaluation of machine learning and deep learning models for customer journey prediction, considering criteria that are particularly relevant to marketing executives.

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

3.1 General Comparative Studies

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

Table 1 presents 15 comparative studies that meet the criteria above. The author(s) and the year of the study's publication are in the first column, the models and model families are in the second column, the number of data sets, the range of the number of instances, the type of the data sets and the classification problem at hand are in the third, fourth, fifth and sixth column, respectively[1]. The metrics and methods used to evaluate the models are in the seventh column.

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

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

[1]The total number of models used in a study is in parenthesis in the first column (same applies to Table 2). In the fifth column, R indicates real and S synthetic data. In the sixth column, B indicates binary and M multi-class classification.

International Conference on Data Mining).

The average number of models used per study is 24, but it ranges from three to 179. Decision trees (DT) are the most frequently used models with 14 occurrences. They are followed by k-nearest neighbors (KNN) and neural networks (NN) with nine each, naïve Bayes (NB) with eight and logistic regression (LR) with seven occurrences.

A similar observation is made for the number of data sets used per study with an average of 32 but ranging from four to 165. Typically, the number of instances per data set is rather low and only few large-scale data sets are used. Most studies explicitly justify their selection and explain the characteristics of the data they used, but some studies provide only limited information on the characteristics of the data they used for no specific reason (e.g., Provost & Domingos, 2003; Khan, Arif, Siddique, & Oishe, 2018). Bauer and Kohavi (1999, p. 113) require data sets to contain at least 1,000 instances to make reliable assessments of models on sufficiently large test sets. Perlich, Provost, and Simonoff (2003, p. 214) use only data sets with at least 700 instances to be able to get learning curves of reasonable length. Just two of the considered studies use synthetic data in addition to real-world data (Michie, Spiegelhalter, & Taylor, 1994; Lim, Loh, & Shih, 2000). The analysis shows that there are several data sets that are used in multiple studies, typically taken from the University of California, Irvine Machine Learning Repository. Just three studies exclusively consider binary classification problems (Bradley, 1997; Perlich et al., 2003; Caruana & Niculescu-Mizil, 2006) while the remaining twelve consider multi-class problems as well.

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

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

are Lim et al. (2000), Caruana and Niculescu-Mizil (2006), Fernández-Delgado, Cernadas, Barro, and Amorim (2014) and Olson, La Cava, Mustahsan, Varik, and Moore (2018).

Table 1: Overview of general comparative studies

Studies	**Models**	**Data sets**	**Instances**	**Types**	**Classes**	**Evaluation metrics**
Weis and Kapouleas (1989)	DA, KNN, BM, NN, RL, DT (9)	4	106-3,772	R	B/M	Accuracy
Shavlik, Mooney, and Towell (1991)	DT, NN (3)	5	226-11,500	R	B/M	Accuracy, time
Michie et al. (1994)	DA, DT, RL, LR, KNN, BM, NB, NN (23)	22	270-58,000	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	270-58,000	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	1,000-58,000	R	B/M	Accuracy
Lim et al. (2000)	DT, RL, DA, LR, NN (33)	32	151-4,435	R/S	B/M	Accuracy, time, complexity
Huang, Lu, and Ling (2003)	NB, DT, SVM (4)	18	132-8,124	R	B/M	Accuracy
Perlich et al. (2003)	DT, LR, BAG (8)	36	700-1,000,000+	R	B	Accuracy, ranking
Provost and Domingos (2003)	DT, BAG (5)	25	unclear, but incl. large-scale datasets	R	B/M	Ranking, accuracy

Continued on next page

Table 1 – *Continued from previous page*

Studies	**Models**	**Data sets**	**Instances**	**Types**	**Classes**	**Evaluation metrics**
Tan and Gilbert (2003)	DT, RL, NB, KNN, SVM, NN, STC, BAG, BOOST (17)	4	106-1,484	R	B/M	Accuracy
Caruana and Niculescu-Mizil (2006)	SVM, NN, LR, NB, KNN, RF, DT, BAG, BOOST (30)	11	9,366-40,222	R	B	Accuracy, MISC-E
Fernández-Delgado et al. (2014)	DA, NB, BM, NN, SVM, DT, RL, BOOST, BAG, STC, RF, ENS, GLM, KNN, LR, REG, MISC-M (179)	121	10-130,064	R	B/M	Ranking, accuracy, tests, MISC-E
Khan et al. (2018)	KNN, SVM, ENN, LMNN (4)	11	∼200 to ∼5,000	R	B/M	Accuracy
Olson et al. (2018)	NB, LR, SVM, KNN, DT, RF, BOOST, MISC-M (13)	165	mostly <5,000 and incl. large-scale datasets	R	B/M	Ranking, accuracy, tests

In general, there is no single model that outperforms all other models, but model performance is highly dependent on the given problem and data set (Salzberg, 1999, p. 11). Several studies explicitly confirm this statement (e.g., Michie et al., 1994; King et al., 1995; Bradley, 1997; Huang et al., 2003; Caruana & Niculescu-Mizil, 2006; Olson et al., 2018). Although Lim et al. (2000) state that there are no statistically significant differences between many models they evaluate, they show that there are huge differences in training time and interpretability though. Some studies that compare only a few models derive more differentiated conclusions. (Bauer & Kohavi, 1999) find that bagging (BAG) generally outperforms boosting (BOOST) while both perform better compared to DT and NB – however, at the cost of interpretability since BAG and BOOST are more complex and less interpretable. Perlich et al. (2003) state that LR tends to perform better for smaller data sets while DT tends to perform better for larger data sets. In a similar nuanced fashion, Tan and Gilbert (2003) find that a 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 & 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. Fernández-Delgado et al. (2014) confirm these results to a certain extent, 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) as well who find BOOST and RF to perform well while NB performs poorly in general. They additionally recommend SVM, LR and an Extra Tree Classifier, but state that certain variants or modifications of these and other models are not competitive at all, such as KNN, some NN, some BOOST and some DT (Olson et al., 2018).

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

taking into account the marketing perspective and the application of customer journey prediction.

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

3.2 Comparative Studies Focused on Customer Journey Prediction

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

It is worth noting that other models apart from machine learning and deep learning are successfully used for purchase prediction on clickstream data. The following mentions a selection of such studies. Moe, Chipman, George, and McCulloch (2004) 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, Li, Srinivasan, and Liechty (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, Haupt, Gebert, and Lessmann (2018) use clickstream data to build graphs of visitor sessions and use corresponding graph metrics to show their importance for predicting purchase events.

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

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

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

Eight out of ten studies use a single data set while only two studies use two data sets (Lang & Rettenmeier, 2017; Sheil et al., 2018). From some studies it is difficult to deduce the exact amount of data they use, for example for business reasons (Lang & Rettenmeier, 2017, p. 3) or because it is not entirely clearly stated (Vieira, 2015). The amount of data used in the selected studies varies widely from as few as a couple thousand sessions (i.e. clearly defined visits to an online shop) over a timespan of several weeks or months (Moe & Fader, 2004; Suh, Lim, Hwang, & Kim, 2004; Boroujerdi et al., 2014; Toth, Tan, Di Fabbrizio, & Datta, 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, Tan, Duan, Liu, & Mong Goh, 2015; Lang & 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 & Fader, 2004; Sakar et al., 2018), dependent on the specific domain of the e-commerce website under consideration. In other cases, the reported conversion rate lies between two and six percent (e.g., Suh et al., 2004; Sarwar et al., 2015; Vieira, 2015).

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

Since the number of different evaluation metrics used in the studies in Table 2 is comparably small they are not summarized. While most studies use some form of accuracy metric (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. Sarwar et al. (2015) and Wu et al. (2015) use a custom metric specifically tailored to the RecSys Challenge 2015 (Ben-Shimon et al., 2015). Sarwar et al. (2015) are the only ones to report the time their models required for training. Only Sakar et al. (2018) explicitly report p-values of statistical significance tests they run to evaluate model performance.

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

Studies	**Models**	**Data**	**Targets**	**Evaluation metrics**
Moe and Fader (2004)	CM, LR, MISC (6)	11,000 sessions	Purchase probability of session	Log-likelihood, Bayesian information criterion, predicted conversion rate
Suh et al. (2004)	DT, NN, LR, ENS (4)	73,000 events	Purchase probability of session	Accuracy, misclassification error, lift
Boroujerdi et al. (2014)	DT, RF, LR, NN, SVM, RL, KNN (16)	60,000 sessions	Purchase probability of session	Precision, recall, F-score
Sarwar et al. (2015)	NB, RF, BM, LR, BOOST (5)	11,800,000 sessions	Purchase or no purchase session and item bought	Custom score from RecSys 2015 Challenge, precision, recall, training time
Vieira (2015)	LR, RF, DBN, SDA (5)	1,000,000 sessions	Purchase within next 24 hours of current session	AUC
Wu et al. (2015)	RNN/LSTM, NN, BOOST (3)	11,800,000 sessions	Purchase or no purchase session and item bought	Custom score from RecSys 2015 Challenge

Continued on next page

Table 2 – *Continued from previous page*

Studies	**Models**	**Data**	**Targets**	**Evaluation metrics**
Lang and Rettenmeier (2017)	LR, NN, RNN/LSTM (3)	several million sessions for two countries	Purchase within next 7 days of current session	Negative log-likelihood, AUC
Toth et al. (2017)	MM, RNN/LSTM (2)	199,000 sessions	Purchase, abandoned cart or browsing-only session	Precision, recall, F-score
Sakar et al. (2018)	DT, RF, SVM, NN (4)	12,000 sessions	Purchase or no purchase session	ACC, F-score, true-positive/true-negative rate, statistical significance
Sheil et al. (2018)	RNN/LSTM/GRU, BOOST (4)	9,200,000 sessions (dataset 1), 1,400,000 sessions (dataset 2)	Purchase or no purchase session	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 created from their other models outperforms all individual models. Wu et al. (2015) and Toth et al. (2017) explicitly explore RNN with long-short term memory (LSTM) for purchase prediction and find that they perform best in their experiments. Lang and Rettenmeier (2017) use a LSTM as well and conclude that LSTM reduce the need for extensive feature engineering, yield increased predictive performance and improve interpretability of predictions. Sheil et al. (2018) find that LSTM performs best for one data set while BOOST performs best for the other data set they use. Sarwar et al. (2015) find that BOOST outperforms all other models for the prediction of sessions that lead to a purchase. Vieira (2015) builds another specialized NN, namely a stacked denoising auto-encoder, that he reports performs best in his experiments. Finally, Sakar et al. (2018) find NN to yield better performance than RF and SVM.

Although one might believe to recognize certain patterns, it might be too rash to conclude that complex models, such as different variants of BOOST, NN or RNN, 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 and LSTM) given that most of them have been conducted in the period from 2014 to 2018.

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

As mentioned in Section 3.1, a variety of preprocessing and feature engineering and selection techniques along with techniques for measuring the effect of sample size is applied across the selected studies shown in Table 2 as well. For example, it is noteworthy that Lang and Rettenmeier (2017) claim that LSTM 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 of these techniques in more detail, which is done in Section 5.

4 Model Evaluation Framework

Anderl et al. condense research on the application and acceptance of marketing models into an evaluation framework they use to assess online attribution models. 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, for example increasing the number of conversions or revenue (Anderl et al., 2014, p. 7). **Objectivity** originates from Lilien's (2011, p. 198) claim for a model to allow for the computation of a variable's relative impact and the objective evaluation of available decision options. **Predictive accuracy** is defined as a model's ability to correctly predict conversions (Anderl et al., 2014, p. 8), picking up Lodish's (2001, p. 54) lesson of the importance of a model's credibility to persuade managers. **Robustness** is defined as a model's ability to deliver "(. . .) stable and reproducible results (. . .)" after multiple runs of the model (Anderl et al., 2014, p. 8), covering Little's (1970, 2004, p. 470, p. 1843) requirement for a model to return useful results. According to Little (1970, 2004, p. 470, pp. 1843-1844), models should be simple and easy to communicate with, which Anderl et al. (2014, p. 8) 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. **Versatility** incorporates Little's (1970, 2004, p. 470, pp. 1843-1844) requirements that models should be easy to control and to adapt, that is models should allow for the inclusion of novel information and data in rapidly and frequently changing environments through a high degree of flexibility (Anderl et al., 2014, p. 10). **Algorithmic efficiency** builds upon Lodish's (2001, p. 54) lesson that models should ideally deliver results on-demand, that is when managers need them, which is particularly important when dealing with large amounts of data (Anderl et al., 2014, p. 10).

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

incorporating not only quantitative metrics but also emphasizes the importance of criteria that capture dimensions that are relevant for marketing executives to actually apply models in practice.

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

5 Experiments

Section 5.1 presents the setup of the experiments and names the tools and software packages used to conduct the experiments. Section 5.2 provides detailed information on the data used in the experiments and the choices and techniques applied to transform the raw data into training and test sets. The target and features are examined in more detail as well. Section 5.3 analyzes the data in a descriptive manner to generate first insights. Section 5.4 finally introduces the models, stating and justifying important choices that are made regarding implementation, parameter choice, training and testing. The objective of Section 5 is to explain the experiments' setup and the reasoning behind the choices regarding models, target, features, training and test sets and methods applied to process the data.

5.1 Experimental Setup

A selection of twelve models is derived from the meta-analysis in 3: LR, DT, NB, KNN, RF, SVM, BOOST, NN with one (NN1), three (NN3) and five (NN5) hidden layers, respectively, RNN and LSTM. Varying the number of hidden layers in NN increases complexity and allows to observe whether predictive performance and other criteria change with complexity for this particular model. Using both RNN and LSTM allows to investigate whether LSTM is generally superior, given its more sophisticated recurrent layer architecture.

The studies presented in 3.2 use different targets for predicting conversions, all of which are justifiable and each having different merits and drawbacks. The extended window approach suggested by Vieira (2015) and Lang and Rettenmeier (2017) is selected, creating a target that captures conversions within the next 24 hours of a given visit. This approach builds on the assumption that multiple sessions within a specified time window can contribute to a conversion, that is assuming that customers tend to purchase more frequently after multiple sessions and seldomly after single, isolated sessions.

To measure the effect of sample size, inspired by Shavlik et al. (1991), Lim et al. (2000), Perlich et al. (2003), Moe and Fader (2004) and Vieira (2015), ten clickstream data samples of different sizes are used in the experiments, ranging from about 4,000 to 2,000,000 unique visitors per sample. Only few studies considered in 3.2 explicitly state how they sample their data or create their training and test sets. For example, Wu et al. (2015, p. 3) decide for a train test split ratio of three to one and add every fourth session to the test set. Lang and Rettenmeier (2017, p. 4) instead use the first three weeks in their data for training, the following week for validation and the subsequent two weeks for testing. For the

following experiments, unique visitors are randomly selected from the entire data set to ensure a stratified class distribution and a generally balanced distribution of attributes over the entire period represented in the data. The size of samples is increased by successively adding additional unique visitors to reach the desired sample size. Each sample is split in the fashion that the resulting training and test sets contain distinct unique visitors. Training and test sets are split with a ratio of four to one. The number of unique visitors is used to specify the size of a sample instead of the actual number of sessions (i.e. instances) in a sample. This is practical given that the number of unique visitors per sample is roughly proportional to the number of sessions per sample as shown in Table 4 in Section 5.3. Creating training and test sets in this manner allows to leverage the entire timespan of the data without cutting individual customer journeys (i.e. a visitor's entire sessions are contained either in the training or the test set but not split across both), thus capturing customer journeys and corresponding behavior in their entirety.

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

5.2 Data

The data used for this thesis stem from a Swiss e-commerce website, comprising 63 GB and spanning six months from May to October 2016. The data can be understood as sequences of events that capture visitors' clicking behavior and additionally contain visitor-level information, such as device type, operating system and the marketing channel via which the visitor came to the online shop. Each row in the raw data represents an event (e.g., page view, product view, addition of a product to the shopping cart, purchase etc.), tagged with a timestamp and additional information that are registered by the tracking software implemented on the website. A visitor or customer is an individual that visits an e-commerce website (e.g., to browse the online shop's product catalogue or to purchase a specific product). 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 a specific number 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 (Adobe Systems Incorporated, 2019, p. 321). The terms session and visit are used interchangeably. A purchase is also referred to as a conversion in the following. The conversion rate in a given period is defined as the ratio of the number of conversions and the number of sessions in that period.

Cleaning and mapping. There are 29.5 million rows in the raw data, ranging from 3.4 to 6.7 million rows per month. The number of rows in the raw data is reduced by about 1.8 percent to 29 million rows after cleaning the raw data (e.g., dropping rows with missing values or broken records). Missing values in the remaining rows are filled according to context (e.g., tagged as Not Specified if applicable). The amount of removed rows varies from 0.3 to 3.7 percent per month. There are 138 columns in the raw data of which 42 are considered to contain information that is useful for this thesis. After splitting the observations in certain columns, the number of columns increases to 48. Since certain columns are encoded, mappings of codes and strings that represent the actual information are done using special mapping files. Some columns are already casted to the right data type and format in this processing step as well. The number of unique visitors is reduced from 4.7 to 4.5 million in this processing step.

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

Preparing target and features. Sessions containing only a single event (i.e. bounce sessions) are removed to reduce noise and because more engaged visitors are more interesting from a marketing perspective, assuming that more active visitors have a higher propensity to purchase. Removing bounce sessions reduces the number of sessions from 6.6 to 3.3 million and the number of unique

visitors from 4.5 to 2.5 million. This is a similar but less drastic step compared to Lang and Rettenmeier (2017, p. 4) who only consider customers with a least 15 previous actions and (Toth et al., 2017, p. 2) who include only sessions with at least five events.

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

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

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

Feature selection. Two measures are taken to reduce the large number of features created in the previous steps and to filter out the most relevant ones. First, features that are too closely correlated with the target are removed (e.g., a feature that indicates whether the customer reached the checkout step during a session). Second, although certain models incorporate measures to assess the importance of individual features (e.g., coefficients in LR and SVM or feature importances in tree-based models), relying on those is not practical since they are model-specific. A model-agnostic measure of feature importance is more desirable because many different models are used in the experiments and all should use the same features (with an exception for RNN and LSTM to be explained later). To avoid introducing bias in the modeling stage, feature selection is done on an independent sample that is neither used for training nor for testing any model. The sample used for feature selection is fairly large, containing roughly 450,000 unique visitors. Its conversion rate is similar to those of the other samples which ensures that its characteristics are similar to those of the other samples as well. After standardizing numerical features by removing the mean and scaling to unit variance, analysis of variance (ANOVA) F-values and according p-values are computed for the target and features in the aforementioned sample. To avoid setting an arbitrary threshold to select the k best features, only those features are selected whose F-value was significant at the one percent significance level, indicated by the according p-value. In other words, the F-test's null hypothesis of a given feature not contributing to the prediction of the target is rejected at the one percent level for 134 features. These 134 features are used for modeling. Table 9 in the Appendix lists all features and their corresponding F- and p-values[2].

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

Preprocessing. The preprocessing steps described in the paragraph above are applied to all other samples as well. In summary, the numerical features

[2]More in depth explanations of the features are provided in the corresponding tracking software's reference document (Adobe Systems Incorporated, 2019).

are standardized by removing the mean and scaling to unit variance and all but those 134 features identified through feature selection are removed from training and test sets. Table 3 summarizes the number of rows, columns (including two identifiers, the target and 134 actual features) and unique visitors after each of the processing stages explained above.

Table 3: Descriptive statistics of data processing stages

Stage	**Rows**	**Columns**	**Unique visitors**
Raw	29,495,770	42	4,717,042
Cleaning and mapping	28,950,434	48	4,450,709
Aggregation	6,552,128	50	4,450,709
Preparing target and features	3,296,711	169	2,453,298
Feature selection/preprocessing	3,296,711	137	2,453,298

5.3 Descriptive Statistics

Table 4 presents descriptive statistics for the ten clickstream data samples and the entire data set. The evenly spread figures result from the random sampling of unique visitors and indicate that the samples' attributes are balanced. The number of sessions per sample is roughly proportional to the number of unique visitors per sample – the ratio is about three to four. Across all samples, 18 percent of visitors have two or more while only about two percent have five or more sessions in the period under consideration. The average number of sessions per visitor is 1.3 to 1.4, the according median is 1 and the standard deviation is about 1.8 to 3.8 sessions per visitor. The share of buyers among all visitors is about 3.3 to 3.5 percent. About 20 percent of conversions are repeated conversions. The conversion rate across all samples and the entire data set ranges from 2.9 to 3.2 percent. Overall, the smaller samples tend to be slightly less balanced. The low conversion rate indicates a severe class imbalance which could prove challenging in the experiments. Balancing the class ratio using sampling techniques, such as SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), tend to be computationally expensive, especially with large amounts of data. Exploring their usefulness and the extent to which they are able improve predictive performance could be investigated in future research.

Table 4: Descriptive statistics of all samples and the entire data set

Samples	Unique visitors	Sessions	Visitors with >= 2 sessions	Visitors with >= 5 sessions	Mean # sessions	Median # sessions	Standard deviation # sessions	Buyers	Conversions	Conversion rate
Sample 1	3,906	5,591	698	65	1.4314	1	3.7541	131	161	0.0288
Sample 2	7,812	10,736	1,430	117	1.3743	1	2.7333	260	319	0.0297
Sample 3	15,625	21,480	2,839	258	1.3747	1	3.0801	523	642	0.0299
Sample 4	31,250	42,836	5,695	541	1.3708	1	2.6352	1,054	1,301	0.0304
Sample 5	62,500	84,757	11,502	1,057	1.3561	1	2.0870	2,079	2,566	0.0303
Sample 6	125,000	168,202	22,860	2,098	1.3456	1	1.8120	4,274	5,247	0.0312
Sample 7	250,000	336,018	45,602	4,145	1.3441	1	1.9582	8,466	10,443	0.0311
Sample 8	500,000	671,664	91,292	8,245	1.3433	1	1.8721	17,110	21,238	0.0316
Sample 9	1,000,000	1,342,635	183,532	16,389	1.3426	1	2.1338	34,582	42,906	0.0320
Sample 10	2,000,000	2,687,600	367,095	32,780	1.3438	1	2.0465	68,985	85,619	0.0319
Full sample	2,453,174	3,296,576	449,864	40,209	1.3438	1	2.0526	84,510	104,831	0.0318

Figure 1 shows the development of activity on the e-commerce website over time, represented by visits, purchases, page views and product views. The numbers are aggregated by week and are based on the data from the fourth stage in Table 3 in Section 5.2. 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 incomplete. These two weeks consequently contain less data points, resulting in lower numbers for visits, purchases, page views and product views. Visits, page views and product views first decrease, creating 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. The occasional peaks and the increase of activity toward the autumn weeks may be caused by temporal marketing campaigns or seasonality effects (e.g., holiday season during the summer weeks), respectively.

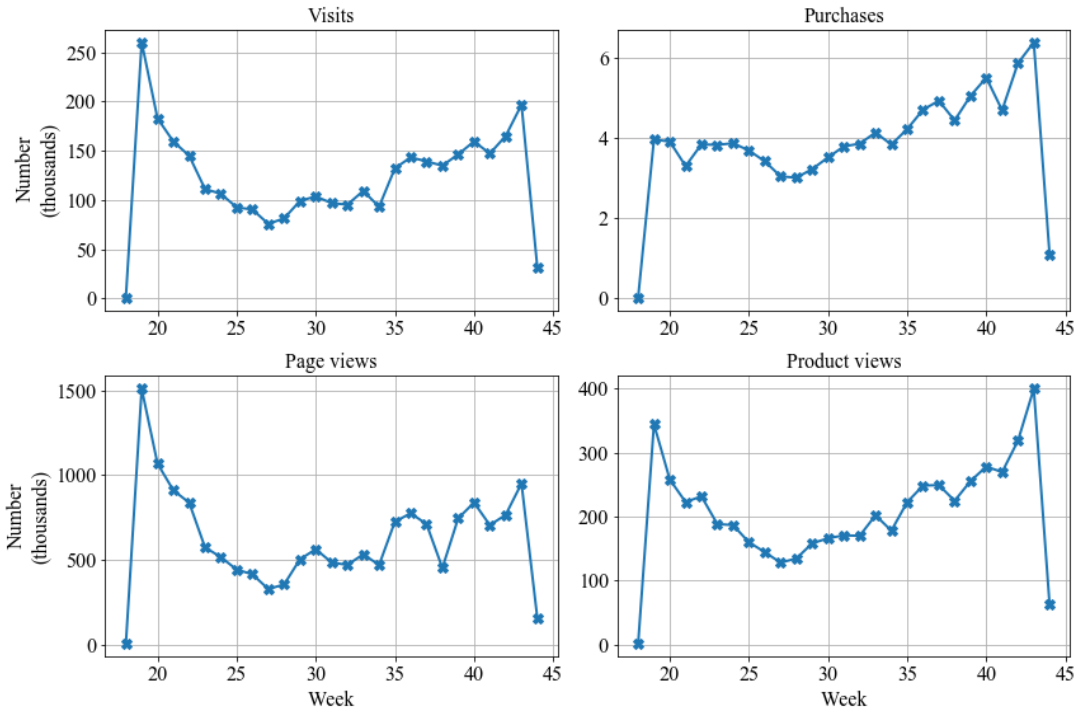

Figure 1: Visits, purchases, page views and product views aggregated by week

5.4 Models

Default hyperparameter settings are chosen for most models to maintain a certain degree of comparability. One could argue, however, that some models' default hyperparameters make them unreasonably superior over other models' defaults, leading to 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 is beyond the scope of this thesis. Tuning the hyperparameters of all models to improve the significance of their comparison could be a task worth taking on in additional experiments. For now, Scikit-learn's default hyperparameter settings are used for LR, DT, NB, KNN, RF, SVM and BOOST. For the (recurrent) neural networks Keras' default hyperparameters are used if no hyperparameter must be specifically specified, which is not always the case though. Detailed information on the respective hyperparameters, implementational details and references for further study are found in the respective Scikit-learn and Keras documentation pages (Pedregosa et al., 2011; Chollet and others, 2015).

Scikit-learn models. 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 chooses the best split at each node, an arbitrary maximum tree depth, a minimum of two instances per split, a minimum of one instance per leaf, equal weighting of instances, an unlimited number of leaf nodes and considering all available features. NB is a **Gaussian naïve Bayes classifier** without prior probabilities of the classes. KNN is a ***k*-nearest neighbors voting classifier** with the number of neighbors being equal to five, uniform weighting of all neighbors in a neighborhood, automatic selection of the appropriate algorithm to compute the nearest neighbors and the distance metric being Euclidean as specified by the Minkowski metric's power parameter p being equal to two. RF is a **random forest classifier** with a forest of ten trees, a *gini* function to measure the quality of a split, an arbitrary maximum tree depth, a minimum of two instances per split, a minimum of one instance per leaf, equal weighting of instances, an unlimited number of leaf nodes and considering all available features. RF is an ensemble that fits several decision trees on various sub-samples of the training set and uses averaging to improve predictive accuracy and to mitigate overfitting. The sub-sample size is equal to the size of the training set, but bootstrap sub-samples are drawn with replacement. SVM is a **linear support vector classifier** with *l2* penalty, squared hinge loss and the regularization parameter C being equal to one. These settings tend to scale better to large amounts of data than other SVM implementation with *rbf* kernels for example. BOOST is a **gradient boosting classifier** with deviance loss function, a learning rate of 0.1 controlling the contribution of each tree, 100 boosting stages, the entire training set being used for fitting the base learners, mean squared error with improvement score by Friedman to measure the quality of a split, a minimum of two instances per split, a minimum of one

instance per leaf, equal weighting of instances, a maximum tree depth of three nodes, an unlimited number of leaf nodes and considering all available features. BOOST builds an additive model by fitting a single regression tree on the negative gradient of the binomial deviance loss function in each of the 100 boosting stages.

Keras neural network models. Building neural networks using Keras requires more decision making regarding model architecture and hyperparameter settings. Besides, for the sake of comparability, the (recurrent) neural networks are not trained using GPUs, which would have probably increased their training speed, but instead CPUs are used for training all models[3]. The neural networks with one, three and five hidden layers, respectively, are built in an analogous manner so that the following explanations in this paragraph apply to all three. The input layer and all hidden layers are fully connected and consist of as many neurons as there are features in the training set (i.e. 134) and use a Rectified Linear Unit (ReLU) activation function. The output layer consists of one neuron and uses a Sigmoid activation function for binary classification. The default Xavier uniform initializer (Glorot & Bengio, 2010) is used to initialize the layers' weight matrices. Binary cross-entropy loss is minimized using the Adam optimizer. The choices of activation functions, weight initialization and the optimizer follow Lang and Rettenmeier's (2017, p. 4) choices for their NN and LSTM. The number of epochs during training (i.e. the number of iterations over the entire training set) is set to ten, following Toth et al. (2017, p. 3). The batch size during training (i.e. the number of instances per gradient update) is set to 256 instances, which is eight times the default batch size of 32. To speed up training if possible, early stopping is configured so that training is ended early if validation loss is not minimized in two subsequent epochs. The choices of batch size and early stopping follow Sheil et al. (2018, pp. 5-6). Using random dropout, 20 percent of units per layer are reset to zero during training to prevent overfitting. Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014) suggest a dropout rate of 50 percent for large neural networks, but since the neural networks considered here are comparably small a substantially lower dropout rate is chosen instead.

Keras recurrent neural network models. The previously discussed models are vector-based in the sense that they require feature vectors to be of fixed length (Olivier, Eren, & Rosales, 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

[3]However, a binary customized to the workstation's CPUs enabling AVX2 FMA support is used to speed up training, which might slightly favor (recurrent) neural networks though.

models (Lang & Rettenmeier, 2017, p. 1). RNN instead are able to directly "(...) operate on sequences of varying length and therefore (...)" present a natural fit for purchase prediction in e-commerce (Lang & Rettenmeier, 2017, p. 1). RNN can be extended by more powerful computational LSTM cells that have been originally developed by (Hochreiter & Schmidhuber, 1997) to address the problem of vanishing gradients and to make the storing of long-term information in RNN architectures possible. Wu et al. (2015), Lang and Rettenmeier (2017), Toth et al. (2017) and (Sheil et al., 2018) use such LSTM cells in their RNN. Both RNN and LSTM are explored in the experiments to investigate the potential superiority of LSTM over traditional RNN, for example in terms of the importance of storing long-term information in customer journey prediction. The RNN and LSTM implemented in the experiments are built in analogous manners with the only difference being the type of computational cell used in the recurrent layer. RNN and LSTM tend to be more computationally expensive than other models due to their rather complex architectures which is why both RNN and LSTM are built using only one recurrent layer, following Lang and Rettenmeier (2017, p. 4) and Toth et al. (2017, p. 3). Each model's recurrent layer consists of 256 RNN and LSTM cells, respectively, since Sheil et al. (2018, p. 6) find 256 cells per recurrent layer to be optimal in their experiments. This choice appears plausible because their experiments are conceptually similar to this thesis' 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 more 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 & Rettenmeier, 2017, pp. 1-2). Therefore, features that are explicitly designed to capture past customer behavior (e.g., purchases and sessions in the past hours and days and page and product views in the previous sessions) are excluded from the training and test sets used for RNN and LSTM, reducing the number of features from 134 to 115. This step allows to investigate their alleged superiority over vector-based models. Second, since RNN and LSTM operate on sequences of sessions, their training and test sets must be transformed from being two-dimensional to being three-dimensional instead. Vector-based models use two-dimensional input data where an instance represents 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 a sequence of 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 Experimental Results

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

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), for example a conversion. LR and SVM satisfy the objectivity criterion because they enable the computation of coefficients that indicate a feature's relative impact and whether the impact is positive or negative. The objectivity criterion is satisfied by DT, RF and BOOST as well since these models are able to return feature importances, that is 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 models are features that capture the time passed since the last purchase, cart-related events, product and page views and visitor-specific features like gender and age. Tables 10 to 14 in the Appendix show the top ten features most indicative of a conversion of LR, SVM, DT, RF and BOOST and the corresponding coefficients and feature importances, respectively.

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

(Pedregosa et al., 2011).

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

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, 2001; Anderl et al., 2014). Learning curves are not only a way to investigate the relationship of sample size and predictive performance but also allow for the comparison of models, which is relevant because models tend to perform differently depending on the amount of training data they are fed (Shavlik et al., 1991; Perlich et al., 2003). Computing learning curves using cross-validation would probably deliver more robust results including standard deviations but is computationally more expensive and time-consuming, especially for large sample sizes, which is why the following learning curve analysis refrains from it.

Class imbalance. Accuracy is close to 100 percent for all models across all samples but is not a suitable measure of predictive accuracy due to the highly imbalanced classes in the data. This leads to a bias towards the majority class (i.e. sessions that do not lead conversions) and emphasizes the inability of accuracy to distinguish between correctly classified predictions of different classes (Sokolova, Japkowicz, & Szpakowicz, 2006, p. 1016). Alternative performance metrics that are able to better account for class imbalance are AUC, precision, recall and F-score because they do not rely on a single performance indicator but are computed

using several components indicative of predictive accuracy. AUC is defined as the area under the Receiver Operating Characteristic (ROC) curve, which is created by plotting the values of the true positive rate on the y-axis and the false positive rate on the x-axis, taken from the confusion matrix created from a classifier's class predictions (Bradley, 1997, pp. 1145-1147). Precision answers the question of what proportion of predicted conversions was actually correct and is defined as the quotient of true positives, that is correctly predicted conversions, and the sum of true positives and false positives, that is all predicted conversions. Recall (also sensitivity) answers the question of what proportion of actual conversions was identified correctly and is defined as the quotient of true positives and the sum of true positives and false negatives, that is all actual conversions. The F-score (also *F1*-score or F-measure) computes the harmonic mean from precision and recall and is defined as two times the product of precision and recall divided by their sum (Sokolova et al., 2006, p. 1016).

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

F-score learning curves. To derive more generalizable conclusions, Figure 3 additionally shows the relationship of F-score and the number of unique visitors in the training set for each model. Test set F-score is on the y-axis and the number

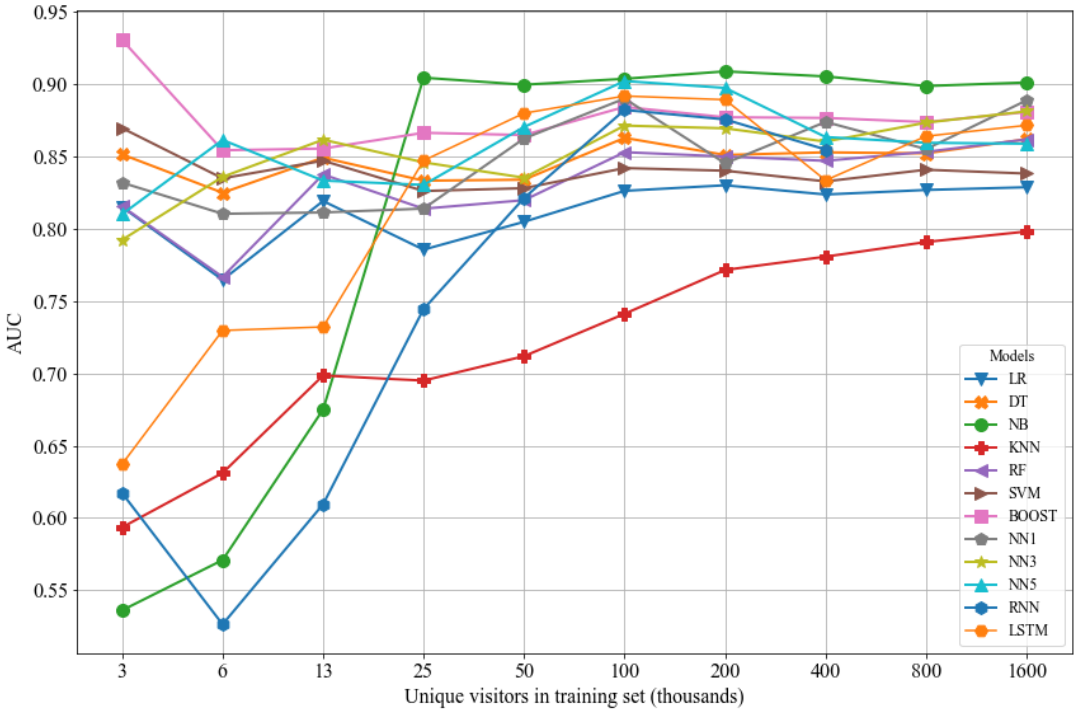

Figure 2: AUC learning curves of all models and samples

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

Both Figures 2 and 3 do not report AUC and F-scores for RNN for training sets with 800,000 and 1,600,000 unique visitors, respectively. This is because validation loss during training turned indefinite for these training sets, which is an indication of vanishing gradients. This is a common issue of RNN that is addressed by Hochreiter and Schmidhuber (1997) who develop LSTM cells that

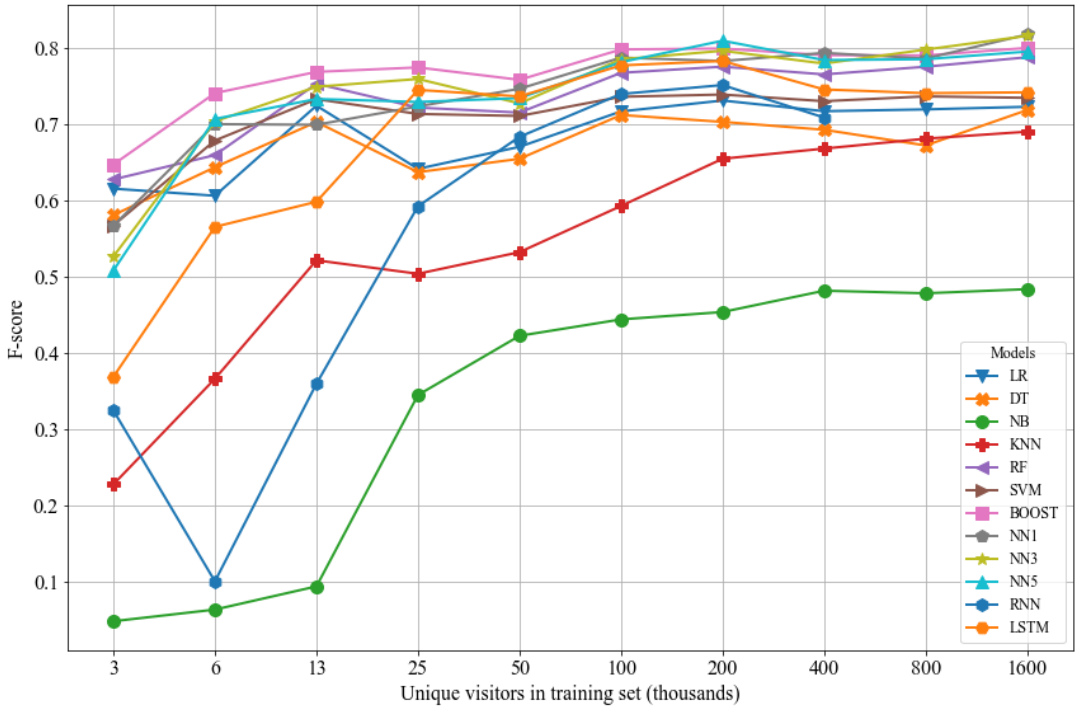

Figure 3: F-score learning curves of all models and samples

mitigate this and other issues of RNN. As a result, LSTM does not suffer from a similar problem so that AUC and F-scores can be computed for LSTM for all samples.

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

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

Table 5 shows the average ranks of all models computed using their AUC and F-scores for all samples, respectively, and the absolute differences between each model's average ranks for AUC and F-score. BOOST has the best average rank for AUC and F-score and a low absolute difference between its ranks. NN1, NN3, NN5 and LSTM have good average ranks for both AUC and F-score and fairly small absolute differences between their respective average ranks. The average ranks of LR, DT, KNN, RF, SVM and RNN are either worse or vary depending on the choice of performance metrics, indicated by the absolute difference between

the models' respective average ranks. The absolute difference between ranks is most extreme for NB. This is an indication that the evaluation and comparison of models tends to depend, among other factors, on the evaluation metric of choice.

Table 5: Average ranks of all models based on AUC and F-score

Models	Average rank (AUC)	Average rank (F-score)	Absolute difference
BOOST	2.8	1.5	1.3
DT	5.9	8.6	2.7
KNN	11.3	10.5	0.8
LR	9.5	7.6	1.9
LSTM	5.6	6.0	0.4
NB	4.1	11.8	7.7
NN1	5.5	3.7	1.8
NN3	4.6	3.3	1.3
NN5	4.4	3.8	0.6
RF	7.5	4.9	2.6
RNN	8.8	9.1	0.3
SVM	7.3	6.6	0.7

The Friedman test can be used in this case because multiple models are to be compared across multiple samples and because only the average performance is of interest and not its variation, which would be problematic since the samples are not independent but is not the case here (Demšar, 2006, p. 5). According to the Friedman test[4], the models' respective ranks are significantly different from from the average AUC rank of 6.44 and the average F-score rank of 6.45[5]. Therefore, the null hypothesis that the models are equal can be rejected at the five percent significance level. Since the null hypothesis of the Friedman test can be rejected, the Nemenyi test can be used for a pairwise comparison of models' performance (Nemenyi, 1963; Demšar, 2006, pp. 11-12). A model outperforms another model at the five percent level if the difference between their average ranks is at least equal to the critical distance of 5.27[6][7].

For example, BOOST is significantly better than KNN, LR and RNN in terms of the average rank based on AUC and significantly better than DT, KNN, LR, NB and RNN in terms the average rank based on F-score. LSTM is only significantly

[4]Considering twelve models and ten samples results in 11 and 9 degrees of freedom for the Friedman test.

[5]F-value for AUC ranking: 6.42; F-value for F-score ranking: 22.28; critical F-value: 1.89.

[6]Based on the number of models and samples under consideration and the resulting critical value of 3.27 for the Nemenyi test.

[7]The equations to calculate the Friedman and Neymeni tests and how to retrieve the tests' respective critical values can be found in Section 3.2.2 of (Demšar, 2006).

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

6.3 Robustness

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

In general, variation in precision, recall and consequently F-score tends to be higher than variation in accuracy and AUC. Standard deviations in accuracy and AUC typically range between less than 0.01 and up to 0.07 for the two small samples and between less than 0.01 and up to 0.02 for the two medium-sized samples. Standard deviations in precision, recall and F-sore typically vary between 0.03 up to 0.1 for the two small samples and between 0.01 and typically up to 0.06. There are a several outliers (narrowly defined here as cases where standard deviation exceeds 0.1), though, predominantly in the two smaller samples and across several models. But overall, models' robustness tends to improve with increasing sample size, indicated by typically lower standard deviations and thus less numerous outliers. The sole exception constitutes RNN with outliers for different metrics in all samples but the one containing 125,000 unique visitors. LR, DT, NB, BOOST and NN1 are the only models that do not report any standard deviations exceeding 0.1 across all metrics and samples. The reason why

for NN1 no outliers but for NN3 and NN5 several outliers are reported may be due to the more complex architecture of the latter two models, increasing their tendency to overfit individual folds. Tables 25 to 28 in the Appendix report cross-validation accuracy, AUC, precision, recall and F-scores with standard deviations in parenthesis for the four aforementioned samples.

6.4 Interpretability

Models and their results should be simple and easy to communicate to foster acceptance and application by marketing executives, which is captured by the criterion of interpretability (Little, 1970, 2004; Anderl et al., 2014). Guidotti et al. (2018) provide a much more comprehensive overview of the research efforts in the field of interpretability in machine learning for the interested reader. In recent years, several studies have been published, stressing and debating the importance and the notions of interpretability and explicability of machine learning models. Despite the substantial amount of research that has been recently conducted in this field and the general consensus that these are important concepts to foster adoption of machine learning applications, a unified and holistic view of what these concepts actually imply and how they can be satisfied and evaluated appears to be hard to agree upon (Doshi-Velez & Kim, 2017; Lipton, 2016). There are single studies that compare different machine learning models in terms of complexity and monotonicity, including models such as decision trees, nearest neighbors and Bayesian networks (Freitas, 2014). But there is also a range of specialized methods developed to explicitly allow for more transparency of machine learning models and their outputs. For example, Ribeiro, Singh, and Guestrin (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 a technique able to explain the predictions of any classifier by learning an interpretable model around a prediction in a local environment around an observation. 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, Taly, and Yan (2017) identify the fundamental axioms of sensitivity and implementation variance which they use to design an attribution method called Integrated gradients that is supposed to enable the attribution of a deep neural network's predictions to its input features and thereby make it more understandable for its users. Considering the current state of research in the field of interpretability of machine learning models with all its available methods to make these models

more transparent would exceed the scope of this thesis. Explicitly investigating the aspects of interpretability and explicability in the comparison of machine learning and deep learning models for a specific use case, in addition to testing some of the suggested methods above in practice, could be an interesting task for future research.

In addition to model complexity, Freitas (2014) suggests monotonicity constraints as a possible means to evaluate a model's degree of interpretability, but the following evaluation stays close to Anderl et al.'s (2014) marketing-centric reading of model interpretability. Therefore, the following evaluation is mainly focused on model complexity and whether a model is able to explain its predictions by assigning meaning and weights to the features used for predicting conversions[8]. 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 and typically contains only a subset of all available features, reducing complexity (depending on the tree size). The hierarchical structure of trees and the possibility to compute the relative importance of features provide insight into which attributes of the data are the most relevant for the model's prediction as well (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 in Section 6.1, other implementations of NB are able to provide such information), making it generally more difficult to comprehend its predictions. For KNN, according to (Freitas, 2014, p. 4), the feature values of the nearest training instances are usually different for every new test instance to be classified and in data sets with many features even neighboring instances can differ substantially, making KNN less explainable overall. Using prototypes (i.e. instances that represent typical data points) instead of the entire training data and computing attribute weights that are proportional to their predictive power are two approaches to improve the explicability of KNN's predictions, but come with additional effort (Freitas, 2014, p. 4). Ensembles like RF and BOOST consist of multiple decision trees that are relatively easy to understand individually, but the ways in which these ensembles combine individual trees and their predictions make them less intuitive. The visualization of individual trees from within these ensembles is possible with the implementations of RF and BOOST used in the experiments above but allows only limited insight because these ensembles consist not of a single but of numerous trees. The calculation of the importance

[8]The aspect of model optimization could be related to the notion of interpretability in a more in-depth assessment as well. This is since the number of hyperparameters to be tuned and complexity of the underlying optimization problem tends to widely differ from model to model.

of individual features for the ensembles' predictions additionally helps making RF and BOOST more comprehensible. SVM is a linear model because it produces a two-dimensional hyperplane in binary classification problems and the computation of coefficients and their signs provides information regarding which features the model considered important for its prediction. Although NN1, NN3, NN5, RNN and LSTM are generally shallow rather than deep in terms of the number of hidden layers, they are the most complex models under consideration, given the multitude of computational units in the input and hidden layers and the architectural choices made when building these models. Besides, there is no straight forward way to compute the impact individual features have on a (recurrent) neural network's predictions. There is the, however not straight forward, possibility to apply specialized methods to try to establish a relationship between a neural network's predictions and its input features (e.g., Sundararajan et al., 2017). Another possible approach are visualizations of conversion probabilities in individual sessions (e.g., Lang & Rettenmeier, 2017).

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

6.5 Versatility

The criterion of versatility requires models to be easy to control and adaptive when conditions change over time, particularly in fast-paced environments like e-commerce, for example when new data or features become available (Little, 1970, 2004; Anderl et al., 2014). All models considered in the experiments above are versatile in the sense that they are easily adapted if new data or features become available. After processing new data and features in the same or a similar manner as explained in Section 5.2, models can be simply retrained using training and test sets extended by fresh data and features. The time models require for being retrained on new data, however, varies substantially, as shown in Section 6.6. If, in addition, the data's level of aggregation or granularity were to be changed or a different target were to be used, models' flexibility depends on the degree to which these changes impact the underlying structure of the data and

prediction problem. For example, if, instead of modeling purchase prediction as a binary classification problem, conversion probabilities were to be predicted (which would be a natural extension of the experiments above), some models and their predictions would need to be substantially adapted in case they can only poorly or not at all predict probabilities in a straight forward fashion, for example NB and SVM (Caruana & Niculescu-Mizil, 2006, p. 163). Fundamental changes in the data structure or the target to predict, however, tend to be potentially less frequent than updated data or new features becoming available. Thus, the considered models tend to satisfy the criterion of versatility to a certain extent only although simple models could tendentially be more versatile than complex models. A generalized assessment, however, remains difficult to derive as the degree of a model's versatility highly depends on the exact changes inflicted on data, features and the prediction problem.

6.6 Algorithmic Efficiency

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

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

Testing times. Figure 5 shows the times the models required for testing for different sizes of the test set. Most models require substantially less time for testing than for training, which seems legit given the train test split ratio of four to one. Interestingly, however, the experiments show that KNN appears to be the only model that requires more time for testing than for training. In comparison

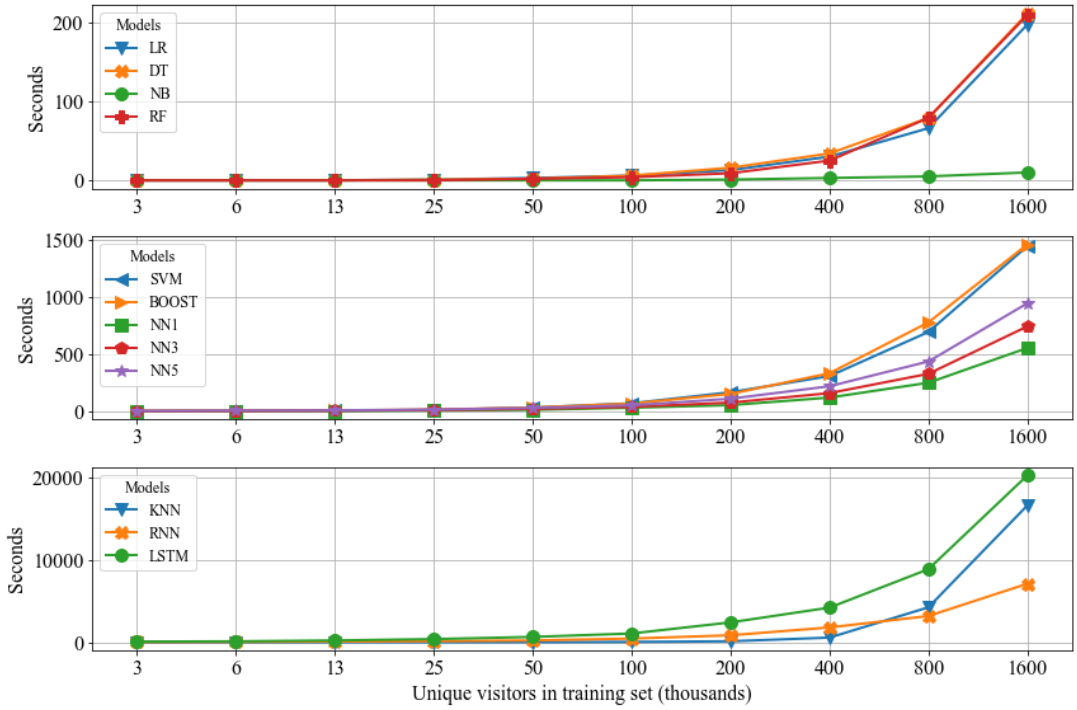

Figure 4: Training times of all models and samples

to the other models, testing takes so much longer for KNN that its corresponding curve is plotted in a separate graph due to the substantial differences in scale. This phenomenon could be ascribed to the characteristics of KNN's specific algorithm structure, that is no general internal model is learned during training, but examples of the training set are stored and then used for a simple majority vote of the nearest neighbors for predictions on the test set (Pedregosa et al., 2011). Overall, all models, except for KNN and also RNN and LSTM in parts, 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 specific use case and business environment (Anderl et al., 2014, p. 22).

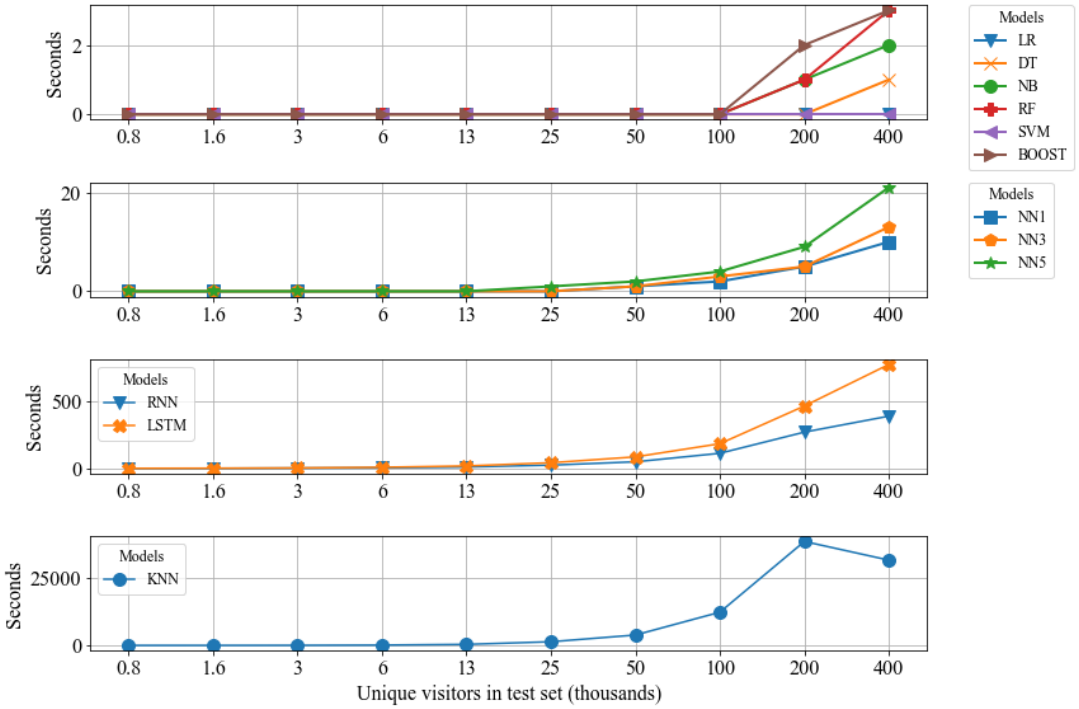

Figure 5: Testing times of all models and samples

7 Discussion and Managerial Implications

Model evaluation and comparison. Table 6 summarizes the evaluation and comparison of the models in terms of the model evaluation framework's six criteria. A check mark indicates that a criterion is satisfied while a checkmark in parenthesis indicates that a criterion is only partly satisfied. A cross mark implies that a model does not sufficiently satisfy a criterion. This summary is not meant to issue an ultimate general judgement on the models under consideration but rather intends to relate them regarding this thesis' particular use case and to illustrate specific tendencies that become apparent in the experiments.

Table 6: Summary of model evaluation and comparison

Models	**OBJ**	**PA**	**ROB**	**INT**	**VER**	**AE** [9]
LR	✓	(✓)	✓	✓	(✓)	✓
DT	✓	✓	✓	✓	(✓)	✓
NB	(✓)	✗	✓	(✓)	(✓)	✓
KNN	(✓)	✗	(✓)	(✓)	(✓)	✗
RF	✓	✓	(✓)	(✓)	(✓)	✓
SVM	✓	(✓)	(✓)	✓	(✓)	(✓)
BOOST	✓	✓	✓	(✓)	(✓)	(✓)
NN1	(✓)	✓	✓	✗	(✓)	(✓)
NN3	(✓)	✓	(✓)	✗	(✓)	(✓)
NN5	(✓)	✓	(✓)	✗	(✓)	(✓)
RNN	(✓)	(✓)	✗	✗	(✓)	✗
LSTM	(✓)	✓	(✓)	✗	(✓)	✗

LR, DT, RF, SVM and BOOST satisfy the criterion of objectivity since they allow for the computation of a feature's relative impact on a prediction. NB, KNN and the (recurrent) neural networks, however, do not provide such insights – at least not without significant additional effort or modifications.

NB's extremely variable average rankings for AUC and F-score make apparent that using a single metric to evaluate a model's predictive performance may not suffice to draw valid conclusions. KNN overall fails to achieve a similar level of predictive performance compared to the other models. Models that are accurate and stable across multiple metrics are NN1, NN3, NN5 and LSTM. Models such as LR, DT, RF, SVM and RNN perform well but are either comparably less accurate or less stable. BOOST stands out as it not only achieves the highest average ranks, but it even significantly outperforms LR, KNN and RNN in terms

[9] OBJ: Objectivity; PA: Predictive accuracy; ROB: Robustness; INT: Interpretability; VER: Versatility; AE: Algorithmic efficiency.

of the average rank based on AUC and LR, DT, NB, KNN and RNN in terms the average rank based on F-score.

Based on the cross-validation results, the most robust models are LR, DT, NB, BOOST and NN1. Less robust models are KNN, RF, SVM, NN3 and NN5 while RNN is clearly the least robust of the considered models. A possible explanation might be that some of the models are prone to overfit on certain folds during cross-validation.

The most interpretable models tend to be LR, SVM and DT. This is since the first two are linear in nature and output the sign and magnitude of coefficients that indicate the relevance of individual features. The latter allows for the computation of feature importances and the creation of graphical representations of the tree's structure. Since RF and BOOST are ensembles, their respective model structure is more complex, making them less easy to comprehend. But they are able to output feature importances and visualizations of individual trees, fostering interpretability at least to some extent. NB and KNN can be modified to be more interpretable, but their respective versions used in the experiments above are not very explainable per se. Due to the number of computational units and hidden layers as well as the resulting numerous network-internal computations, NN1, NN3 and NN5 tend to be hardly interpretable, even less so RNN and LSTM as they entail even more complex architectures.

The degree of a model's ability to adapt to changing conditions is difficult to assess without knowing the exact nature of the given changes. Thus, all models receive a check mark in parenthesis for the criterion of versatility in Table 6. This accounts for the tendency that they all tend to be fairly easily updated if updated data or new features become available. But assessing the consequences of more fundamental (however potentially less frequent) changes, such as changes in the data's level of granularity or a novel target to predict, poses a more serious challenge.

As models should not only be generally able to adapt to changes, they should be quick in doing so as well. The fastest models are LR, DT, NB and RF, which is not surprising given that they are among the least complex of the models considered. But more complex models, such as SVM, BOOST, NN1, NN3, and NN5, are still able to achieve somewhat reasonable run times. KNN, RNN and LSTM, however, tend to require much more time for training and testing, which harms their abilities to be tested for robustness and to be updated when changes in the business environment or data occur.

Three more interesting observations are noteworthy. First, the differences in the number of hidden layers in NN1, NN3 and NN5 appear to have no significant effect on neither predictive accuracy nor a substantial effect on training and

testing times. The other criteria except for robustness remain largely unaffected as well. NN3 and NN5 seem to be somewhat less robust than NN1 as indicated by larger standard deviations in cross-validation for NN3 and NN5. This might be due to their larger number of hidden layers that makes them potentially more prone to overfitting. Admittedly, all three neural networks are quite shallow, which might be the reason why no fundamental differences seem to be distinguishable. Investigating the differences between neural networks with more hidden layers could therefore produce additional insights. Second, LSTM indeed seems to be more powerful than RNN. LSTM does not suffer from the problem of vanishing gradients, achieves better predictive performance on average and is more robust – however at the cost of longer run times. Third, sequence-based models tend not to generally outperform vector-based models – RNN even less so than LSTM. Although LSTM outperforms some vector-based models in terms of predictive performance or robustness, it does overall not represent a favorable alternative to many of the vector-based models when all six criteria of the evaluation framework are taken into account. It is worth noting that LSTM and RNN achieve competitive predictive performance on larger samples not relying on features that explicitly capture past visitor behavior. Instead, they seem to be able to learn such patterns on their own.

Considering all of the above, it is evident that there is no single best model that outperforms all others in all regards. Nevertheless, one is able to recognize certain tendencies, concluding that favorable alternatives constitute LR, DT, RF and BOOST as they are objective, fairly accurate, reasonably robust, interpretable, versatile and efficient in terms of run time as well. When it comes to optimizing these models though, significant differences reveal themselves as the number of hyperparameters to tune widely differs from model to model and thus also the effort associated with the task of model optimization (e.g., BOOST has much more hyperparameters to be tuned compared to LR)[10].

Managerial implications. Valuable insights for marketing can be derived from the experiments. For example, it is noteworthy that many models' predictive performance saturates for relatively small amounts of data already. This indicates that not necessarily the entirety of available data needs to be used to produce useful results. Moreover, some models allow for the computation of features' relative impacts on predicting conversions. These insights can be used by marketing executives to improve their decision making regarding different business aspects, such as customer retention management or conversion rate optimization. Potential

[10]On a side note, deep neural networks tend to be particularly hard to optimize, given the large number of hyperparameters and architectural choices as well as the complexity of the underlying optimization problem. See Goodfellow, Bengio, and Courville (2016) for details.

measures could be to use these insights for more customer-centric personalization or couponing, tailored to the respective customer's needs learned from the data and models.

But the experiments also show that using such models can be difficult given the plethora of available choices and associated caveats to consider. A thorough model evaluation using different criteria and also several performance metrics is therefore decisive to avoid false conclusions. A marketing executive needs to decide which compromises and trade-offs she is willing to accept. Should a model be able to recognize as many (potential) buyers as possible, that is high recall but including many false positives and thus low precision, or should a model rather be very confident in its predictions risking overlooking some (potential) buyers, that is high precision but low recall? Is it worth potentially sacrificing interpretability and algorithmic efficiency for the sake of better predictive accuracy? The application of machine learning and deep learning models for customer journey prediction can help marketing executives shed light onto these and other questions. This might lead to improving their decision making and in turn making their own decisions more explainable and justifiable. There are several promising alternatives for marketing executives to choose from, depending on the specific requirements of the problem, data and business environment and the trade-offs that need to be balanced.

8 Conclusion

The objective of this thesis is to investigate a selection of machine learning and deep learning models to predict conversions using e-commerce clickstream data and to evaluate them considering criteria that are relevant to marketing executives who use them as decision support. Data is a natural component of e-commerce and in combination with increasingly powerful and widespread machine learning and deep learning applications offers abundant potential to improve customer satisfaction and different business aspects, making this topic highly relevant.

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

Overall, LR, DT, RF and BOOST constitute the models that appear to best satisfy the evaluation framework's criteria since they are objective, accurate, robust, fairly interpretable, fast and allow for the computation of features' relative impacts on predictions. The latter particularly helps to better understand the more complex model structures of RF and BOOST. It is worth noting that BOOST ranks higher on average than the other three mentioned models in terms of AUC and F-score, even significantly outperforming LR and DT depending on the evaluation metric. No model, however, significantly outperforms all others, confirming similar findings and hypotheses derived from the meta-analysis. Besides, sequence-based models (RNN and LSTM) are not found to generally outperform most other vector-based models – especially not if one considers all of the model evaluation framework's criteria.

This thesis contributes to research by conducting a multi-dimensional meta-analysis of comparative machine learning studies from the past three decades, identifying general tendencies and particularly relevant works. It proceeds with the consolidation and extension of existing work in the sphere of customer journey prediction using machine learning and deep learning. This is achieved by increasing the scope of models and metrics used, based on the meta-analysis. Most importantly, it evaluates and compares the models under consideration with regard to what management and marketing science have found to be the most

relevant criteria to the addressees of these models, namely marketing executives. The experiments provide marketing executives with a comprehensive evaluation and comparison of a variety of models that can be used as a starting point to then make necessary adjustments and choices individual use cases and business environments require.

There are several limitations to this thesis that offer potential for future research. First, the meta-analysis of comparative studies could be extended as there are further insights certain to be retrieved from the vast body of available literature. Moreover, the analysis could strive to further condense empirical and theoretical results from comparative machine learning research to ultimately produce generalized rules of thumb or guidelines, regarding which model or model family tends to be recommendable for which problem and under which assumptions.

Second, the models are implemented and compared mostly using default hyperparameter settings. Although this considerably facilitates the conduction of the experiments, it might bias the comparison and evaluation in the sense that some models' defaults are inherently superior. Fine-tuning the hyperparameters of all considered models could enhance the significance of their comparison. Additionally, the entire evaluation could be done more thoroughly and individual criteria particularly emphasized depending on the specific use case at hand (e.g., testing and comparing methods to make complex models more interpretable).

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

Finally, the scope of customer journey prediction is broad, but only one aspect is explored in this thesis. A natural extension to predicting conversions in a binary fashion would be to predict conversion probabilities instead since they might represent a customer's propensity to purchase more accurately (e.g., Moe & Fader, 2004; Boroujerdi et al., 2014; Lang & Rettenmeier, 2017). In addition, the product(s) likely to be purchased in a given session could be predicted as well, moving the problem closer to the domain of recommender systems (e.g., Sarwar et al., 2015; Wu et al., 2015). These modified applications could then again be trialed using the model evaluation framework proposed by Anderl et al. (2014).

A Appendix

Table 7: Model abbreviations

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 8: Description of metric groups

Metric groups	**Descriptions**
Accuracy	Metrics related to accuracy, such as error, AUC, precision, recall 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.

Continued on next page

Table 8 – *Continued from previous page*

Metric groups	**Descriptions**
Miscellaneous	Other less frequently used metrics like cross-entropy and Cohen k.
Qualitative	Metric capturing comphrehensibility 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 range test.
Time	Metrics like training, testing and prediction time.

Table 9: Feature selection F- and p-*values* of all features

Features	***F*-values**	***p*-values**
cart_view_boolean_sum	139,298.2051	0.0000
user_gender_first_female	137,220.1141	0.0000
cart_addition_boolean_sum	113,868.7672	0.0000
user_gender_first_male	85,542.2805	0.0000
net_promoter_score_first_Unknown	72,214.0085	0.0000
user_age_36-45_first	56,486.5146	0.0000
user_age_26-35_first	45,626.9895	0.0000
page_view_boolean_sum	42,805.1345	0.0000
visit_page_num_max	38,690.9081	0.0000
user_age_46-55_first	36,042.6317	0.0000
product_view_boolean_sum	26,028.0626	0.0000
user_age_14-25_first	24,796.6652	0.0000
visit_duration_seconds	22,874.2933	0.0000
cart_removal_boolean_sum	22,469.2529	0.0000
net_promoter_score_first_10	19,854.6134	0.0000
last_purchase_num_max	19,774.5008	0.0000
user_age_56-65_first	16,330.8687	0.0000
net_promoter_score_first_8	15,699.7299	0.0000
marketing_channel_first_Paid Search G. Shopping	7,899.2662	0.0000
net_promoter_score_first_7	7,730.3923	0.0000
net_promoter_score_first_9	7,634.2026	0.0000
user_age_65_plus_first	6,726.4179	0.0000
net_promoter_score_first_5	6,705.8387	0.0000

Continued on next page

Table 9 – *Continued from previous page*

Features	***F*-values**	***p*-values**
net_promoter_score_first_6	5,426.7076	0.0000
device_type_user_agent_first_reduced_desktop	3,687.5914	0.0000
suggested_search_results_clicked_sum	3,290.8795	0.0000
marketing_channel_first_Paid Search Other	2,744.6967	0.0000
campaign_view_boolean_sum	2,734.6419	0.0000
product_categories_first_level_1_Schoenheit & Gesundheit	2,465.7384	0.0000
standard_search_started_sum	2,397.9616	0.0000
device_type_user_agent_first_reduced_smartphone	2,054.8272	0.0000
device_operating_system_user_agent_first_reduced _Windows	1,922.9261	0.0000
product_views_in_last_visit_2-5	1,815.8281	0.0000
device_brand_name_user_agent_first_reduced _Unknown	1,762.1659	0.0000
standard_search_results_clicked_sum	1,453.8475	0.0000
purchase_in_last_30_days	1,404.9873	0.0000
device_operating_system_user_agent_first_reduced _Mac	1,353.0430	0.0000
product_categories_first_level_1_Unknown	1,351.0870	0.0000
daily_visitor_first	1,324.8893	0.0000
weekly_visitor_first	1,298.3265	0.0000
device_operating_system_user_agent_first_reduced _iOS	1,263.1567	0.0000
net_promoter_score_first_3	1,251.4915	0.0000
monthly_visitor_first	1,193.9328	0.0000
net_promoter_score_first_4	1,193.9158	0.0000
yearly_visitor_first	1,172.6729	0.0000
quarterly_visitor_first	1,169.5488	0.0000
purchase_in_last_30_plus_days	1,048.3661	0.0000
visit_in_last_24_hours	990.9017	0.0000
product_views_in_last_visit_6-10	939.3123	0.0000
hit_of_logged_in_user_first	913.3604	0.0000
page_views_in_last_visit_11-20	904.4414	0.0000
page_views_in_last_visit_20_plus	844.8740	0.0000
may	798.0432	0.0000
marketing_channel_first_Social Media Prospecting	742.3289	0.0000

Continued on next page

Table 9 – *Continued from previous page*

Features	***F*-values**	***p*-values**
registered_user_first	729.7445	0.0000
product_items_sum	680.0924	0.0000
device_browser_user_agent_first_reduced_Firefox	600.6560	0.0000
referrer_type_first_internal	594.9086	0.0000
purchase_in_last_7_days	581.1653	0.0000
page_views_in_last_visit_6-10	564.9381	0.0000
referrer_type_first_social_network	556.6437	0.0000
marketing_channel_first_Display Prospecting	554.9200	0.0000
device_browser_user_agent_first_reduced_Facebook	550.5770	0.0000
referrer_type_first_search_engine	540.2521	0.0000
search_engine_first_reduced_Unknown	540.2521	0.0000
device_browser_user_agent_first_reduced_Internet Explorer	489.6883	0.0000
product_categories_first_level_1_Wohnen & Haushalt	481.8430	0.0000
product_views_in_last_visit_11-20	474.5107	0.0000
net_promoter_score_first_2	446.5033	0.0000
net_promoter_score_first_1	393.1214	0.0000
device_brand_name_user_agent_first_reduced _Samsung	389.1774	0.0000
referrer_type_first_external	351.8508	0.0000
product_views_in_last_visit_20_plus	287.7069	0.0000
product_categories_first_level_1_Mode & Accessoires	242.1921	0.0000
device_brand_name_user_agent_first_reduced _Apple	238.3461	0.0000
search_page_num_first	237.7717	0.0000
night	215.3672	0.0000
product_categories_first_level_1_Baumarkt & Garten	190.2410	0.0000
evening	185.1531	0.0000
august	181.2891	0.0000
page_views_in_last_visit_2-5	174.1737	0.0000
visit_in_last_7_days	169.8574	0.0000
device_browser_user_agent_first_reduced_Microsoft Edge	169.0213	0.0000

Continued on next page

Table 9 – *Continued from previous page*

Features	***F*-values**	***p*-values**
product_categories_first_level_1_Computer & Elektronik	159.2840	0.0000
device_type_user_agent_first_reduced_tablet	157.5823	0.0000
device_browser_user_agent_first_reduced_Safari	154.0295	0.0000
purchase_in_last_24_hours	139.2971	0.0000
product_views_in_last_visit_0	127.3281	0.0000
visit_during_tv_spot_first	125.1221	0.0000
cookies_first	120.1939	0.0000
Switzerland_first	117.1452	0.0000
morning	110.3111	0.0000
afternoon	102.9941	0.0000
marketing_channel_first_Direct	94.3774	0.0000
visit_num	85.4637	0.0000
marketing_channel_first_Social Media Non-Paid	79.4110	0.0000
product_categories_first_level_1_Sport & Freizeit	77.6181	0.0000
persistent_cookie_first	75.7754	0.0000
device_browser_user_agent_first_reduced_Samsung Browser	65.5183	0.0000
marketing_channel_first_Unknown	62.8962	0.0000
hourly_visitor_first	59.0399	0.0000
page_views_in_last_visit_1	58.9725	0.0000
device_type_user_agent_first_reduced_phablet	56.2352	0.0000
product_categories_first_level_1_Lebensmittel & Getraenke	43.5593	0.0000
device_operating_system_user_agent_first _reduced_Windows Phone	39.2718	0.0000
july	38.8917	0.0000
marketing_channel_first_Social Media Campaigns	35.5019	0.0000
marketing_channel_first_Referring Domains	35.1411	0.0000
device_brand_name_user_agent_first_reduced_Sony	32.0423	0.0000
repeat_orders_first	30.6992	0.0000
device_brand_name_user_agent_first_reduced _Nokia	27.4495	0.0000
product_views_in_last_visit_1	25.8129	0.0000
june	25.6435	0.0000

Continued on next page

Table 9 – *Continued from previous page*

Features	***F*-values**	***p*-values**
device_brand_name_user_agent_first_reduced _Other	25.5425	0.0000
connection_type_first_Mobile Carrier	23.9268	0.0000
weekday	21.4913	0.0000
weekend	21.4913	0.0000
device_brand_name_user_agent_first_reduced _Huawei	18.1476	0.0000
device_brand_name_user_agent_first_reduced_HTC	16.2417	0.0001
device_type_user_agent_first_reduced_portable media player	14.6949	0.0001
search_engine_first_reduced_Microsoft	14.0107	0.0002
marketing_channel_first_Paid Search Brand	13.6271	0.0002
visit_in_last_30_plus_days	13.5832	0.0002
product_item_price_sum	13.0656	0.0003
october	12.9397	0.0003
search_engine_first_reduced_Yahoo!	12.7671	0.0004
device_brand_name_user_agent_first_reduced _Microsoft	12.5035	0.0004
device_brand_name_user_agent_first_reduced _Google	11.7961	0.0006
september	10.8032	0.0010
visit_in_last_30_days	10.0541	0.0015
device_brand_name_user_agent_first_reduced_Wiko	8.5434	0.0035
device_brand_name_user_agent_first_reduced _Lenovo	7.9891	0.0047
device_brand_name_user_agent_first_reduced _Motorola	7.7613	0.0053
device_brand_name_user_agent_first_reduced_LG	7.0339	0.0080
page_views_in_last_visit_0	6.4080	0.0114
device_browser_user_agent_first_reduced_Chrome	5.8322	0.0157
marketing_channel_first_Social Media Remarketing	5.4663	0.0194
middle_of_month	4.3210	0.0376
device_type_user_agent_first_reduced_Unknown	3.7599	0.0525
device_brand_name_user_agent_first_reduced _OnePlus	3.3408	0.0676
beginning_of_month	3.2844	0.0699

Continued on next page

Table 9 – *Continued from previous page*

Features	***F*-values**	***p*-values**
marketing_channel_first_Mail Other	2.7042	0.1001
device_operating_system_user_agent_first_reduced _Unknown	2.1994	0.1381
device_brand_name_user_agent_first_reduced _Toshiba	2.1609	0.1416
referrer_type_first_nojs	2.0329	0.1539
device_operating_system_user_agent_first_reduced _Ubuntu	1.8954	0.1686
new_visit_first	1.6032	0.2055
device_brand_name_user_agent_first_reduced_RIM	1.5677	0.2105
marketing_channel_first_Editorial	1.2814	0.2576
device_browser_user_agent_first_reduced_Other	0.8961	0.3438
marketing_channel_first_Other Campaign	0.6283	0.4280
marketing_channel_first_Display Remarketing	0.4788	0.4890
device_browser_user_agent_first_reduced_Opera	0.3342	0.5632
device_brand_name_user_agent_first_reduced_Asus	0.3290	0.5662
marketing_channel_first_Organic Search	0.2826	0.5950
search_engine_first_reduced_Other	0.2728	0.6014
end_of_month	0.1175	0.7317
connection_type_first_Modem	0.0652	0.7985
device_operating_system_user_agent_first_reduced _Other	0.0514	0.8206
cart_value_sum	0.0435	0.8349
marketing_channel_first_Mail Flaschenpost	0.0361	0.8493
marketing_channel_first_Other - unidentified	0.0326	0.8568
device_operating_system_user_agent_first_reduced _GNU/Linux	0.0210	0.8848
product_categories_first_level_1_Medien & Unterhaltung	0.0169	0.8966
device_operating_system_user_agent_first_reduced _Chrome OS	0.0117	0.9138
marketing_channel_first_Digital Documents	0.0102	0.9197

Table 10: Top ten features LR

Features	**Coefficients**
purchase_in_last_24_hours	-4.4730
purchase_in_last_30_plus_days	-3.4944
purchase_in_last_7_days	-3.2762
purchase_in_last_30_days	-3.1003
user_gender_first_female	2.7581
user_gender_first_male	2.6112
repeat_orders_first	2.3587
product_categories_first_level_1_Unknown	-1.7776
user_age_56-65_first	1.7141
user_age_36-45_first	1.7120

Table 11: Top ten features SVM

Features	**Coefficients**
repeat_orders_first	1.4314
purchase_in_last_24_hours	-1.4154
purchase_in_last_30_plus_days	-1.2520
purchase_in_last_7_days	-1.1850
purchase_in_last_30_days	-1.1666
user_gender_first_female	0.5846
user_gender_first_male	0.5341
persistent_cookie_first	0.4599
user_age_56-65_first	0.4429
user_age_65_plus_first	0.4327

Table 12: Top ten features DT

Features	**Feature importances**
cart_view_boolean_sum	0.4074
last_purchase_num_max	0.1558
product_item_price_sum	0.0470
user_gender_first_male	0.0424
user_gender_first_female	0.0375
visit_duration_seconds	0.0356
product_items_sum	0.0268
visit_page_num_max	0.0246

Continued on next page

Table 12 – *Continued from previous page*

Features	**Feature importances**
page_view_boolean_sum	0.0180
product_view_boolean_sum	0.0145

Table 13: Top ten features RF

Features	**Feature importances**
cart_view_boolean_sum	0.1105
user_gender_first_female	0.0873
user_gender_first_male	0.0729
last_purchase_num_max	0.0685
cart_addition_boolean_sum	0.0536
product_items_sum	0.0432
page_view_boolean_sum	0.0392
net_promoter_score_first_Unknown	0.0372
product_item_price_sum	0.0372
visit_duration_seconds	0.0357

Table 14: Top ten features BOOST

Features	**Feature importances**
cart_view_boolean_sum	0.6279
last_purchase_num_max	0.1582
user_gender_first_male	0.0418
user_gender_first_female	0.0411
page_view_boolean_sum	0.0319
visit_num	0.0164
product_item_price_sum	0.0138
product_items_sum	0.0115
device_type_user_agent_first_reduced_smartphone	0.0080
user_age_36-45_first	0.0072

Table 15: Predictive performance metrics of all models for training and test sets with 3,125 and 781 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9820	0.8149	1075	9	16	11	0.5926	0.6400	0.6154
DT	0.9766	0.8513	1067	7	18	19	0.4865	0.7200	0.5806
NB	0.2079	0.5362	209	3	22	877	0.0245	0.8800	0.0476
KNN	0.9694	0.5936	1072	20	5	14	0.2632	0.2000	0.2273
RF	0.9829	0.8154	1076	9	16	10	0.6154	0.6400	0.6275
SVM	0.9739	0.8694	1063	6	19	23	0.4524	0.7600	0.5672
BOOST	0.9784	0.9303	1065	3	22	21	0.5116	0.8800	0.6471
NN1	0.9766	0.8317	1068	8	17	18	0.4857	0.6800	0.5667
NN3	0.9757	0.7922	1069	10	15	17	0.4688	0.6000	0.5263
NN5	0.9721	0.8099	1064	9	16	22	0.4211	0.6400	0.5079
RNN	0.9775	0.6172	1080	19	6	6	0.5000	0.2400	0.3243
LSTM	0.9784	0.6372	1080	18	7	6	0.5385	0.2800	0.3684

Table 16: Predictive performance metrics of all models for training and test sets with 6,250 and 1,562 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9806	0.7645	1946	26	30	13	0.6977	0.5357	0.6061
DT	0.9797	0.8247	1937	19	37	22	0.6271	0.6607	0.6435
NB	0.2164	0.5710	383	3	53	1576	0.0325	0.9464	0.0629
KNN	0.9742	0.6311	1948	41	15	11	0.5769	0.2679	0.3659
RF	0.9846	0.7666	1954	26	30	5	0.8571	0.5357	0.6593
SVM	0.9821	0.8347	1941	18	38	18	0.6786	0.6786	0.6786
BOOST	0.9861	0.8541	1947	16	40	12	0.7692	0.7143	0.7407
NN1	0.9851	0.8102	1950	21	35	9	0.7955	0.6250	0.7000
NN3	0.9841	0.8357	1945	18	38	14	0.7308	0.6786	0.7037
NN5	0.9831	0.8612	1940	15	41	19	0.6833	0.7321	0.7069
RNN	0.9732	0.5265	1958	53	3	1	0.7500	0.0536	0.1000
LSTM	0.9801	0.7296	1949	30	26	10	0.7222	0.4643	0.5652

Table 17: Predictive performance metrics of all models for training and test sets with 12,500 and 3,125 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9845	0.8190	4168	49	88	18	0.8302	0.6423	0.7243
DT	0.9810	0.8490	4144	40	97	42	0.6978	0.7080	0.7029
NB	0.4115	0.6749	1648	6	131	2538	0.0491	0.9562	0.0934
KNN	0.9766	0.6985	4167	82	55	19	0.7432	0.4015	0.5213
RF	0.9859	0.8374	4169	44	93	17	0.8455	0.6788	0.7530
SVM	0.9838	0.8469	4157	41	96	29	0.7680	0.7007	0.7328
BOOST	0.9864	0.8553	4166	39	98	20	0.8305	0.7153	0.7686
NN1	0.9829	0.8111	4163	51	86	23	0.7890	0.6277	0.6992
NN3	0.9845	0.8614	4156	37	100	30	0.7692	0.7299	0.7491
NN5	0.9845	0.8331	4164	45	92	22	0.8070	0.6715	0.7331
RNN	0.9752	0.6095	4186	107	30	0	1.0000	0.2190	0.3593
LSTM	0.9801	0.7320	4173	73	64	13	0.8312	0.4672	0.5981

Table 18: Predictive performance metrics of all models for training and test sets with 25,000 and 6,250 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9801	0.7854	8238	111	152	59	0.7204	0.5779	0.6414
DT	0.9762	0.8331	8177	84	179	120	0.5987	0.6806	0.6370
NB	0.8929	0.9042	7402	22	241	895	0.2121	0.9163	0.3445
KNN	0.9761	0.6949	8251	159	104	46	0.6933	0.3954	0.5036
RF	0.9850	0.8137	8266	97	166	31	0.8426	0.6312	0.7217
SVM	0.9838	0.8259	8248	90	173	49	0.7793	0.6578	0.7134
BOOST	0.9868	0.8662	8253	69	194	44	0.8151	0.7376	0.7745
NN1	0.9852	0.8138	8267	97	166	30	0.8469	0.6312	0.7233
NN3	0.9864	0.8457	8261	80	183	36	0.8356	0.6958	0.7593
NN5	0.9848	0.8302	8255	88	175	42	0.8065	0.6654	0.7292
RNN	0.9791	0.7444	8251	133	130	46	0.7386	0.4943	0.5923
LSTM	0.9853	0.8470	8250	79	184	47	0.7965	0.6996	0.7449

Table 19: Predictive performance metrics of all models for training and test sets with 50,000 and 12,500 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9817	0.8047	16025	193	310	112	0.7346	0.6163	0.6703
DT	0.9783	0.8338	15937	161	342	200	0.6310	0.6799	0.6545
NB	0.9281	0.8994	15007	66	437	1130	0.2789	0.8688	0.4222
KNN	0.9772	0.7118	16044	287	216	93	0.6990	0.4294	0.5320
RF	0.9845	0.8196	16058	179	324	79	0.8040	0.6441	0.7152
SVM	0.9837	0.8279	16036	170	333	101	0.7673	0.6620	0.7108
BOOST	0.9858	0.8646	16034	133	370	103	0.7822	0.7356	0.7582
NN1	0.9850	0.8622	16022	135	368	115	0.7619	0.7316	0.7465
NN3	0.9847	0.8351	16045	163	340	92	0.7870	0.6759	0.7273
NN5	0.9836	0.8702	15990	126	377	147	0.7195	0.7495	0.7342
RNN	0.9818	0.8211	16010	176	327	127	0.7203	0.6501	0.6834
LSTM	0.9834	0.8797	15976	116	387	161	0.7062	0.7694	0.7364

Table 20: Predictive performance metrics of all models for training and test sets with 100,000 and 25,000 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9834	0.8261	32380	367	706	191	0.7871	0.6580	0.7168
DT	0.9810	0.8627	32215	283	790	356	0.6894	0.7363	0.7120
NB	0.9300	0.9035	30351	134	939	2220	0.2972	0.8751	0.4438
KNN	0.9786	0.7411	32402	550	523	169	0.7558	0.4874	0.5926
RF	0.9863	0.8528	32420	311	762	151	0.8346	0.7102	0.7674
SVM	0.9842	0.8418	32373	333	740	198	0.7889	0.6897	0.7360
BOOST	0.9875	0.8840	32394	243	830	177	0.8242	0.7735	0.7981
NN1	0.9864	0.8898	32344	229	844	227	0.7880	0.7866	0.7873
NN3	0.9870	0.8711	32403	271	802	168	0.8268	0.7474	0.7851
NN5	0.9855	0.9019	32284	201	872	287	0.7524	0.8127	0.7814
RNN	0.9826	0.8820	32228	242	831	343	0.7078	0.7745	0.7397
LSTM	0.9855	0.8916	32308	224	849	263	0.7635	0.7912	0.7771

Table 21: Predictive performance metrics of all models for training and test sets with 200,000 and 50,000 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9848	0.8299	64721	699	1386	321	0.8120	0.6647	0.7310
DT	0.9813	0.8509	64389	601	1484	653	0.6944	0.7118	0.7030
NB	0.9340	0.9086	60858	247	1838	4184	0.3052	0.8815	0.4534
KNN	0.9821	0.7714	64784	945	1140	258	0.8155	0.5468	0.6546
RF	0.9874	0.8498	64812	619	1466	230	0.8644	0.7031	0.7755
SVM	0.9850	0.8400	64688	656	1429	354	0.8015	0.6854	0.7389
BOOST	0.9882	0.8769	64752	504	1581	290	0.8450	0.7583	0.7993
NN1	0.9881	0.8451	64882	641	1444	160	0.9002	0.6926	0.7829
NN3	0.9882	0.8693	64787	537	1548	255	0.8586	0.7424	0.7963
NN5	0.9883	0.8972	64674	417	1668	368	0.8193	0.8000	0.8095
RNN	0.9844	0.8754	64496	502	1583	546	0.7435	0.7592	0.7513
LSTM	0.9865	0.8890	64582	448	1637	460	0.7806	0.7851	0.7829

Table 22: Predictive performance metrics of all models for training and test sets with 400,000 and 100,000 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9835	0.8235	130076	1501	2820	725	0.7955	0.6526	0.7170
DT	0.9797	0.8526	129275	1223	3098	1526	0.6700	0.7170	0.6927
NB	0.9402	0.9051	123296	572	3749	7505	0.3331	0.8676	0.4814
KNN	0.9820	0.7806	130250	1878	2443	551	0.8160	0.5654	0.6679
RF	0.9863	0.8468	130258	1306	3015	543	0.8474	0.6978	0.7653
SVM	0.9841	0.8326	130082	1423	2898	719	0.8012	0.6707	0.7302
BOOST	0.9872	0.8764	130110	1045	3276	691	0.8258	0.7582	0.7905
NN1	0.9875	0.8737	130181	1071	3250	620	0.8398	0.7521	0.7936
NN3	0.9869	0.8601	130220	1190	3131	581	0.8435	0.7246	0.7795
NN5	0.9871	0.8632	130225	1163	3158	576	0.8457	0.7308	0.7841
RNN	0.9811	0.8546	129457	1212	3109	1344	0.6982	0.7195	0.7087
LSTM	0.9854	0.8333	130244	1422	2899	557	0.8388	0.6709	0.7455

Table 23: Predictive performance metrics of all models for training and test sets with 800,000 and 200,000 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9836	0.8267	259921	2932	5667	1489	0.7919	0.6590	0.7194
DT	0.9777	0.8519	257815	2428	6171	3595	0.6319	0.7176	0.6720
NB	0.9406	0.8985	246644	1260	7339	14766	0.3320	0.8535	0.4780
KNN	0.9825	0.7907	260250	3561	5038	1160	0.8128	0.5859	0.6809
RF	0.9869	0.8530	260367	2494	6105	1043	0.8541	0.7100	0.7754
SVM	0.9843	0.8406	259871	2690	5909	1539	0.7934	0.6872	0.7365
BOOST	0.9873	0.8736	260106	2131	6468	1304	0.8322	0.7522	0.7902
NN1	0.9876	0.8549	260539	2467	6132	871	0.8756	0.7131	0.7861
NN3	0.9879	0.8732	260284	2144	6455	1126	0.8515	0.7507	0.7979
NN5	0.9874	0.8593	260399	2387	6212	1011	0.8600	0.7224	0.7852
RNN	nan	nan	0	261410	0	0	nan	nan	nan
LSTM	0.9836	0.8638	259250	2271	6328	2160	0.7455	0.7359	0.7407

Table 24: Predictive performance metrics of all models for training and test sets with 1,600,000 and 400,000 unique visitors, respectively

Models	**Accuracy**	**AUC**	**True negatives**	**False negatives**	**True positives**	**False positives**	**Precision**	**Recall**	***F*-score**
LR	0.9838	0.8285	516605	5781	11360	2932	0.7949	0.6627	0.7228
DT	0.9816	0.8618	514246	4564	12577	5291	0.7039	0.7337	0.7185
NB	0.9415	0.9009	490561	2440	14701	28976	0.3366	0.8577	0.4834
KNN	0.9828	0.7980	517140	6847	10294	2397	0.8111	0.6005	0.6901
RF	0.9875	0.8616	517501	4678	12463	2036	0.8596	0.7271	0.7878
SVM	0.9843	0.8380	516546	5455	11686	2991	0.7962	0.6818	0.7346
BOOST	0.9878	0.8805	516987	4012	13129	2550	0.8374	0.7659	0.8001
NN1	0.9889	0.8886	517309	3746	13395	2228	0.8574	0.7815	0.8177
NN3	0.9890	0.8810	517630	4017	13124	1907	0.8731	0.7656	0.8159
NN5	0.9882	0.8586	517975	4795	12346	1562	0.8877	0.7203	0.7953
RNN	nan	nan	0	519537	0	0	nan	nan	nan
LSTM	0.9833	0.8713	514823	4256	12885	4714	0.7321	0.7517	0.7418

Table 25: Cross-validation metrics of all models for the sample with 7,812 unique visitors

Models	**Accuracy**	**AUC**	**Precision**	**Recall**	***F*-score**
LR	0.9808 (0.0044)	0.7791 (0.0422)	0.736 (0.0775)	0.5647 (0.0855)	0.6329 (0.0563)
DT	0.9794 (0.0045)	0.831 (0.0374)	0.6501 (0.0985)	0.673 (0.0747)	0.6568 (0.0672)
NB	0.2673 (0.0285)	0.6098 (0.0168)	0.0375 (0.0081)	0.9736 (0.0398)	0.0721 (0.0149)
KNN	0.9734 (0.0038)	0.6367 (0.0433)	0.6158 (0.1112)	0.2788 (0.0886)	0.3737 (0.0906)
RF	0.9821 (0.006)	0.7799 (0.067)	0.7934 (0.0946)	0.5644 (0.1352)	0.6484 (0.086)
SVM	0.9764 (0.0173)	0.8118 (0.0251)	0.6746 (0.1756)	0.6366 (0.0464)	0.6403 (0.1197)
BOOST	0.9843 (0.0048)	0.8369 (0.0426)	0.7692 (0.0738)	0.6803 (0.0856)	0.7184 (0.0592)
NN1	0.9826 (0.0057)	0.8363 (0.0422)	0.7337 (0.0715)	0.6803 (0.0841)	0.7023 (0.0609)
NN3	0.9819 (0.0056)	0.834 (0.0438)	0.7222 (0.1158)	0.6769 (0.0893)	0.6911 (0.074)
NN5	0.9821 (0.0034)	0.8618 (0.0616)	0.6926 (0.0765)	0.7334 (0.1252)	0.7035 (0.0612)
RNN	0.9747 (0.0074)	0.5983 (0.0675)	0.7838 (0.1969)	0.1982 (0.1354)	0.3005 (0.1654)
LSTM	0.9764 (0.0064)	0.6972 (0.0407)	0.6729 (0.1113)	0.4006 (0.0819)	0.4975 (0.0803)

Table 26: Cross-validation metrics of all models for the sample with 31,250 unique visitors

Models	**Accuracy**	**AUC**	**Precision**	**Recall**	***F*-score**
LR	0.982 (0.0018)	0.8136 (0.0242)	0.7413 (0.0551)	0.6341 (0.049)	0.6814 (0.0335)
DT	0.9795 (0.0024)	0.8403 (0.0271)	0.655 (0.0399)	0.692 (0.055)	0.6713 (0.0316)
NB	0.8829 (0.0292)	0.8957 (0.0178)	0.2013 (0.0439)	0.9093 (0.0289)	0.3275 (0.0595)
KNN	0.9773 (0.0034)	0.7165 (0.0206)	0.7037 (0.0388)	0.4387 (0.0413)	0.539 (0.0319)
RF	0.9857 (0.0017)	0.8263 (0.0171)	0.8363 (0.046)	0.6566 (0.0339)	0.7351 (0.0321)
SVM	0.9832 (0.0032)	0.8339 (0.0244)	0.7558 (0.076)	0.6748 (0.0494)	0.7103 (0.044)
BOOST	0.9867 (0.0027)	0.8665 (0.0173)	0.8087 (0.0471)	0.7384 (0.0335)	0.7716 (0.0364)
NN1	0.9855 (0.0017)	0.8477 (0.0327)	0.8032 (0.0449)	0.701 (0.067)	0.7455 (0.0309)
NN3	0.9858 (0.002)	0.8409 (0.0335)	0.8212 (0.0523)	0.6865 (0.0684)	0.7443 (0.031)
NN5	0.9858 (0.0023)	0.8699 (0.027)	0.7775 (0.0496)	0.7465 (0.0542)	0.7603 (0.0406)
RNN	0.9801 (0.0037)	0.7961 (0.0443)	0.7181 (0.0697)	0.6002 (0.0918)	0.6453 (0.0392)
LSTM	0.9839 (0.0016)	0.8663 (0.0251)	0.734 (0.0539)	0.741 (0.052)	0.7347 (0.0237)

Table 27: Cross-validation metrics of all models for the sample with 125,000 unique visitors

Models	**Accuracy**	**AUC**	**Precision**	**Recall**	***F*-score**
LR	0.9834 (0.0015)	0.8237 (0.0137)	0.7809 (0.021)	0.6533 (0.0273)	0.7111 (0.0191)
DT	0.98 (0.0017)	0.8467 (0.0156)	0.6704 (0.027)	0.7045 (0.0309)	0.6869 (0.0264)
NB	0.9296 (0.0057)	0.9032 (0.0061)	0.292 (0.0264)	0.8749 (0.0118)	0.4373 (0.0299)
KNN	0.9791 (0.002)	0.7438 (0.0177)	0.7525 (0.0222)	0.4928 (0.0355)	0.595 (0.0293)
RF	0.9862 (0.0011)	0.8435 (0.0076)	0.8382 (0.0294)	0.6913 (0.0152)	0.7575 (0.0157)
SVM	0.984 (0.0021)	0.8438 (0.0157)	0.7719 (0.0414)	0.6943 (0.0312)	0.7304 (0.0285)
BOOST	0.9874 (0.0013)	0.878 (0.0119)	0.8221 (0.0151)	0.7613 (0.0236)	0.7904 (0.0169)
NN1	0.9872 (0.0008)	0.8595 (0.0154)	0.8453 (0.0159)	0.7232 (0.0314)	0.779 (0.0157)
NN3	0.9872 (0.0014)	0.8673 (0.0143)	0.8325 (0.0176)	0.7393 (0.0286)	0.7829 (0.0197)
NN5	0.9869 (0.0007)	0.8712 (0.0095)	0.8175 (0.016)	0.7477 (0.0193)	0.7809 (0.0144)
RNN	0.9814 (0.0017)	0.802 (0.0244)	0.7518 (0.0557)	0.6106 (0.0498)	0.6712 (0.0303)
LSTM	0.9854 (0.0015)	0.8392 (0.0209)	0.8213 (0.0394)	0.6832 (0.0428)	0.7442 (0.0196)

Table 28: Cross-validation metrics of all models for the sample with 500,000 unique visitors

Models	**Accuracy**	**AUC**	**Precision**	**Recall**	***F*-score**
LR	0.9837 (0.0008)	0.8284 (0.006)	0.7871 (0.011)	0.6627 (0.0119)	0.7195 (0.0089)
DT	0.9807 (0.0004)	0.8544 (0.0054)	0.6847 (0.013)	0.7197 (0.0111)	0.7017 (0.0076)
NB	0.9401 (0.0019)	0.9017 (0.0032)	0.3293 (0.0078)	0.8606 (0.0055)	0.4763 (0.0085)
KNN	0.9822 (0.0005)	0.7859 (0.0047)	0.8071 (0.0094)	0.5763 (0.0095)	0.6724 (0.0066)
RF	0.9866 (0.0004)	0.8514 (0.0061)	0.8452 (0.0081)	0.7069 (0.0122)	0.7699 (0.0083)
SVM	0.9839 (0.001)	0.8441 (0.0061)	0.7756 (0.027)	0.6948 (0.013)	0.7326 (0.0095)
BOOST	0.9873 (0.0007)	0.8803 (0.0048)	0.8209 (0.0137)	0.766 (0.0095)	0.7924 (0.0082)
NN1	0.988 (0.0006)	0.8682 (0.0109)	0.8626 (0.0158)	0.7403 (0.0222)	0.7965 (0.01)
NN3	0.9879 (0.0004)	0.875 (0.0124)	0.8478 (0.0195)	0.7544 (0.0257)	0.7978 (0.0063)
NN5	0.9878 (0.0006)	0.8703 (0.0144)	0.8517 (0.0125)	0.7449 (0.0294)	0.7942 (0.0134)
RNN	0.6877 (0.4745)	0.7614 (0.1811)	0.7069 (0.043)	0.5301 (0.3674)	0.7292 (0.0132)
LSTM	0.9849 (0.0004)	0.8814 (0.015)	0.7576 (0.0271)	0.7709 (0.0312)	0.7633 (0.0081)

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *Tensorflow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/`

Adobe Systems Incorporated. (2019). *Adobe experience cloud: Analytics help and reference.* Retrieved 14.04.2019, from `https://adobe.ly/2VO35Fj`

Anderl, E., Becker, I., Wangenheim, F. V., & Schumann, J. H. (2014). Mapping the customer journey: A graph-based framework for online attribution modeling. *SSRN*(2343077). doi: 10.2139/ssrn.2343077

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*(1/2), 105–139. doi: 10.1023/A:1007515423169

Baumann, A., Haupt, J., Gebert, F., & Lessmann, S. (2018). Changing perspectives: Using graph metrics to predict purchase probabilities. *Expert Systems with Applications*, *94*, 137–148. doi: 10.1016/j.eswa.2017.10.046

Ben-Shimon, D., Tsikinovsky, A., Friedmann, M., Shapira, B., Rokach, L., & Hoerle, J. (2015). Recsys challenge 2015 and the yoochoose dataset. *Proceedings of the 9th ACM Conference on Recommender Systems*, 357–358.

Boroujerdi, E. G., Mehri, S., Garmaroudi, S. S., Pezeshki, M., Mehrabadi, F. R., Malakouti, S., & Khadivi, S. (2014). A study on prediction of user's tendency toward purchases in websites based on behavior models. In *2014 6th conference on information and knowledge technology (ikt)* (pp. 61–66).

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), 1145–1159.

Bucklin, R. E., & Sismeiro, C. (2009). Click here for internet insight: Advances in clickstream data analysis in marketing. *Journal of Interactive Marketing*, *23*(1), 35–48. doi: 10.1016/j.intmar.2008.10.004

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, 161–168. doi: 10.1145/1143844.1143865

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. doi: 10.1613/jair.953

Chollet, F., et al. (2015). *Keras.* Retrieved from `https://keras.io/`

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*(Jan), 1–30.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

European Commission. (1994). *Comparative testing and evaluation of statistical and logical learning algorithms for large-scale applications in classification, prediction and control.* Retrieved 14.04.2019, from `https://cordis.europa.eu/project/rcn/8791/factsheet/en`

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*(1), 3313-3181.

Freitas, A. A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, *15*(1), 1–10.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, *32*(200), 675–701.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, *11*(1), 86–92.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT press.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 93.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, J., Lu, J., & Ling, C. X. (2003). Comparing naive bayes, decision trees, and svm with auc and accuracy. In *Third ieee international conference on data mining* (pp. 553–556).

Khan, M. M. R., Arif, R. B., Siddique, A. B., & Oishe, M. R. (2018). Study and observation of the variation of accuracies of knn, svm, lmnn, enn algorithms on eleven different datasets from uci machine learning repository. In *2018 4th international conference on electrical engineering and information & communication technology (iceeict)* (pp. 124–129).

King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, *9*(3), 289–333. doi: 10.1080/08839519508945477

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lang, T., & Rettenmeier, M. (2017). Understanding consumer behavior with recurrent neural networks. In *Workshop on machine learning methods for recommender systems.*

Lilien, G. L. (2011). Bridging the academic–practitioner divide in marketing decision models. *Journal of Marketing*, *75*(4), 196–210. doi: 10.1509/jmkg.75.4.196

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*(3), 203–228.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Little, J. D. C. (1970). Models and managers: The concept of a decision calculus. *Management Science*, *16*(8), B-466-B-485. doi: 10.1287/mnsc.16.8.B466

Little, J. D. C. (2004). Models and managers: The concept of a decision calculus. *Management Science*, *50*(12_supplement), 1841–1853. doi: 10.1287/mnsc.1040.0267

Lodish, L. M. (2001). Building marketing models that make money. *Interfaces*, *31*(3_supplement), S45-S55. doi: 10.1287/inte.31.3s.45.9681

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine learning, neural and statistical classification*. Ellis Horwood Limited.

Moe, W. W., Chipman, H., George, E. I., & McCulloch, R. E. (2004). A bayesian treed model of online purchasing behavior using in-store navigational clickstream. *Working paper*.

Moe, W. W., & Fader, P. S. (2004). Dynamic conversion behavior at e-commerce sites. *Management Science*, *50*(3), 326–335. doi: 10.1287/mnsc.1040.0153

Montgomery, A. L., Li, S., Srinivasan, K., & Liechty, J. C. (2004). Modeling online browsing and path analysis using clickstream data. *Marketing Science*, *23*(4), 579–595. doi: 10.1287/mksc.1040.0073

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons* (PhD thesis). Princeton University.

Olivier, C., Eren, M., & Rosales, R. (2014). Simple and scalable response prediction for display advertising [j]. *ACM Transactions on Intelligent Systems and Technology*, *5*(4), 1–34.

Olson, R. S., La Cava, W., Mustahsan, Z., Varik, A., & Moore, J. H. (2018). Data-driven advice for applying machine learning to bioinformatics problems. In *Pacific symposium on biocomputing* (Vol. 23, pp. 192–203). World Scientific Publishing Company.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from `https://scikit-learn.org/stable/index.html`

Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, *4*(Jun), 211–255.

Provost, F., & Domingos, P. (2003). Tree induction for probability-based ranking. *Machine Learning*, *52*(3), 199–215.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd acm*

sigkdd international conference on knowledge discovery and data mining (pp. 1135–1144).

Sakar, C. O., Polat, S. O., Katircioglu, M., & Kastro, Y. (2018). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. *Neural Computing and Applications*, *39*(12), 11243. doi: 10.1007/s00521-018-3523-0

Salzberg, S. L. (1999). On comparing classifiers: A critique of current research and methods. *Data mining and knowledge discovery*, *1*(1).

Sarwar, S. M., Hasan, M., & Ignatov, D. I. (2015). *Two-stage cascaded classifier for purchase prediction.* Retrieved from `http://arxiv.org/pdf/1508.03856v1`

Shavlik, J. W., Mooney, R. J., & Towell, G. G. (1991). Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning*, *6*(2), 111–143.

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

Sismeiro, C., & Bucklin, R. E. (2004). Modeling purchase behavior at an e-commerce web site: A task-completion approach. *Journal of marketing research*, *41*(3), 306–323.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In D. Hutchison et al. (Eds.), *Ai 2006: Advances in artificial intelligence* (Vol. 4304, pp. 1015–1021). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/11941439{\textunderscore}114

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Suh, E., Lim, S., Hwang, H., & Kim, S. (2004). A prediction model for the purchase probability of anonymous customers to support real time web marketing: a case study. *Expert Systems with Applications*, *27*(2), 245–255. doi: 10.1016/j.eswa.2004.01.008

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3319–3328).

Tan, A. C., & Gilbert, D. (2003). An empirical comparison of supervised machine learning techniques in bioinformatics. In *Proceedings of the first asia-pacific bioinformatics conference on bioinformatics 2003-volume 19* (pp. 219–222).

Toth, A., Tan, L., Di Fabbrizio, G., & Datta, A. (2017). Predicting shopping behavior with mixture of rnns. In *Acm sigir forum. acm.*

Vieira, A. (2015). *Predicting online user behaviour using deep learning algorithms.* Retrieved from `http://arxiv.org/pdf/1511.06247v3`

Wedel, M., & Kannan, P. K. (2016). Marketing analytics for data-rich environments. *Journal of Marketing*, *80*(6), 97–121. doi: 10.1509/jm.15.0413

Weis, S. M., & Kapouleas, I. (1989). An empirical comparison of patter recognition, neural nets, and machine learning classification algorithms. In *Proceedings of the 11th international joint conference on ai* (pp. 781–787).

Wu, Z., Tan, B. H., Duan, R., Liu, Y., & Mong Goh, R. S. (2015). Neural modeling of buying behaviour for e-commerce from clicking patterns. In D. Ben-Shimon, M. Friedmann, L. Rokach, & B. Shapira (Eds.), *Proceedings of the 2015 international acm recommender systems challenge on - recsys '15 challenge* (pp. 1–4). New York, New York, USA: ACM Press. doi: 10.1145/2813448.2813521