Erasmus School of Economics

MSc Data Science and Marketing Analytics

Master Thesis

Modelling the Price of Display Advertisements in a Real-time Bidding Environment

Name student: Koen Bosmans

Student ID number: 539823

Supervisor: Dr. A.J. Koning

Second assessor: S.L. Malek

Date final version: 31-12-2020

ABSTRACT

Real-time bidding (RTB) is among the most influential state-of-the art marketing technologies. By allowing ad space to be auctioned in real-time, the market for online advertisements has significantly shifted, which opened a range of new research possibilities. In this paper, the price of ad impressions has been modelled using five different methods. By using real RTB data, a model comparison study was performed that aimed to discover which techniques were able to most accurately predict the price of ad space. Conclusevely, the results have indicated that accuracy was moderate across all models with no clear winners. Limiting factors could be attributed to data quality and model set-up. In the last section, recommendations were made on how performance could be improved upon in future research.

Keywords: Real-Time Bidding, Demand-Side Platform, Digital Advertising, Quantitative Marketing, Bid Landscape Forecasting

TABLE OF CONTENTS

List of Abbreviations

ADN	Ad network
ADX	Ad exchange
AIC	Akaike information criterion
BIC	Bayesian information criterion
CV	Cross-validation
DSP	Demand-side platform
EM	Expectation maximization
FMR	Finite mixture of regressions
KNN	K-nearest neighbors
MAE	Mean absolute error
RSS	Residual sum of squares
RTB	Real-time bidding
SSP	Supply-side platform

1. INTRODUCTION

Upon browsing the web, it is hardly impossible not to be exposed to the overwhelming amount of digital advertisements that are present. Not only do these ads keep increasing in volume, its contents are getting more and more tailored as well. This is caused by the rapid evolution within digital advertising, which industry has grown strongly over the past few decades. Where previously online ads were sold over the phone to negotiate terms, nowadays most deals are settled in the time it takes to load a webpage.

After the first digital advertisement launched in 1994, the demand for online ad space has constantly risen. In 2019, digital ad spending in the US surpassed the marketing budget spent on traditional media and this difference is expected to continue to grow in the future (eMarketer, 2019). The increasing demand for online advertising inventory lies in parallel with the rise of global internet, which created a more complex environment for digital advertising. It triggered the need for more sophisticated technologies to help automate and streamline the process of online ad buying, which resulted in the emergence of programmatic advertising. This marketing technology allows to automatically buy digital ad space by predefining a set of parameters. Currently, businesses can use programmatic advertising to launch a targeted online ad campaign by entering a few input fields.

The infrastructure that enables the automated buying and selling of display opportunity is called real-time bidding (RTB) (Wang, Zhang & Yuan, 2017). RTB is a digital auction process that allows ad impressions to be put up for bid in real-time on the ad exchange (ADX). There are generally two types of platforms connected to the ADX, which are the supply-side platform (SSP) and the demand-side platform (DSP). Publishers trying to sell digital ad space are organized under a SSP that registers their ad inventory and accepts winning bids. The DSP represents the advertisers and helps manage their online ad campaigns by systematically bidding on the offered ad space on the ADX.

RTB is related to various scientific fields including finance, artificial intelligence, machine learning and marketing. From the perspective of advertisers, buying ad space is an investment of acquiring new customers with similar risks involved as to trading on the financial markets (Zhang et al., 2017). Furthermore, elements from artificial intelligence and machine learning can be used to optimize bidding strategies and minimize cost per conversion as illustrated by Ren et al. (2018). To marketers, RTB has reshaped the landscape of digital advertising by

enabling to trade impressions on a per-case basis. This enhances transparency and efficiency of online campaigns by allowing for personalized advertising, which is positively related to the effectiveness of an ad (Tucker, 2014). Moreover, it presents a new trade-off for both advertisers and publishers who must now decide between reservation contracts and RTB for the buying and selling of ad space. This has several implications on the strategy and profitability of both parties as was analysed by Sayedi (2018) using a game-theoretic model.

Unfortunately, a lot about RTB remains unknown due to its rapidly changing environment and a lack of usable datasets (Zhang, Yuan, Wang & Shen, 2014). With RTB becoming increasingly present in society it is imperative to obtain an adequate understanding of the technology and its implications. Transparency is needed regarding a variety of topics such as pricing, viewability, fraud and more. In this paper, focus will be dedicated to investigating various statistical models' ability in predicting the price that is paid for impressions in the RTB market. This could provide a framework to publishers for forecasting their ad revenue and help understand customer value from an advertising perspective. Moreover, it might interest advertisers that aim to further optimize their bidding algorithms or assist economists that pursue to capture the characteristics of this newly opened market.

By using the data of a major Chinese DSP, multiple statistical and machine learning models have been compared on their accuracy in predicting the price of ad space on the RTB market. The techniques that are used include different linear regressions, a nearest neighbor model and an artificial neural network. Since market heterogeneity may affect model performance, a mixture of regressions has been performed to assess the contribution of market segments in modelling ad prices. Altogether, the research aims to answer the following research question: How to most accurately predict the price of an impression in the real-time bidding market for display advertisements?

This paper comprises the following structure. In the next section, the methodology of this research will be extensively covered. This includes a summary of the RTB environment, a general description of the data, and an explanation of the statistical techniques and selection criteria that are used. Subsequently, the results of the statistical models are presented and evaluated. Lastly, the final section lists the concluding remarks, discusses the limitations and provides recommendations for future research.

2. METHODOLOGY

2.1. Real-Time Bidding Ecosystem

The first platforms focusing on the RTB-based trading of ad impressions emerged over a decade ago and are currently known as ad exchanges (Wang, Zhang & Yuan, 2017). Contrary to a traditional ad network (ADN), these ADXs introduced real-time auctions to balance and centralize the demand and supply in the digital advertising market. This fundamentally changed the landscape of online advertising in multiple areas. Their entrance signified the shift from contextual advertising to behavioral targeting in display advertising and presented a range of new benefits to advertisers and publishers. Furthermore, RTB significantly scaled up the transaction volume in display advertising which is now likely to exceed the daily number of shares that are traded on the financial market (Wang, Zhang & Yuan 2017).

To capture the RTB system, it is useful to first zoom in on a simplified path of an individual impression as described by Wang, Zhang and Yuan (2017). For every user that visits a webpage, an impression is generated at a web publisher. In the time it takes the page to load, an ad request gets sent to an ADX that queries the bids from different advertisers. The advertiser with the highest bid is selected and notified after which its ad gets displayed to the specific user. Lastly, the advertiser can track user feedback such as whether the person clicked or converged following from the ad. Altogether, this entire process – from the impression being fetched to the advertisement being shown – takes approximately 100 milliseconds (Wang, Zhang and Yuan, 2017).

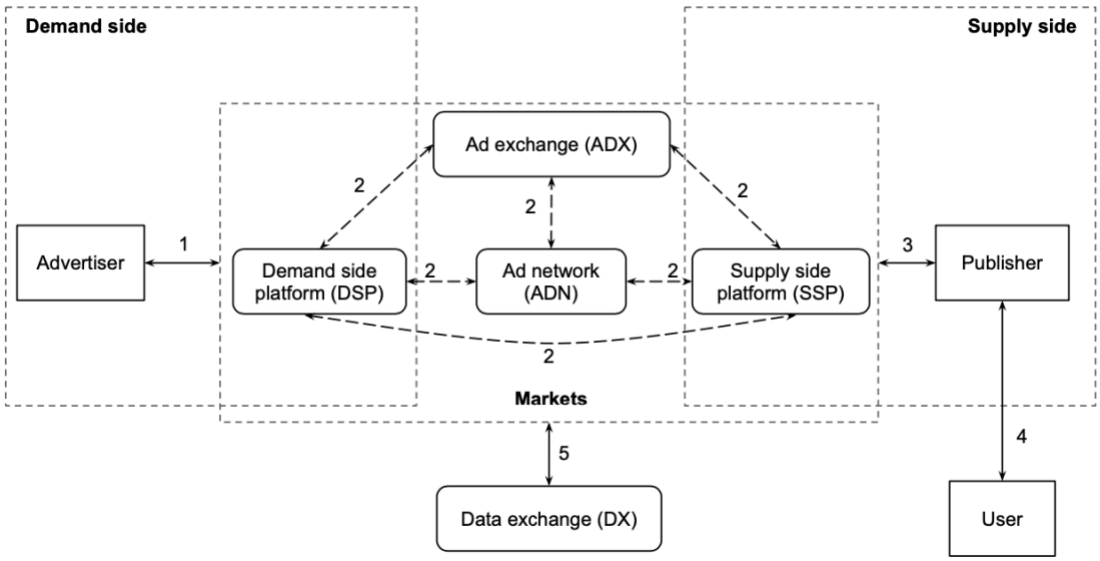
There are multiple entities that play an important role in the RTB process. In general, there are two main platforms connected to the ad exchange which are the supply-side platform and the demand-side platform (Yuan, Abidin, Sloan & Wang, 2012). A SSP represents the publishers by automatically registering ad space, accepting bids and placing ads. It aggregates the impressions of multiple publishers and ADNs, which are then put up for bid on the ADX. A DSP serves the advertisers in their digital ad campaigns by automatically bidding for impressions on multiple ADXs. The platform relies on sophisticated algorithms for doing so that provide a competitive return to clients. This return is often measured using the expected value of conversions and clicks that follow from an ad campaign (Wang, Zhang & Yuan 2017).

To train the algorithm, it requires user data. Some of this is automatically provided by the publisher at the time the impression gets created. However, since comprehensive user data is in such high demand, data exchanges (DX) emerged to seize the profit opportunity. A DX collects historical user data using advanced information technology, which it sells to companies in the digital advertising market. A graphical overview that summarizes the RTB environment is provided in Figure 1 (Yuan et al., 2012).

To successfully trade ad impressions, the auction set-up adopts a couple distinct features. An important characteristic of a RTB auction is that it is commonly structured as a Vickrey or second-price auction (Wang, Zhang & Yuan 2017). The bids are submitted in sealed form and the winner of the auction pays the price of the second highest bidder. This ensures that participants submit bids that are equal to their own valuation as demonstrated by Milgrom (2004). Furthermore, most ad impressions that are being auctioned contain a floor price. This is the minimal price for which a publisher would allow an impression to be sold as is communicated to the participants. Impressions failing to deliver that value on the auction are often bundled and sold using reservation contracts.

Figure 1

Overview of RTB Market



Note. Adapted from *Internet advertising: an interplay among advertisers, online publishers, ad exchanges and web users,* by S. Yuan et al., 2012.

2.2. Data

The dataset used in this research is based on RTB data released by the Chinese DSP iPinYou (Liao, Peng, Liu & Shen, 2014). As one of the largest DSPs in the county, iPinYou provides businesses with big data and AI solutions in the context of digital marketing. In 2013, the company organized a global RTB algorithm contest to help improve its DSP bidding algorithm and stimulate scientific growth in the RTB environment (Liao et al., 2014). After the competition ended, the datasets were made public for further research.

In total, the iPinYou dataset consists of close to twenty million impressions and the bids of twelve advertisers over the course of three different time periods or seasons. It is structured in four different logs of biddings, impressions, clicks and final conversions that each contain information regarding the user, the ad slot and the bidding information. An overview of the log data format has been provided in Table B1. However, for a more thorough description of the data it is recommended to consult the iPinYou competition supplement by Liao et al. (2014) and the research of Zhang et al. (2014).

This paper selected the impression log of the data for its analysis since it contains all the advertisements that have been broadcasted to a web visitor. Here, the objective was to predict the paying price of an impression since it resembles the general valuation of the ad space. The price unit was denoted in RMB per CPM as is a common metric in display advertising (Wang, Zhang & Yuan, 2017). The variables that were used for modelling this price include: the timestamp, user agent, region, ad exchange, ad slot width, ad slot height, ad slot visibility and the ad slot floor price.

For the analysis, a sub-sample of the data was used for faster calculations with limited information loss. The third season has been selected for sampling since it contains the most recent data and comprises the longest consecutive time span. From this data, 10,000 impressions have been randomly sampled for conducting the statistical analyses and model evaluation. To further prepare the data, multiple preparation steps were performed. This included removing missing values, converting the timestamp, recoding user agent information, and merging the ad slot width and height to common resolutions.

2.3. Supervised Learning

Forecasting the paying price in auctions is categorized as a supervised learning problem. In this area of machine learning, a set of independent variables is used to predict one or more dependent variables. Since price is a quantitative variable, a regression technique can be used to estimate the statistical relation. This research compares various supervised learning techniques for modelling paying price, which is elaborated on in the remainder of this section.

2.3.1. Linear Regression

A standard approach for modelling conventional auctions is by estimating the linear relationship between the variables. Linear models have a long history in statistics and are still widely used as of today (James, Witten, Hastie & Tibshirani, 2013). There are a great number of extensions and generalizations of the linear model all varying in terms of complexity and applicability. In this paper, a general linear regression will be set up that serves as a benchmark model to its extensions and machine learning techniques.

Let us start by considering a linear regression model that uses one predictor variable X for predicting the response variable Y linearly as formalized by James et al. (2013). This model can be written in the following functional form $Y \approx \beta_0 + \beta_1 X$ where β_0 and β_1 are the model coefficients or parameters. These coefficients capture the linear relationship between the variables in the population with β_0 as intercept and β_1 as slope. Since this population is unlikely to be fully observable its parameters are usually unknown. By taking a sample with n observations of x and y, and i being equal to $1, \dots, n$, the future value of y can be estimated as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. The hat symbol, ˆ, is used to denote the estimated value of a parameter or the predicted value of the response variable.

In the real-world, the predicted value $\hat{y}$ often differs from the observed value y resulting in a residual or error. Mathematically, for every ith observation the error e is given by $e_i = y_i - \hat{y}_i$, which let us write $y = \hat{\beta}_0 + \hat{\beta}_1 x + e$. The errors play a fundamental role in the linear model as they are used for estimating the model coefficients by optimizing a cost-function such as the least squared criterion. This is a method of finding coefficients subject to minimizing the residual sum of squares (RSS) defined as $RSS = \sum_{i=1}^{n} e_i^2$. Conceptually, it attempts to fit a line to the data as to minimize its distance to all of the observed values.

The simple regression can easily be extended to a multiple regression model, which is expressed as $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$ with p predictor variables. Here, the population error ε is introduced to acknowledge that most real-life relationships are not perfectly linear. The coefficients of this model are estimated in a similar procedure as for the simple regression by minimizing the RSS. However, instead of fitting a line it aims to fit a multidimensional hyperplane to the data making it harder to visualize.

2.3.2. Box-Cox Transformation

The standard linear model relies on various statistical assumptions to reasonably depict real-world relations and provide meaningful results (James et al., 2013). In general, these include adhering to a linear response-predictor relationship, a constant variance of the errors and no correlation among the errors. When these assumptions are not met, potential problems may arise that can lead to illegitimate claims (James et al., 2013). Identifying and overcoming these problems has been extensively researched and there are countless ways of doing so.

A popular approach for addressing violations of the model assumptions is to perform a Box-Cox transformation on the dependent variable as was introduced by Box and Cox in 1964. By transforming y, its distribution will be adjusted to better meet statistical properties of the linear model. The Box-Cox transformation specifically aims to reduce non-normality of the errors, but it can also help to combat heteroscedasticity and non-linearity (Box & Cox, 1964). Mathematically, the Box-Cox transformation of y – given that the variable is strictly positive – using the power parameter λ can be expressed as

$$y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda} - 1}{\lambda} & if\ \lambda \neq 0; \\ \log(y) & if\ \lambda = 0. \end{cases}$$

What distinguishes the Box-Cox transformation from other power transformation techniques is that it uses a maximum-likelihood estimation for finding the optimal value for λ. In statistics, likelihood refers to the goodness of fit of a certain distribution in belonging to the observed data (Friedman, Hastie & Tibshirani, 2001). This value is expressed by the likelihood function which consists of unknown model parameters and a given sample. Likelihood estimation aims to find the model parameters by maximizing the likelihood function under some assumed type of distribution with respect to the observed data. This yields the distribution under which a given sample is most probable. For the Box-Cox transformation, the maximized log-likelihood

function for a fixed λ is expressed as $L_{max}(\lambda) = -\frac{1}{2} n \log \hat{\sigma}^2(\lambda) + (\lambda - 1) \sum_{i=1}^{n} \log y_i$. By plotting the maximized log-likelihood $L_{max}(\lambda)$ against λ for a trial series of values, the optimal λ can be found.

2.3.3. Finite Mixture of Regressions

This research uses a mixture model to control for market heterogeneity and uncover the market segmentation for online display advertisements. Finite mixture models use a probabilistic approach to detect the presence of sub-populations within a population without the need of observing them a priori (McLachlan & Basford, 1988). This model class outperforms the commonly used heuristic-based methods that are used in market segmentation since these are likely to contain insufficient statistical basis to uncover the true cluster nature (Tuma & Decker, 2013).

In the context of clustering, mixture models assume that a set of observations can be generated by a mixture of models that each represent a different sub-population or cluster (McLachlan & Basford, 1988). Hence, each cluster is characterized by a certain model in contrast to the similarity among observations that is used in traditional statistical clustering techniques. A mixture model relies on maximum likelihood to fit a set of models to the data and find the underlying group structure (McLachlan & Basford, 1988).

This approach allows to obtain a probabilistic clustering of the observations by deriving the estimated posterior probabilities of group memberships. Hence, each observation can be related to multiple sub-groups in the data with a different probability. Allowing for uncertainty of belonging to a cluster expressed by a membership function is commonly referred to as fuzzy or soft clustering (Yang, 1993). By subsequently assigning each entity to the cluster it has the highest probability of belonging to, a non-fuzzy or hard clustering solution can be obtained.

The most common implementation of a mixture model involves fitting a mixture of Gaussian distributions to a multi-modal dataset (McLachlan & Basford, 1988). This will improve density estimation and help discover sub-populations in the data. However, when applied in a market segmentation analysis its solution will be mostly descriptive (Wedel & Kamakura, 2001). The resulting segments will consist of observations that are largely homogenous in their attributes such as product features or geographical information.

A different approach to the regular mixture model that is often used in market segmentation is to fit a mixture of regressions to the data (DeSarbo & Cron, 1988). Here, clusters are formed based on the inferred relationship between the dependent variable and a set of independent variables within each group (Wedel & Kamakura, 2001). This yields segments with a varying responsiveness of the dependent variable to different sub-groups of the independent variables. In addition, the method provides fitted regression models for each of the components. These could be used for examining the effects of features across segments or for providing estimations to new observations.

In supervised learning, the value of a clustering solution is predominantly determined by its contribution to modelling the dependent variable. Since a Gaussian mixture model belongs to the class of unsupervised methods all variables are treated similarly with the absence of pre-existing labels. This implicates its results cannot guarantee a relation to exist between the clusters and the response (Wedel & Kamakura, 2001). In contrast, a mixture of regressions aims at deriving clusters that are related to the responsiveness of a dependent variable. Therefore, its results are likely to be of greater value when integrated into predictive analysis.

In general, a mixture model is characterized by two types of parameters: *(i)* the component weights or mixture proportions π_k and *(ii)* the component specific model parameters θ_k such as the mean and variance for the Gaussian model. Together these form the parameter vector φ_k of the mixture method that is defined as $\varphi_k = \{\pi_k, \theta_k\}$. The model assumes that the data can be described by a mixture of K components with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$. Here, every k^{th} component has its own model that is parameterized by θ_k. If we let $Y = (Y_1, \dots, Y_p)$ be a p-dimensional vector of feature variables, the density of Y can be modeled by a mixture of K components. The conditional distribution of the data is formalized as $h(y \mid \varphi_k) = \sum_{k=1}^{K} \pi_k f\,(y \mid \theta_k)$, where $f(y \mid \theta_k)$ refers to the p-variate density function of the data under component k.

The parameters of the mixture model φ_k can be estimated using the maximum likelihood principle. For a random sample of N observations $\{y_1, \dots, y_N\}$ the log-likelihood function of φ_k is defined as $Log\ L = \sum_{n=1}^{N} \log h(y_n, \varphi_k)$. In general, it is analytically impossible to derive the maximum likelihood estimates of φ_k by differentiating this function (McLachlan &

Basford, 1988). Therefore, the expectation maximization algorithm (EM) as introduced by Dempster, Laird and Rubin in 1977 is commonly used to obtain estimates of the parameters. This algorithm provides a generic approach for calculating the maximum-likelihood estimates in a variety of instances with incomplete data. Since the group membership of observations in the mixture model is unknown, EM provides a practical workaround for obtaining the solution.

To estimate the parameters of a mixture model, EM operates as a two-step algorithm. It iterates between calculating the posterior probabilities of each observation for a fixed set of model parameters and then optimizing the model parameters given these posterior probabilities (McLachlan & Basford, 1988). The posterior probability $\tau_i(y_n \mid \varphi_i)$ is the probability that y_n belongs to the i^{th} component of the mixture and can be expressed using Bayes' Theorem as follows $\tau_i(y_n \mid \varphi_i) = \frac{\pi_i f(y_n \mid \theta_i)}{\sum_{k=1}^{K} \pi_k f(y_n \mid \theta_k)}$. After several iterations, the algorithm converges to a local maximum for a set of $\hat{\varphi}_k$ corresponding to the maximum likelihood solution.

To extent upon a general mixture model, consider the following mixture of regressions $h(y \mid x,\ \varphi_k) = \sum_{k=1}^{K} \pi_k f\ (y \mid x,\ \theta_k)$. Here, y represents a (possibly multivariate) dependent variable that follows the conditional density h related to a vector of independent variables x. If f corresponds to a univariate normal density with mean $\beta_k x$ and variance σ_k^2 such that $\theta_k = (\beta_k x,\ \sigma_k^2)$, the model describes a mixture of standard linear regressions (DeSarbo and Cron, 1988). The log-likelihood for a random sample of observations is given by $Log\ L = \sum_{n=1}^{N} \log h(y_n | x_n, \varphi_k)$. To obtain the maximum likelihood estimates of the model parameters the model relies on the same EM procedure as described earlier.

A central question in every mixture model is to determine the number of components K to include in the mixture. McLachlan and Rathnayake (2014) have written a paper on this subject where they review different methods that are commonly used to answer this question. In general, there are two approaches for choosing K that both rely on comparing the mixture solutions for different number of components. First, an information criteria that is based on some penalized form of the log-likelihood can be consulted such as the Bayesian Information Criterion (BIC). Second, a formal hypothesis test can be performed using the likelihood ratio test. However, since the null distribution of the likelihood ratio test statistic does not meet the standard regularity conditions it may lead to biased results (McLachlan & Rathnayake, 2014).

2.3.4. Multinomial Logistic Regression

The segmentation of new observations cannot be directly inferred since these are related to the unknown price of an impression. Therefore, a new model needs to be trained that assigns each testing observations to one of the derived segments. Since the variable of interest is qualitative, a classification model should be set-up in order to predict this outcome. There are many different types of classification techniques, but this paper adopts the multinomial logistic regression for this task. The reason for choosing this model is because it is one of the most common methods for linear classification and can return a probabilistic classification for new data points (Friedman, Hastie & Tibshirani, 2001).

In its basic form, the logistic regression is used to model the probability of a binary dependent variable to one or more predictor variables. Here, it relies on the logistic function to prevent insensible predictions below zero or above one from occurring as would be the case when using a linear function for modelling this relation (James et al, 2013). Subsequently, maximum likelihood is used to derive the estimated parameters of the regression model.

Mathematically, the logistic function that relates a predictor variable X to the binary dependent variable Y can be expressed as $P(Y = 1|X = x) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$. This yields an S-shaped curve to visualize the relation between X and Y, which always provides estimates of Y that fall within the domain [0,1] regardless of the values of X (James et al, 2013). We can rewrite the logistic function to return the odds of Y as a function of X in the following way $\frac{P(Y = 1|X = x)}{1-P(Y = 1|X = x)} = e^{\beta_0+\beta_1 X}$. By taking the logarithm of both sides we obtain the log-odds or logit of Y, which is expressed as a linear function of X given by $\log\left(\frac{P(Y = 1|X = x)}{1-P(Y = 1|X = x)}\right) = \beta_0 + \beta_1 X$.

To estimate the parameters of the logistic regression the maximum likelihood approach is used (James et al, 2013). For a given sample of n observations of x and y with $i = 1, \dots, n$ the log-likelihood function can be defined as $Log\ L = \prod_{i|y_i=1} p(y_i = 1|x_i) \prod_{i'|y_{i'}=0}\big(1 - p(y_{i'} = 1|x_i)\big)$. By maximizing this function, we can derive the maximum likelihood estimates for the model parameters. This yields values for $\widehat{\beta_0}$ and $\widehat{\beta_1}$ so that the sample is most likely under the model.

The logistic regression can be extended beyond the analysis of binary variables to handle the case of categorical variables with more than two categories (Menard, 2002). When a logistic regression is generalized to solve multi-class problems it is often referred to as a multinomial logistic regression. A common approach for setting up such a model is to nominate one value of the dependent variable as reference category (Menard, 2002). The probability of membership of all other categories is calculated relative to this baseline by defining multiple functions.

If we consider Y to be a multinomial variable with K classes that are related to X, the corresponding log-odds can be modeled as $\log\left(\frac{P(Y=1|X=x)}{P(Y=K|X=x)}\right) = \beta_{1,0} + \beta_{1,1}X,\ldots,$ $\log\left(\frac{P(Y=K-1|X=x)}{P(Y=K|X=x)}\right) = \beta_{K-1,0} + \beta_{K-1,1}X$. Note that Y is modeled using $K-1$ equations and that for $K=2$ it returns to the logistic regression for a binary outcome variable. Estimating the parameters of a multinomial logistic regression proceeds in a similar way as for the regular logistic regression and can be accomplished with the maximum likelihood procedure (Friedman, Hastie & Tibshirani, 2001).

2.3.5. K-Nearest Neighbors

With the surge of data and computational power over the last decades, the field of machine learning has significantly grown. This discipline offers researchers the opportunity to combine elements of computer science and statistics for powerful yet flexible modelling. Its methods have become increasingly popular in various applications including the auction domain. Here, Jank and Shmueli (2010) have applied a K-nearest neighbors algorithm (KNN) that proved very competitive to a linear model – especially in the case of heterogeneous auctions.

KNN is one of the simplest and most well-known techniques in machine learning (James et al., 2013). In contrast to a linear model, KNN belongs to the class of non-parametric models meaning it requires no assumptions regarding the true shape of the regression function. Instead, it relies on the assumption that the response variable of a new observation is strongly related to the response value of the most similar objects in the dataset.

For each new observation x_0, KNN aims to identify the K most similar points in the data represented by N_0. Here, similarity is often measured using the Euclidean distance or a related

distance measure. To control for a different scale of the independent variables and increase model accuracy, the data is often first normalized before calculating the distance measure (Friedman, Hastie & Tibshirani, 2001). After the model has found the K closest points, it will use these to form a prediction of the new observation its response. In a regression, the prediction of the response variable is calculated as $\widehat{y_i} = \frac{1}{K} \sum_{x_i \in N_0} y_i$ with $i = 1, \dots, K$. Intuitively, the target variable of a new observation is estimated by taking the average value of that variable from the K most similar points in the data.

Determining K is an imperative task because its value can influence predictions and is strongly related to the bias-variance trade-off. In statistics, bias is defined as the systematic error of an estimator and can have various problems at its root (James et al., 2013). The variance of a statistical method refers to the degree to which the predicted outcome changes when different training data is used for estimation root (James et al., 2013). For KNN, a small K will have a low bias and a high variance whereas larger values for K typically lead to a lower variance at the cost of a higher bias. In general, the desired value for K will depend on the application and the needs of the end-user.

In this research, the objective is to maximize prediction accuracy of the models. Therefore, K will be chosen as to improve correctness of the model. This can be accomplished by finding a value for K that minimizes the mean absolute error (MAE) of the predictions, which is defined as $MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \widehat{y_i}|$. However, the reference data cannot be used for evaluating the model since it will lead to severe overfitting.

Extracting a validation or hold-out set at the beginning of the analysis can help overcome this, but a more robust solution is to cross-validate the training data for calculating the MAE. Cross-validation (CV) is a refinement of the validation approach that is based on resampling without replacement. A common CV method is k-fold cross-validation, which is based on randomly dividing the data into *k* equal sized groups or folds (James et al., 2013). The first fold serves as validation set with the remaining *k – 1* groups being used for training the model. Subsequently, the hold-out set is used for calculating the error after which the process is repeated *k* times, each time picking a different validation set. The k-fold cross-validated error CV_k that can be

calculated from this method is expressed as $CV_k = \frac{1}{k}\sum_{i=1}^{k} MAE_i$. By plotting this error against a range of values for K, the optimal number of neighbors can be determined.

2.3.6. Artificial Neural Network

The Artificial Neural Network (ANN) is a fundamental component of deep learning – a sub-field of machine learning which models are inspired by the biological brain. Analogous to a brain, the ANN is made up of many simple processors (neurons) that operate in parallel, are widely connected and learn from experience (Specht, 1991). More specifically, the network can learn to perform a task by considering a set of examples without the need of being programmed with task-specific rules. This in contrast to regular machine learning algorithms that do require this type of pre-programming.

ANNs have a proven track-record across a variety of applications. In recent years, many machine learning and pattern-recognition competitions have been won by extensive versions of the model (Schmidhuber, 2015). Furthermore, commercial neural networks are now able to perform with sufficient precision to be used in various organizations such as banks and distribution centers. One of the reasons behind ANN's performance lies in its ability to capture complex patterns that less sophisticated techniques are unable to do. This might be beneficial in the task of grasping the complexity of bidding behavior. For example, Jank and Shmueli (2010) included a neural network in their model comparison for predicting the final price in online auctions where it ranked among the best performing methods.

The neurons constituting the network are structured in different layers as is shown in the schematic diagram in Figure 2 (Nielsen, 2015). The number of neurons for a given layer is equal to n with $i = 1, \dots, n$ and the number of layers in the network equals k with $j = 1, \dots, k$. Consider each neuron to be an object that can process and hold information expressed by a numerical value known as its activation $a_{i,j}$. The first layer of the network is called the input layer and it takes in the information corresponding to the dependent variables. Hence, the number of input neurons is equal to the number of explanatory variables. The last layer is called the output layer, which provides a final prediction of the independent variable. Here, the number of output neurons is equal to the number of target classes in classification or equals one for regression tasks. All neurons in between belong to the hidden layers of the network that

produce most of the model's calculations. The number of hidden layers is somewhat arbitrary and depends on the complexity of the data and available computational power (Nielsen, 2015).

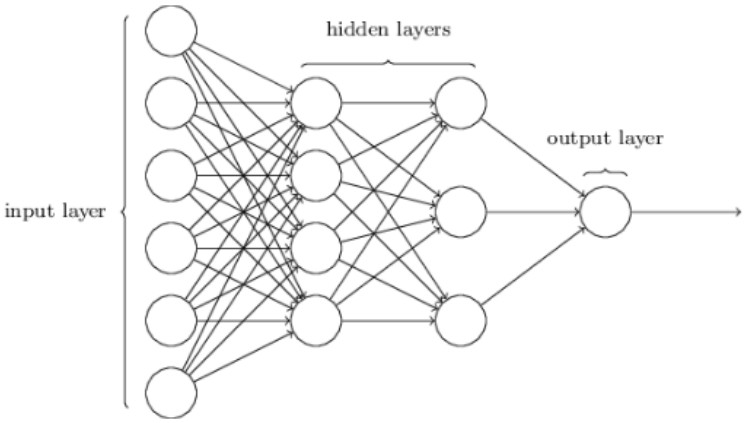
The neurons of neighboring layers are all connected to each other through channels with every channel containing a number known as a weight w. When the activation of a neuron is transmitted to the next neuron it gets multiplied by this weight. In addition, the neurons in the hidden layer contain a value for bias b and some activation function $\sigma(x)$ for processing the input. The bias is added to the input sum of weighted activations to control for when a neuron starts getting meaningfully active. Afterwards, this weighted sum together with the bias gets passed through the activation functions to determine the activation of the neuron. Mathematically, this activation can be expressed as follows $a_{i,j} = \sigma(w_{1,j-1}a_{1,j-1} + w_{2,j-1}a_{2,j-1} + \cdots + w_{n,j-1}a_{n,j-1} + b_{i,j})$. There are different forms of the activation function, but most commonly used is the sigmoid function that is formalized as $\sigma(x) = \frac{1}{1+e^{-x}}$ (Nielsen, 2015). Applying these calculations to all neurons in the network will yield the predicted value of a given observation.

The process being described above is known as forward propagation. On its own, forward propagation cannot learn from the data since no training steps are involved in its sequence. To successfully spot underlying patterns the addition of backward propagation is needed. Starting at the output layer, the error is computed by subtracting the predicted output a from the actual target value $y(x)$, which is formalized in the following cost function $C(w, b) = \frac{1}{2n}\sum |y(x) - a|^2$ where n refers to the total number of training inputs. The magnitude and direction of the error are then transferred back into the network after which the weights and biases get adjusted to reduce the cost.

A solution is derived by using the gradient descent algorithm, which is an optimization technique that iteratively tries to find the direction of the steepest descent in a function using partial derivatives (Nielsen, 2015). To limit calculation time, the stochastic gradient descent technique is commonly used in ANN. Here, the data is first divided into multiple batches before finding the gradient descent and recalculating the weights and biases.

Figure 2

Overview Neural Network Structure



Note. Adapted from *Neural Networks and Deep Learning,* by M.A. Nielsen, 2015.

2.4. Model Selection

In 1987, Box stated: "Essentially, all models are wrong, but some are useful" (p. 424). This contributed to the both technical and philosophical discussion regarding the true definition of a model and how its usefulness should be expressed. A common approach is to try and capture this using a statistical performance metric. In the field of model selection, various methods have been formalized to compare different models' performance for determining the most optimal one (Friedman, Hastie & Tibshirani, 2001).

The types of performance metrics can roughly be broken down into two categories (Breiman, 2001). First, there is the type of measures focusing on goodness-of-fit such as the Akaike information criterion (AIC) and the BIC. Second, there are metrics that rely on prediction accuracy for model validation. Examples include the root mean squared error (RMSE) and the mean absolute error. What measurement is more appropriate depends on the objective of research and the richness of data (Friedman, Hastie & Tibshirani, 2001).

In this paper, both linear and machine learning techniques have been evaluated on their performance. However, since the machine learning models are not linear in their parameters, likelihood measures such as the AIC and BIC cannot be directly calculated (Friedman, Hastie & Tibshirani, 2001). Moreover, predictive power has been considered a central component of this research where data scarcity was not an issue. Therefore, it was chosen to rely on an error-based measure for the model selection procedure.

The RMSE and MAE are both regularly used performance metrics in studies on model evaluation (Chai & Draxler, 2014). Although appearing similar, there are some distinct differences among the two. Most notably is that the RMSE penalizes variance in the error terms by giving a heavier weight to larger absolute values of the error (Chai & Draxler, 2014). In contrast, the MAE applies the same weight to each error regardless of its magnitude and is therefore not sensitive to variability in the errors.

Mathematically, the error measures are defined as $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$ and $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$. When both calculated, the result of the RMSE will always be greater or equal to that of the MAE. The difference between the two terms indicates how much variability is present among the residuals and can be used for statistical inference. Instead of choosing one metric over the other, Chai and Draxler (2014) recommend using a combination of measures to better capture the differences of model performance. For this reason, both metrics were used for evaluating the performance across models.

Calculating the error measure for model selection should exclusively be done for a testing set. This prevents favoring a model that has modeled the noise of the training data known as overfitting (James et al., 2013). In other terms, the testing error serves as a better estimate of model performance in a real-world setting than the training error. Consequently, 33% of the data was withheld at the beginning of this research. After training the model on the remaining 67% of the data, this testing set was used to compute the error measures of the models.

3. RESULTS

3.1. Linear Model

As the first predictive model in this research a multiple regression was performed to capture the linear relationship in the data. More specifically, the price that was paid for an impression on the RTB market was modelled as a linear function of multiple variables including information on the user and the ad slot. An extensive overview of the regression results can be found in Table B2. The R^2 value of 0.283 signaled that roughly one-fourth of the variance in the paying price has been explained by the variance of the independent variables. Here, the

variables with the strongest relation to the price according to their t-values were the floor price, the ad exchanges and the ad sizes.

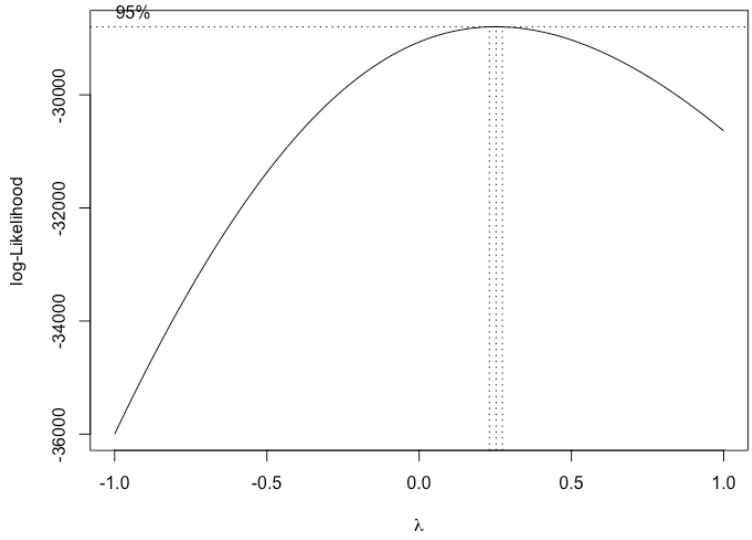
The diagnostics of the regression model have been presented in Figure A1. When inspecting these plots, multiple violations to the assumptions of the linear model could be observed. For example, the upper left plot showed how the residuals of the fitted values differ for the lower and higher prices versus those from the middle segment. This inconsistency indicates that the data may not adhere to a linear relationship which violates a fundamental assumption of the regression model (James et al., 2013). Furthermore, the Q-Q plot in the upper right plot shows that non-normality in the errors is likely present which contradicts another model assumption.

3.2. Box-Cox Transformation

To combat the model violations of the linear regression, a Box-Cox transformation of the dependent variable was performed. By calculating the log-likelihood for different values of lambda in de domain of minus one to one the most optimal power transformation for the linear model was obtained. Figure 3 displays the levels of log-likelihood for every lambda with a 95% confidence interval of the values that have the highest likelihood. Since the value one was not included in this domain, a transformation was required. The optimal value for lambda amounts to 0.26, but since 0.25 lies within the interval – and offers a more intuitive relation – it was set as transformation parameter.

Figure 3

Optimal Lambda from Box-Cox Analysis



After finding the optimal value for lambda, a new linear model was fitted that included a power transformation of the dependent variable as obtained by the Box-Cox analysis. The diagnostics of this model have been presented in Figure A2 and showed multiple improvements in comparison to the first linear model. For instance, the residuals contain less bias and average to zero for the vast number of fitted values. Moreover, the Q-Q plot is more centered towards the diagonal indicating the residuals are more normally distributed. Altogether, these results showed the regression assumptions were better met for the Box-Cox model which signaled that its statistical results should be more statistically sound.

The results of the Box-Cox model have been presented in Table B3. Despite the improvements of the regression diagnostics, the R^2 of the model slightly decreased to 0.272. Hence, transforming the dependent variable did not lead to a better fit of the regression. When inspecting the regression coefficients, there are no remarkable differences in comparison to those of the regular regression. Although the magnitude of the coefficients evidently shrunk due to the transformation of the dependent variable, the t-values and direction of the coefficients have stayed predominantly similar.

3.3. Finite Mixture of Regressions

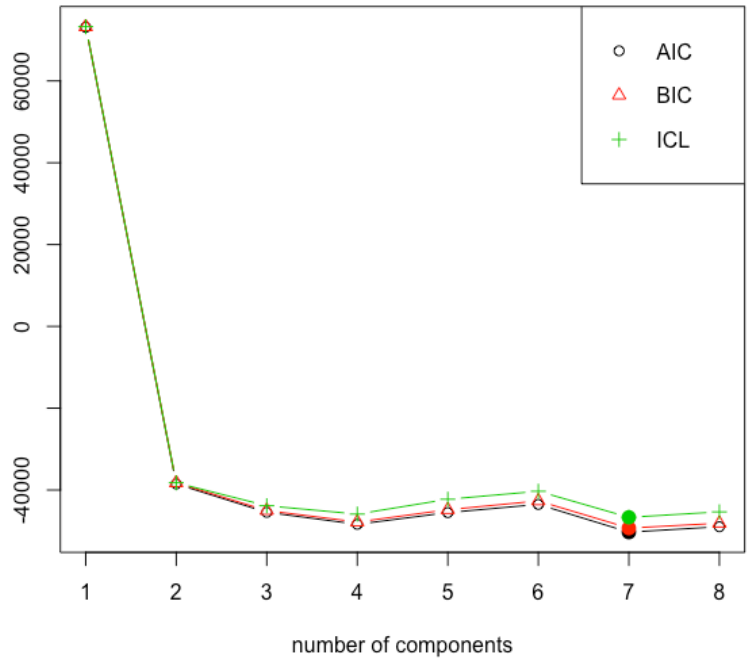
To address market heterogeneity, a market segmentation analysis has been performed using a finite mixture of regressions (FMR). By fitting a mixture of regressions to the training data, the component-based structure of the variables was obtained returning sub-groups with a varying responsiveness of the price towards the other variabels. Afterwards, the regression models corresponding to these segments were used to set-up a predictive model for estimating the price of the testing data.

The number of components to include in the mixture was determined using the BIC. This is a common metric for comparing different mixture solutions (McLachlan & Rathnayake, 2014). In this paper, a negatively oriented version of the BIC was adapted meaning we aim to minimize this value. Since the underlying EM algorithm converges to a local maximum, it has been run repeatedly with different starting values to select the optimal version of each model. In total, every mixture of regressions has been fitted five times for $K = 1, 2, \dots, 8$ components after which the best of each set of models was selected. In Figure 4, the results of this tuning

analysis have been presented. Since the BIC did not vastly improve for mixtures with more than four components it was chosen to set $K = 4$.

Figure 4

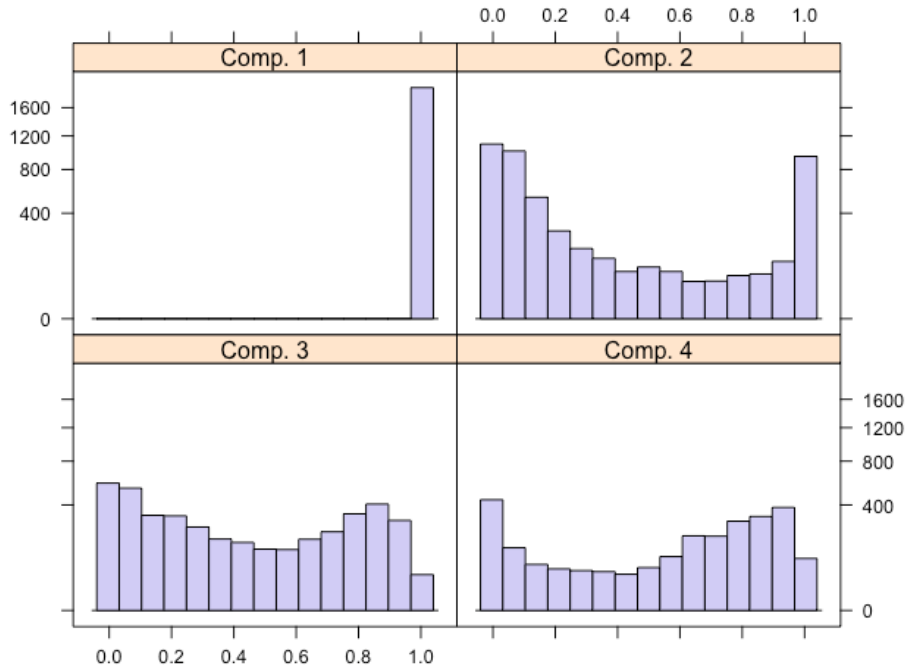
BIC for Different Number of Components of the FMR



The final mixture model used four regressions to model the data each representing a different segment in the market for display advertisements. To visually assess the quality of a clustering solution we can inspect the rootogram of posterior class probabilities over the components (Leisch, 2004). A rootogram is very similar to a histogram, but its height bars are on a rooted scale so low counts become more visible and peaks less emphasized. If the posterior probabilities in a rootogram are concentrated towards 0 and 1 it means the components are well-separated. In Figure 5, the rootograms following from the mixture of regressions in this paper have been presented. This shows that segments 1 and 2 are relatively distinctive while 3 and 4 less so.

Figure 5

Rootogram of Posterior Probabilities $> 1e^{-4}$ *per component*



An overview of the regression coefficients over the components has been provided in Table B4. The variability among these values confirmed that the components picked up different patterns in the data representing a varying responsiveness within the market segments. For example, the first segment modeled observations whose price was almost fully determined by the floor price with a few corrections for the ad exchange, size and visibility. In contrast, the second segment contained impressions whose price was less strongly related to the floor price, but more heavily to the size and region of the ad slot.

To integrate the results of the finite mixture regression into a predictive model both a hard and soft clustering approach have been applied. Hard clustering assumes that each observation belongs strictly to one cluster or segment (Yang, 1993). In line with this assumption, the price of a testing observation was predicted by first selecting the segment it has the highest probability of belonging to as estimated by the classification analysis. Subsequently, the corresponding regression model from the finite mixture of regressions was used to predict its price. In contrast, a soft clustering assumes that each data point can belong to multiple segments with a varying degree of (un)certainty (Yang, 1993). Here, the price of a new observation was

predicted by multiplying the predictions of the four regression models with the estimated probabilities of segment membership from the classification model.

3.4. Multinomial Logistic Regression

Since the paying price of the testing data has been considered unknown, it was impossible to derive its group membership directly using the posterior probabilities of the mixture model. Therefore, a multinomial logistic regression model has been set-up to model the probabilities of a testing observation in belonging to each of four the segments. The model was trained by relating the predictor variables of the training data to the segments that were derived using the finite mixture of regressions. Here, the training observations were first assigned to the segment they had the highest probability of belonging to as expressed by the posterior probabilities. Next, the performance of the logistic regression model was evaluated by comparing the predicted classes with the actual segments in a confusion matrix as presented in Table 1.

Table 1

Confusion Matrix of Multinomial Logistic Regression

		Reference			
		1	2	3	4
Prediction	1	1503	151	237	272
	2	44	475	378	231
	3	107	402	593	463
	4	264	402	501	677

From Table 1 can be seen that the model performs particularly well at predicting the first segment which has a balanced accuracy of 82.3%. For all the other segments the model is a much poorer predictor which have respective balanced accuracies of 60.4%, 57.6% and 59.1%. This, however, comes as no surprise since the rootogram in Figure 5 shows that segment one is clearly most separable when compared to the others segments.

3.5. K-Nearest Neighbors

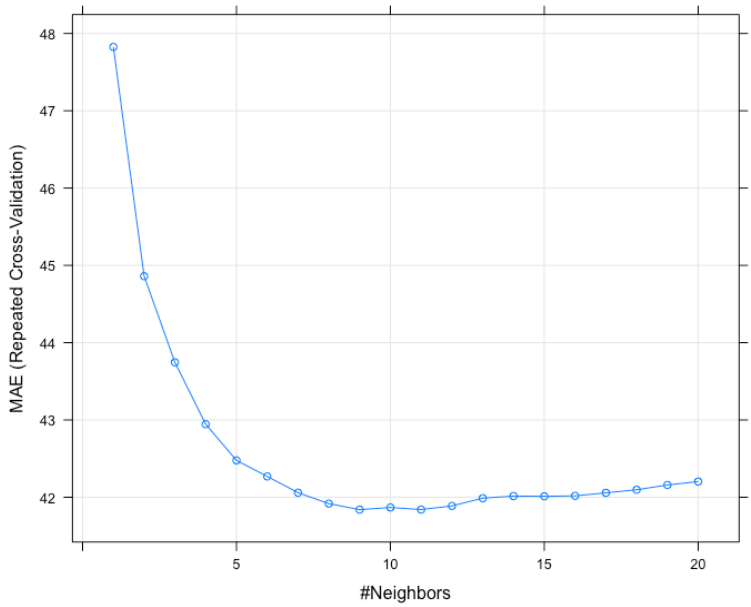
The first machine learning technique used in this research is the KNN model. Contrary to the previous models, KNN is not parameterized by a linear function but it estimates the price of an observation using its similarity to other data points. Since this is measured in distance, the

independent data has been normalized using the min-max transformation prior to the analysis to exclude scale as a factor of influence.

Training the model required one significant training step, which was to determine the number of neighbors that had to be considered for calculating the predicted value. This has been derived by performing a 5-fold cross-validation on the training data and averaging the MAE for different numbers of neighbors across the cross-validated samples. The results of this analysis are presented in Figure 6 that shows the optimal value of K to be equal to nine. Hence, the predicted price of a testing observation is calculated by taking the average value of the nine most similar training observations.

Figure 6

Cross-validated Error for number of Neighbors in KNN



3.6. Artificial Neural Network

For the last predictive model in this research a neural network was trained. Since this method can easily take a long time to calculate, region id was omitted from the set of predictors. This variable did not show a great contribution at the other models and its many categories would impose a great computational expense on the network. In addition, the ANN algorithm only handles numerical inputs. Therefore, the categorical data were converted to numerical variables using a one-hot encoding. Furthermore, a min-max transformation was applied to scale the predictor variables.

To calculate the activations of the neural network, the commonly used sigmoid function was adopted (Nielsen, 2015). The propagation algorithm that has been used for iteratively recalculating the weights was the resilient backpropagation algorithm. This algorithm was designed by Riedmiller and Heinrich Braun in 1992 and has increased efficiency in comparison to regular backpropagation.

The resulting network was set-up using a series of trial and error on the training data since the model is too computationally expensive to cross-validate. Nodes and layers were added until the model would no longer improve or converge. This led to a single-layer neural network with 8 nodes as visualized in Figure A3.

3.7. Model Selection

After having trained the models on the training data their performance was evaluated using the testing set. First, the price of every testing observation was predicted using the different models. Next, the performance measures were calculated by comparing the predictions to the actual price that was paid for an impression. This resulted in the following overview of performance metrics for the model comparison as presented in Table 2.

Table 2

RMSE and MAE of Predictive Models

	RMSE	MAE
Linear Regression model	54.16	41.83
Regression model with Box-Cox transformation	56.01	39.49
Finite Mixture of Regressions with soft clusters	53.32	38.74
Finite Mixture of Regressions with hard clusters	62.55	40.73
K-nearest neighbors model	54.62	40.82
Artificial Neural Network	54.54	40.87

In general, it can be concluded that the models are relatively poor at predicting the price of an advertisement considering that the variable has a mean of 77.2 and a standard deviation of 64.4. This could indicate that the independent variables in this research were not strongly related to

the price that was paid for an impression. Alternatively, the techniques that have been used might be unable to capture the appropriate patterns in the data related to pricing.

Table 2 also shows that the RMSE is consequently higher than the MAE. The differences between these error metrics range from approximately 30% to 50% for the predictive models. This suggests that there is variance present in the errors as the RMSE penalizes larger absolute values of the error term (Chai & Draxler, 2014). The origin of this variance can be observed from the diagnostic plots in Figure A1 and Figure A2, which display the difference between the fitted vs the actual price for the regressions. These figures show that the highest absolute errors are strongly centered around the lower predictions and almost non-occurring for the higher predictions. This indicates that the extreme errors are mostly due to missing high-priced impressions instead of overpricing cheaper ad slots.

The most promising model that was used for predicting the price of an impression was the finite mixture of regressions with soft clusters. It ranked highest on both performance metrics when compared to the other models. This could indicate that controlling for market heterogeneity may improve prediction accuracy of pricing models in the RTB market. However, since the differences in error terms are relatively small the evidence is not conclusive. This could potentially be further increased by training a more sophisticated classification algorithm to more accurately assign new observations to the derived segments.

In any case, the mixture model performed at its best when using a probabilistic or soft clustering of the market segmentation to predict the price. Both error metrics improved for the predictions made with soft clusters when compared to hard clusters. This means that the price of an impression is better described by a mixture of segments than by a single one and the segments should not be interpreted as ground truths.

4. CONCLUSION

In this research, an effort has been made in modelling the paying price of an impressions in the real-time bidding market for display advertisements. After describing the RTB eco-system and introducing the dataset, multiple statistical and machine learning models were trained to predict the price of a digital advertisement slot. Finally, their performance was evaluated by comparing the error measures resulting from predicting prices for a testing set.

The results reveal that the models are moderately accurate in predicting the price of the testing data. This could be explained by two limiting factors. First, the feature data may not be strongly related to the price of an impression. Individual valuations of the customer could vastly differ over the advertisers without general trends. Moreover, it could be possible that advertisers are in possession of more data through data exchanges and cookie data to base their bids on which are not included in this research. Second, the statistical models used in this research might not be able to model the patterns that relate the prices to the advertisement data. If advertisers use sophisticated statistical techniques to determine their valuation more complex patterns in the data might be present.

Another finding of the analysis is that there exists a systematic discrepancy between the two error terms that are used. The RMSE is consistently higher than the MAE, which suggests that variability in the errors is present. This is partly driven by the inability of the models to accurately detect and predict highly priced impressions. The reasons for this obstacle are likely similar to those that generally impede the model performance. Advertisers may use strategic prospecting and retargeting algorithms to determine their bids. When retargeting a customer using cookie data they can start aggressively bidding on the digital ad space of the user. This behavior can be difficult for algorithms to learn and becomes more complex if valuations greatly differ across advertisers.

Incorporating a market segmentation to predict the price of an impression shows a slight improvement compared to the other pricing models. By fitting a mixture of regressions, multiple segments with a varying responsiveness of the price to the predictors can be derived that control for some degree of market heterogeneity. Combined with a classification model that provides a probabilistic segmentation of new observations, this method ranks as best performing method in this study. However, the difference in accuracy to other models is small and overall error rates remain an issue. Therefore, more research is needed to assess the competitiveness of such a stacking model in the RTB market or for different applications with market heterogeneity.

In general, the RTB market for display advertisements shows to be a challenging environment to model. This makes it difficult for a publisher to obtain an accurate understanding of the value of its ad space to the advertising industry. Future research could try to overcome this in various

ways. Adding more personal data may naturally improve predictions, but this might be hard to come by for public use. Furthermore, incorporating a stronger advertiser oriented approach for modelling price may help better detect highly valued impressions. This could be accomplished by including click-through-rates and conversion rates as subcomponents of the prediction model. As a result, overall performance will likely improve and a better understanding of the market can be obtained.

References

Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, *26*(2), 211-243.

Box, G. E., & Draper, N. R. (1987). *Empirical model-building and response surfaces.* John Wiley & Sons.

Breiman, L. (2001). Statistical modeling: the two cultures. *Statistical Science*, 16(3), 199–215.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3), 1247-1250.

DeSarbo, W. S., & Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2), 249-282.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1-22.

eMarketer. (2019, March). *US Digital Ad Spending 2019*. https://www.emarketer.com/content/us-digital-ad-spending-2019

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*.New York: Springer.

Jank, W., & Shmueli, G. (2010). *Modeling online auctions* (Vol. 91). John Wiley & Sons.

Liao, H., Peng, L., Liu, Z., & Shen, X. (2014). iPinYou global rtb bidding algorithm competition dataset. *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 1-6.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: Inference and applications to clustering* (Vol. 38). New York: M. Dekker.

McLachlan, G. J., & Rathnayake, S. (2014). On the number of components in a Gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5), 341-355.

Menard, S. (2002). *Applied logistic regression analysis* (Vol. 106). Sage.

Milgrom, P. R. (2004). Putting auction theory to work. *Cambridge University Press.*

Nielsen, M. A. (2015). Neural networks and deep learning (Vol. 2018). *San Francisco, CA:*

Determination press.

Ren, K., Zhang, W., Chang, K., Rong, Y., Yu, Y., & Wang, J. (2018). Bidding machine: Learning to bid for directly optimizing profits in display advertising. *Ieee Transactions on Knowledge and Data Engineering, 30*(4). 645-659.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *IEEE international conference on neural networks*, 586-591.

Sayedi, A. (2018). Real-time bidding in online display advertising. *Marketing Science, 37*(4), 553-568.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

Specht, D. F. (1991). A general regression neural network. *IEEE transactions on neural networks*, *2*(6), 568-576.

Tucker, C. E. (2014). Social networks, personalized advertising, and privacy controls. *Journal of marketing research*, *51*(5), 546-562.

Tuma, M., & Decker, R. (2013). Finite Mixture Models in Market Segmentation: A Review and Suggestions for Best Practices. *Electronic Journal of Business Research Methods*, *11*(1).

Wang, J., Zhang, W., & Yuan, S. (2017). Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in Information Retrieval*, *11*(4-5), 297-435.

Wedel, M., & Kamakura, W. A. (2012). *Market segmentation: Conceptual and methodological foundations* (Vol. 8). Springer Science & Business Media.

Yuan, S., Abidin, A. Z., Sloan, M., and Wang, J. (2012). Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users. *arXiv preprint arXiv:1206.1754.*

Yang, M. S. (1993). A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11), 1-16.

Zhang, H., Zhang, W., Rong, Y., Ren, K., Li, W., & Wang, J. (2017). Managing risk of bidding in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 581-590.

Zhang, W., Yuan, S., Wang, J., & Shen, X. (2014). Real-time bidding benchmarking with iPinYou dataset. *arXiv preprint arXiv:1407.7073.*

Appendix A. Figures

Figure A1

Model Diagnostics Linear Model

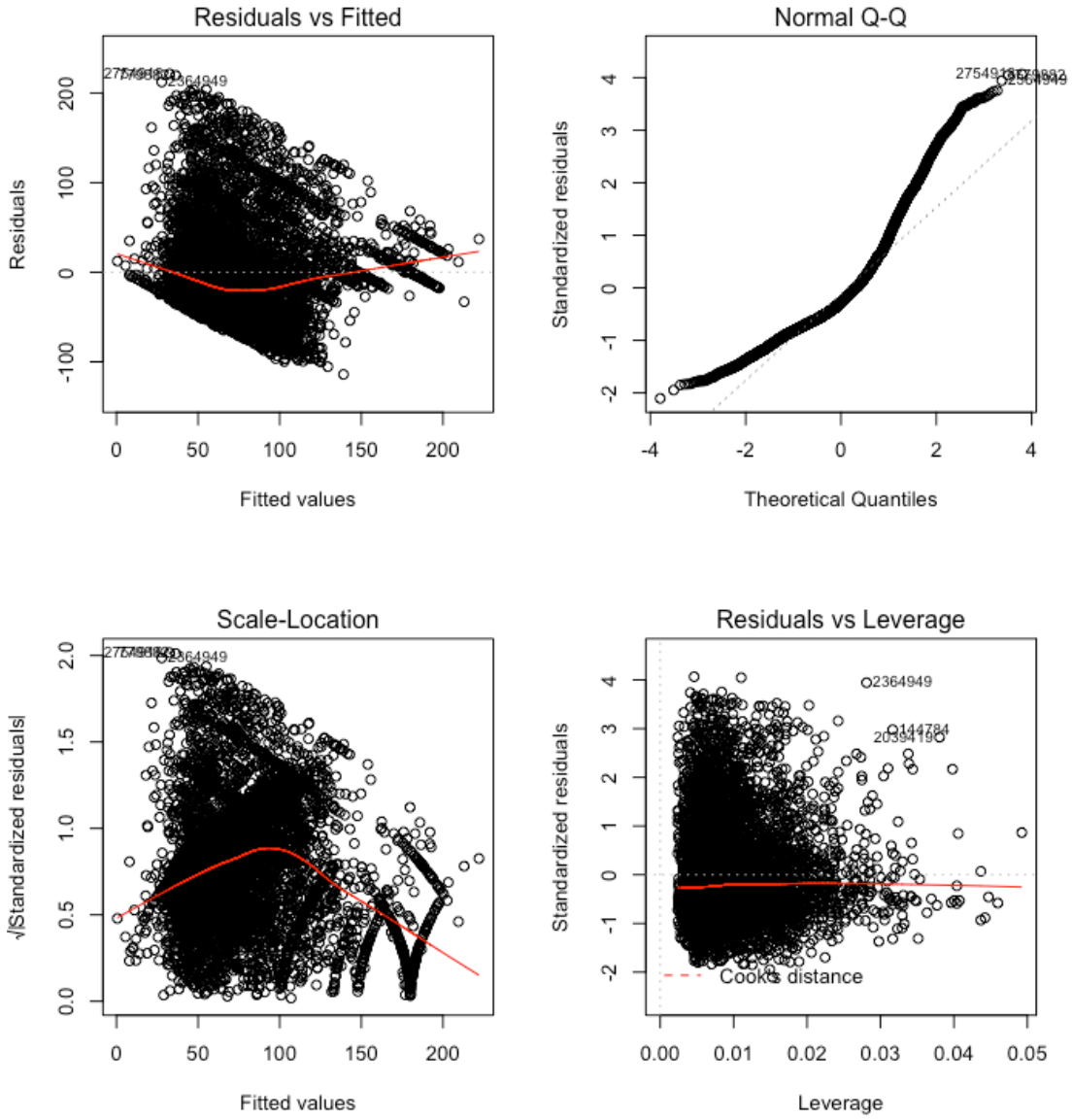


Figure A2

Model Diagnostics Linear Model with Box-Cox Transformation

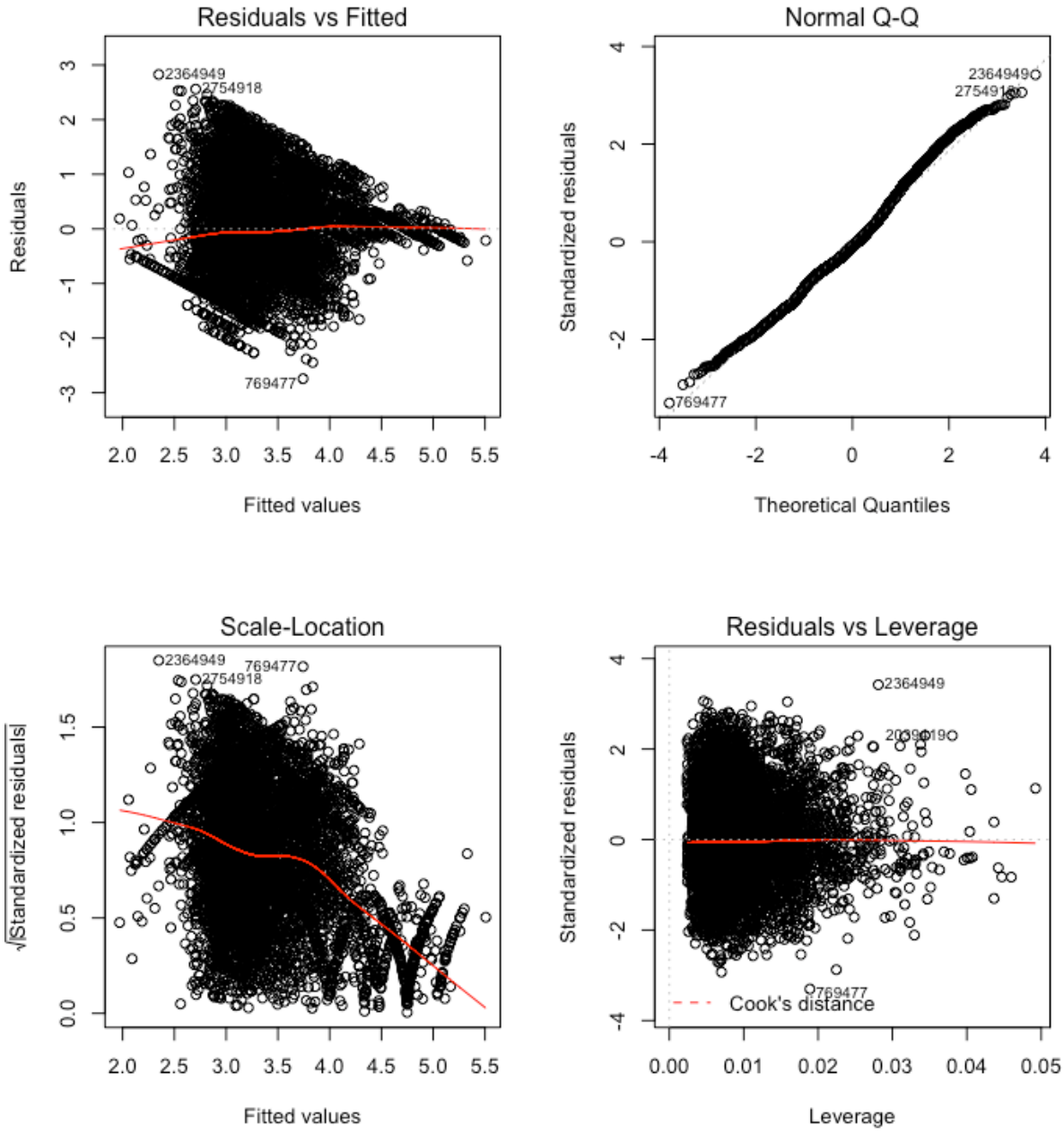
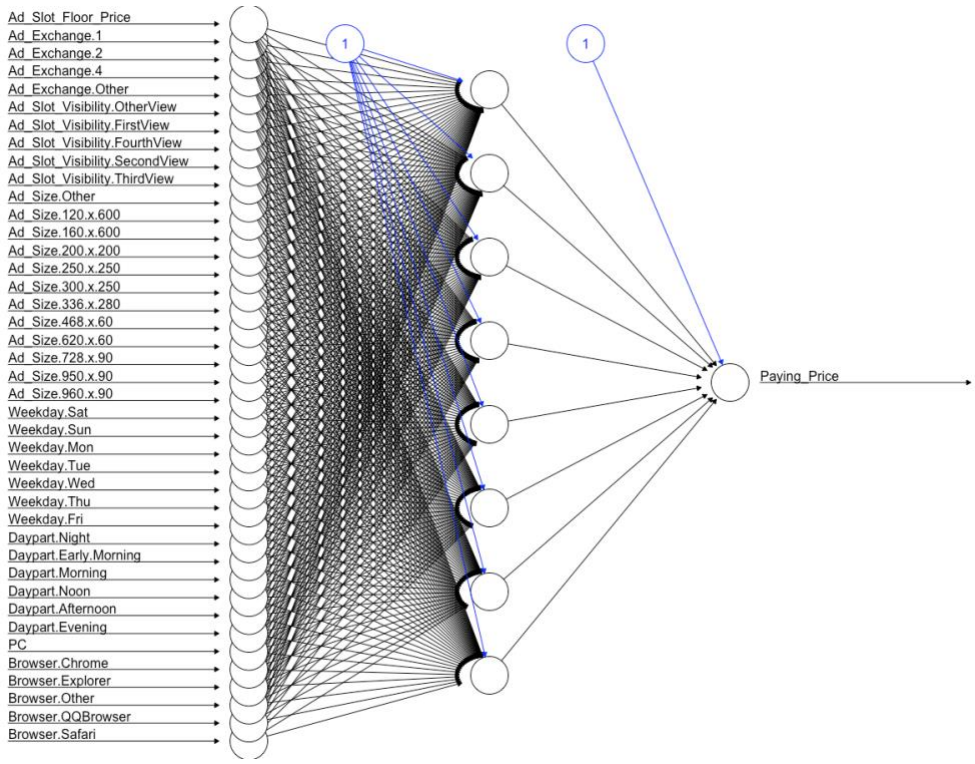


Figure A3

Neural Network Structure for a Single Hidden Layer with 8 Nodes



Appendix B. Tables

Table B1

Overview iPinYou Log Data Format

Col #	Description	Example
1	Bid_ID	5df19fc12e1ea5fd2809a630ced62725
2	Timestamp	20131021211100500
3	Log_Type	1
4	iPinYou_ID	D8NLsb7Wx0D
5	User_Agent	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
6	IP	60.28.101.
7	Region_ID	2
8	City_ID	2
9	Ad_Exchange	4
10	Domain	f8febf1a31b70b3c14ef8338756254e8
11	URL	3d8ad5f8c014bc3dc09861c37788f835
12	Anonymous_URL	null
13	Ad_Slot_ID	9223372032560700000
14	Ad_Slot_Width	960
15	Ad_Slot_Height	90
16	Ad_Slot_Visibility	FirstView
17	Ad_Slot_Format	Na
18	Ad_Slot_Floor_Price	0
19	Creative_ID	10717
20	Bidding_Price	294
21	Paying_Price	20
22	Landing_Page_URL	null
23	Advertiser_ID	2821
24	User_Profile_ID	1,005,713,800,100,590,000,…,000

Table B2

Regression Output Linear Model

	Estimate	Std. Error	t-value	Pr(>|t|)	
(Intercept)	24.01	11.49	2.09	0.04	*
Ad_Slot_Floor_Price	0.58	0.02	31.01	0.00	***
Region_ID1	19.65	5.51	3.56	0.00	***
Region_ID2	7.73	6.96	1.11	0.27	
Region_ID3	11.35	5.42	2.10	0.04	*
Region_ID15	4.14	6.37	0.65	0.52	
Region_ID27	1.79	7.76	0.23	0.82	
Region_ID40	9.75	6.03	1.62	0.11	
Region_ID55	11.56	6.96	1.66	0.10	.
Region_ID65	8.86	6.65	1.33	0.18	
Region_ID79	19.16	6.10	3.14	0.00	**
Region_ID80	12.29	5.17	2.38	0.02	*
Region_ID94	8.76	5.21	1.68	0.09	.
Region_ID106	11.07	5.88	1.88	0.06	.
Region_ID124	9.35	6.04	1.55	0.12	
Region_ID134	10.24	6.94	1.48	0.14	
Region_ID146	17.27	5.18	3.33	0.00	***
Region_ID164	8.94	5.48	1.63	0.10	
Region_ID183	3.33	5.64	0.59	0.56	
Region_ID201	12.77	6.10	2.09	0.04	*
Region_ID216	9.47	4.52	2.09	0.04	*
Region_ID238	6.89	6.45	1.07	0.29	
Region_ID999	0.50	5.31	0.09	0.93	
Region_ID275	10.29	7.25	1.42	0.16	
Region_ID276	10.70	5.72	1.87	0.06	.
Region_ID308	12.17	7.66	1.59	0.11	
Region_ID333	9.63	6.11	1.58	0.12	
Ad_Exchange2	-40.38	3.12	-12.95	0.00	***
Ad_Exchange4	-71.35	3.67	-19.46	0.00	***
Ad_ExchangeOther	-30.41	9.86	-3.08	0.00	**
Ad_Slot_VisibilityFirstView	15.44	2.38	6.49	0.00	***
Ad_Slot_VisibilityFourthView	-7.48	4.33	-1.73	0.08	.
Ad_Slot_VisibilitySecondView	7.09	3.81	1.86	0.06	.
Ad_Slot_VisibilityThirdView	-16.28	4.23	-3.85	0.00	***
coef.Ad_Size120x600	62.62	8.88	7.05	0.00	***
coef.Ad_Size160x600	55.50	9.04	6.14	0.00	***
coef.Ad_Size200x200	37.34	8.29	4.51	0.00	***
coef.Ad_Size250x250	31.27	8.12	3.85	0.00	***
coef.Ad_Size300x250	58.50	7.75	7.55	0.00	***

coef.Ad_Size336x280	46.99	7.96	5.90	0.00	***
coef.Ad_Size468x60	46.98	10.06	4.67	0.00	***
coef.Ad_Size620x60	6.17	9.34	0.66	0.51	
coef.Ad_Size728x90	49.99	7.81	6.40	0.00	***
coef.Ad_Size950x90	34.63	9.44	3.67	0.00	***
coef.Ad_Size960x90	47.49	8.49	5.60	0.00	***
WeekdaySun	-3.42	3.02	-1.14	0.26	
WeekdayMon	-2.27	2.77	-0.82	0.41	
WeekdayTue	5.23	2.50	2.09	0.04	*
WeekdayWed	5.61	3.06	1.83	0.07	.
WeekdayThu	-7.65	3.22	-2.37	0.02	*
WeekdayFri	0.00	2.85	0.00	1.00	
DaypartEarlyMorning	11.71	4.73	2.48	0.01	*
DaypartMorning	13.69	2.93	4.67	0.00	***
DaypartNoon	10.84	3.06	3.54	0.00	***
DaypartAfternoon	13.48	2.43	5.55	0.00	***
DaypartEvening	13.97	2.71	5.15	0.00	***
PC	3.71	6.82	0.55	0.59	
BrowserExplorer	4.04	1.64	2.46	0.01	*
BrowserOther	9.21	2.45	3.77	0.00	***
BrowserQQBrowser	1.43	3.20	0.45	0.65	
BrowserSafari	12.57	6.85	1.83	0.07	.

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.59 on 6639 degrees of freedom

Multiple R-squared: 0.2833,

F-statistic: 43.73 on 60 and 6639 DF, p-value: < 2.2e-16

Table B3

Regression Output Linear Model with Box-Cox Transformation

	Estimate	Std. Error	t-value	Pr(>|t|)	
(Intercept)	2.34	0.17	13.94	0.00	***
Ad_Slot_Floor_Price	0.01	0.00	30.46	0.00	***
Region_ID1	0.28	0.08	3.48	0.00	***
Region_ID2	0.13	0.10	1.29	0.20	
Region_ID3	0.13	0.08	1.65	0.10	.
Region_ID15	0.07	0.09	0.71	0.48	
Region_ID27	0.02	0.11	0.17	0.86	
Region_ID40	0.17	0.09	1.95	0.05	.
Region_ID55	0.16	0.10	1.53	0.13	
Region_ID65	0.12	0.10	1.19	0.23	
Region_ID79	0.31	0.09	3.43	0.00	***
Region_ID80	0.16	0.08	2.14	0.03	*
Region_ID94	0.11	0.08	1.43	0.15	
Region_ID106	0.12	0.09	1.36	0.17	
Region_ID124	0.14	0.09	1.64	0.10	
Region_ID134	0.09	0.10	0.92	0.36	
Region_ID146	0.26	0.08	3.49	0.00	***
Region_ID164	0.14	0.08	1.70	0.09	.
Region_ID183	0.04	0.08	0.48	0.63	
Region_ID201	0.13	0.09	1.43	0.15	
Region_ID216	0.15	0.07	2.23	0.03	*
Region_ID238	0.08	0.09	0.82	0.41	
Region_ID999	-0.03	0.08	-0.41	0.68	
Region_ID275	0.15	0.11	1.43	0.15	
Region_ID276	0.15	0.08	1.86	0.06	.
Region_ID308	0.16	0.11	1.43	0.15	
Region_ID333	0.13	0.09	1.43	0.15	
Ad_Exchange2	-0.62	0.05	-13.57	0.00	***
Ad_Exchange4	-1.03	0.05	-19.19	0.00	***
Ad_ExchangeOther	-0.29	0.14	-2.03	0.04	*
Ad_Slot_VisibilityFirstView	0.25	0.03	7.19	0.00	***
Ad_Slot_VisibilityFourthView	-0.08	0.06	-1.26	0.21	
Ad_Slot_VisibilitySecondView	0.19	0.06	3.47	0.00	***
Ad_Slot_VisibilityThirdView	-0.19	0.06	-3.04	0.00	**
coef.Ad_Size120x600	0.99	0.13	7.64	0.00	***
coef.Ad_Size160x600	0.95	0.13	7.19	0.00	***
coef.Ad_Size200x200	0.71	0.12	5.91	0.00	***
coef.Ad_Size250x250	0.52	0.12	4.43	0.00	***
coef.Ad_Size300x250	1.04	0.11	9.16	0.00	***

coef.Ad_Size336x280	0.83	0.12	7.13	0.00	***
coef.Ad_Size468x60	0.78	0.15	5.34	0.00	***
coef.Ad_Size620x60	0.37	0.14	2.74	0.01	**
coef.Ad_Size728x90	1.01	0.11	8.87	0.00	***
coef.Ad_Size950x90	0.71	0.14	5.12	0.00	***
coef.Ad_Size960x90	0.92	0.12	7.41	0.00	***
WeekdaySun	-0.04	0.04	-1.01	0.31	
WeekdayMon	-0.02	0.04	-0.47	0.64	
WeekdayTue	0.07	0.04	1.86	0.06	.
WeekdayWed	0.06	0.04	1.40	0.16	
WeekdayThu	-0.11	0.05	-2.33	0.02	*
WeekdayFri	0.01	0.04	0.30	0.77	
DaypartEarlyMorning	0.22	0.07	3.14	0.00	**
DaypartMorning	0.22	0.04	5.04	0.00	***
DaypartNoon	0.17	0.04	3.75	0.00	***
DaypartAfternoon	0.21	0.04	5.82	0.00	***
DaypartEvening	0.21	0.04	5.41	0.00	***
PC	0.02	0.10	0.22	0.82	
BrowserExplorer	0.05	0.02	2.10	0.04	*
BrowserOther	0.14	0.04	3.85	0.00	***
BrowserQQBrowser	-0.01	0.05	-0.26	0.79	
BrowserSafari	0.18	0.10	1.80	0.07	.

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7962 on 6639 degrees of freedom

Multiple R-squared: 0.2724,

F-statistic: 41.43 on 60 and 6639 DF, p-value: < 2.2e-16

Table B4

Regression Output Finite Mixture of Regressions

	Comp.1	Comp.2	Comp.3	Comp.4
coef.(Intercept)	0.00	4.59	108.00	6.13
coef.Ad_Slot_Floor_Price	1.00	0.30	0.80	0.75
coef.Region_ID1	0.00	20.79	-1.27	1.19
coef.Region_ID2	0.00	2.07	-8.52	3.89
coef.Region_ID3	0.00	21.61	-0.89	-1.66
coef.Region_ID15	0.00	-19.85	5.99	-0.04
coef.Region_ID27	0.00	-22.73	4.61	2.91
coef.Region_ID40	0.00	10.72	-6.11	8.01
coef.Region_ID55	0.00	20.33	-7.78	-1.11
coef.Region_ID65	0.00	21.89	3.17	6.48
coef.Region_ID79	0.00	22.75	5.94	4.50
coef.Region_ID80	0.00	13.20	6.45	1.08
coef.Region_ID94	0.00	6.47	0.53	-1.74
coef.Region_ID106	0.00	22.13	-9.53	-2.43
coef.Region_ID124	0.00	9.72	1.57	0.28
coef.Region_ID134	0.00	36.11	2.14	-0.47
coef.Region_ID146	0.00	14.53	4.65	0.78
coef.Region_ID164	0.00	-0.22	1.74	0.63
coef.Region_ID183	0.00	2.14	-14.57	0.14
coef.Region_ID201	0.00	19.39	32.93	2.34
coef.Region_ID216	0.00	10.49	2.02	2.61
coef.Region_ID238	0.00	13.35	-4.54	0.23
coef.Region_ID999	0.00	0.41	-15.52	-0.70
coef.Region_ID275	0.00	10.25	7.57	2.65
coef.Region_ID276	0.00	11.28	-2.31	0.60
coef.Region_ID308	0.00	9.66	3.57	-1.44
coef.Region_ID333	0.00	-1.67	2.09	0.95
coef.Ad_Exchange2	-26.00	-30.21	-89.36	4.93
coef.Ad_Exchange4	-25.00	-56.71	-90.21	-4.27
coef.Ad_ExchangeOther	-26.00	-62.07	-86.68	-26.57
coef.Ad_Slot_VisibilityFirstView	1.00	22.65	18.92	6.02
coef.Ad_Slot_VisibilityFourthView	25.00	-56.44	143.17	22.81
coef.Ad_Slot_VisibilitySecondView	1.00	3.81	5.82	10.30
coef.Ad_Slot_VisibilityThirdView	11.00	15.55	0.45	13.63
coef.Ad_Size120x600	0.00	88.03	-73.89	21.95
coef.Ad_Size160x600	0.00	123.11	-72.79	4.02
coef.Ad_Size200x200	0.00	96.09	-87.47	3.86
coef.Ad_Size250x250	0.00	113.67	-76.26	-0.22
coef.Ad_Size300x250	0.00	113.11	-61.92	4.42

coef.Ad_Size336x280	0.00	115.44	-52.70	5.27
coef.Ad_Size468x60	0.00	86.86	-73.07	-2.57
coef.Ad_Size620x60	50.00	93.00	-95.21	28.87
coef.Ad_Size728x90	0.00	85.94	-62.36	9.54
coef.Ad_Size950x90	50.00	69.60	-69.01	150.81
coef.Ad_Size960x90	0.00	71.91	-77.87	2.61
coef.WeekdaySun	0.00	-16.16	-2.50	-1.24
coef.WeekdayMon	0.00	-16.29	-2.30	-2.65
coef.WeekdayTue	0.00	-6.81	-2.69	-1.93
coef.WeekdayWed	0.00	-16.42	1.87	-0.75
coef.WeekdayThu	0.00	-27.29	9.39	-3.33
coef.WeekdayFri	0.00	-15.26	2.99	-0.52
coef.DaypartEarlyMorning	0.00	13.68	6.24	1.55
coef.DaypartMorning	0.00	25.14	5.61	1.03
coef.DaypartNoon	0.00	23.29	7.22	0.82
coef.DaypartAfternoon	0.00	20.71	9.72	2.45
coef.DaypartEvening	0.00	22.83	11.21	3.90
coef.PC	0.00	5.08	4.29	1.24
coef.BrowserExplorer	0.00	3.87	-0.88	-0.25
coef.BrowserOther	0.00	4.81	-2.28	0.63
coef.BrowserQQBrowser	0.00	7.21	10.59	-3.98
coef.BrowserSafari	0.00	26.83	13.53	7.86