A Regression-based Analysis on the Effect of Crowd Management Methods in an Amusement Park

Loes Modderman
STUDENT NUMBER: 2044288

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
MASTER TRACK DATA SCIENCE & SOCIETY , BUSINESS
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Drew Hendrickson
dr. Maryam Alimardani

Permanently Confidential

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
January 2021

Preface

This thesis is written as part of the fulfillment of the master Data Science & Society of Tilburg University. The motivation to do this research comes from my love of amusement parks, with the Efteling having a special place in my heart. I could not let this change go by to do research for the Efteling on crowd management, especially with being it of such great importance within the pandemic that started in 2020. I hope this thesis will give insight into several crowd management methods and their effect in an amusement park.

I want to thank my supervisor, professor Hendrickson, for his guidance and support during the whole process, especially in such trying times. Much thanks also to Abouzar, Jonas, Tamara, and Robbert for providing me with the necessary data and support to do this research. Additionally, I would like to thank my friends and classmates Maartje and Jeroen, and my partner Tieme for supporting me, be it technical or emotional, during the creation of my thesis.

I wish you a lot of reading pleasure.

Loes Modderman

Tilburg, January 15, 2021

A Regression-based Analysis on the Effect of Crowd Management Methods in an Amusement Park

Loes Modderman

Amusement parks deal with crowdedness and managing this crowd almost every day. This crowdedness may cause waiting times in general to get higher which in turn may cause dissatisfied guests (Furnham, Treglown, and Horne 2020). The research field of managing flow and capacity in amusement parks has been studying for several years how to control the crowds using several methods mostly by running simulations (Ahmadi 1997; Cheng et al. 2013; Zhang, Li, and Su 2017; Yuan and Zheng 2018). This study researches the effect of three different crowd management methods on the waiting time of attractions and the crowd distribution in amusement park the Efteling. The three methods are the placement of physical signing across the park, the sending out of push notifications containing information and tips about crowdedness, and a recommendation app for a phone to recommend attractions and restaurants. This research will analyse these effects using data collected from real life. The data that is used for this study is provided by amusement park the Efteling. The results show that none of the crowd management methods have an effect on the waiting times nor the crowd distribution. However, these results may be inaccurate and can be improved by further optimising the prediction models that are used to compute the results.

Keywords: *regression analysis, crowd management, amusement park, confidence interval, effect size*

Contents

1. Introduction

Amusement parks can be very crowded because the majority of guests all want to go to the same attraction at the same time. There are several studies done to investigate how to manage the crowd in amusement parks (Cheng et al. 2013; Yuan and Zheng 2018; Zhang, Li, and Su 2017). In collaboration with amusement park the 'Efteling' I will be researching the effect of two existing crowd management methods and one new method on the waiting times of attraction and on the crowd distribution in this amusement park. The two existing methods consist of physical signing placed throughout the park with alternative routes and push-notifications being send to the guests with information and tips about peak hours throughout the day. The new crowd management method for amusement parks is a personalized recommendation system mobile application that gives guests recommendations about attractions and food venues based on waiting time and distance (Abbaspourghomi 2020). It is currently not known what the effect is of these methods on the crowd distribution and the waiting times in an amusement park.

This thesis is about analysing these effects for a specific amusement park in The Netherlands, the Efteling. The main goals and research questions of this research are as follows:

1. *What are the effects of three different crowd management methods on the waiting time of attractions in an amusement park?*
2. *What are the effects of three different crowd management methods on the crowd distribution in an amusement park?*

A regression-based analysis will be used to study the effect of the above mentioned crowd management methods in this amusement park. This analysis will be conducted by making predictions using a regression model set up with this research. The predictions and actual values are then used to compute confidence intervals and effect sizes to evaluate the effects of the different crowd management methods. The crowd distribution is measured using a crowdedness index as computed by Abbaspourghomi (2020). To create a working model and to answer the main research questions the following subquestions are composed for both research questions:

i. What features can be used for the model and how important are they?

Using a feature selection technique, the available features will be inspected on their correlation with each other and on the dependent variable according to Spearman's correlation and/or Point-Biserial correlation. In this case, features that have a high correlation with the dependent variable are deemed important and features with a high correlation with each other are considered to be less useful for the model.

ii. Which set of transformed and non transformed features performs best in terms of error?

The features to be used in the model, according to the previous subquestion, will be transformed using different transformation techniques. All possible combinations of that feature set with or without a transformation of the data will be used to train a basic regression model and will be evaluated on an RMSE value. Several iterations are done in combination with subquestion iii where in each iteration the model changes based on the results of subquestion iii.

iii. What regression model with which parameters performs best in terms of error?

Different combinations of regression models and hyperparameter choices are calculated using a grid search method to find the best performing model according to an RMSE value. Several iterations are done in combination with subquestion ii where in each iteration the set of features changes based on the results of subquestion ii.

iv. What crowd management method or combination of methods has an effect based on confidence intervals, and what is the magnitude of this effect, based on explained variation?

The best performing model that is chosen, according to the previous subquestion, will be used to analyse which crowd management method or combination gives a different outcome compared to not using that method. This comparison will be made by computing a 95% confidence interval and by calculating the effect size, that will be determined based on the explained variation.

Subquestions ii and iii are both looking at all the possible different combinations and taking the combination with the best result the model for the analysis done in subquestion iv. Question ii and iii could be combined, however that would mean that the amount of combinations would grow immensely larger, which in turn causes immensely larger computation times. To keep these computation times doable for this thesis, I executed question ii and iii separately. I will talk more about the implications of this decision in the discussion section.

These subquestions all have a (sequential) relationship. To answer question iv, question ii and iii need to be known first. And to answer question iii, question ii needs to be known. To answer question ii, question i and iii needs to be clear. A more detailed flowchart of how these processes relate to each other can be found in Appendix A.

There are three factors that motivate the research to be undertaken in this master thesis. First, at the time of writing there is a worldwide pandemic of COVID-19 going on that asks for more social/physical distancing of people everywhere. That is why it is important to employ the best working crowd management methods.

Second, the validation of the effect of these crowd management methods is important to know to make the best use of it in the amusement park.

Third, the research in this thesis will be relevant to expand upon the research that has already been done in the field of managing flow and crowds in amusement parks as will be discussed in Section 2

This thesis is organised as follows. Section 2 gives an overview of the related research for this thesis. The methods to be used are laid out in Section 3. In Section 4, I present the data and experimental setup. The results are shown in Section 5. In Section 6, I write about the interpretation of the results and discuss the findings of this research as well as its limitations. I conclude this thesis in Section 7 and propose a possible direction for further research.

2. Related Works

This section shows the studies already done in relation to this thesis. It discusses the effect of queues on customer satisfaction, the behavioural changes that could happen due to 'nudges', the current practices to evaluate methods for managing flow and crowds, and the recommended process of evaluating intervention effects.

2.1 Effect of queues on customer satisfaction

Analysing what methods influence the waiting times in an amusement park could be useful. According to Davis and Heineke (1998), a customer's reaction to waiting in line can influence a customer's perception of the service delivered. Fink and Gillett (2006) goes one step further and states that customers become more dissatisfied the longer they wait in line. The length of the queue is here the most important factor for how long the customer waits in line (Lu et al. 2013). Following Pruyn and Smidts (1993), such dissatisfaction or irritation as a result of waiting in line seems to affect the satisfaction of the service provided and the perceived friendliness of the service personnel.

Keeping the customers satisfied has great value because it is shown that this has a positive effect on an organisation's profitability and the customer loyalty (Singh 2006). Thus, keeping the waiting times short and therefore the customers more satisfied is important for an organisation's well being.

When looking from a business perspective, this could be a motive to research what kind of methods cause behavioural changes, that may shorten these waiting times, in order to create more customer satisfaction. The following section discusses how behavioural changes may be provoked by small changes, called 'nudges'.

2.2 behavioural changes by nudges

Changes in human choice behaviour could be achieved by implementing small changes in their environment. Such a small change could be a 'nudge', which is a psychologically informed tool that is designed to influence choice behaviour concerning the improvement of health and well-being, but not at the expense of forbidding any options (Lin, Osman, and Ashcroft 2017). The method of nudging has gained popularity in the last years. However, the effectiveness of nudging varies considerably across studies. It might be less effective than is thought and the effectiveness is in part related to the category and context of the nudge (Hummel and Maedche 2019).

A nudge could be presented in several ways, such as a sign with information on it be it physical or digital, or a recommendation being done on a product. Senecal and Nantel (2004) voices that the people who got a product recommendation, selected those products twice as often as people who did not get a recommendation. Where an online recommendation was even more influential compared to traditional recommendations such as 'human experts' and 'other consumers'. The type of the product also had an influence on people following the product recommendations. Another study (Lee and Kwon 2008) also showed that an online shopping recommendation mechanism enhanced consumer's positive purchase intentions and their actual purchases.

Knowing that such methods could have a positive effect on human behaviour this could be taken into account in the current study on the effects of several methods.

2.3 Managing flow and crowds

This section will put into perspective several of the studies that focused on achieving these behavioural changes in the context of managing flow and crowds. For a clear and more in-depth overview, these studies are separated on their application in general situations and their application specific for amusement parks.

Managing flow and crowds in general.

Kolli and Karlapalem (2013) study the occurrence of stampedes in large crowds that are caused by a lack of management of the crowds. They present a multi-agent management simulation system that models the above-described problem. These agents are then used to conduct experiments to manage the crowds at certain moments to avoid the possibility of stampedes occurring. As being said, these experiments are not run in real life, making it difficult to see the actual effect of their studied method.

A study about the possibilities and limitations of ICT measures for crowd management during urban mass events is done by Zomer et al. (2015). They propose that a crowd management system for such situations should consider both the characteristics of the crowd as well as of the urban mass event. However, they state that a limitation for such a crowd management system could be that there is inadequate knowledge on the activity choice behaviour of crowd members. The three methods to be studied in this study are interestingly related to the findings of Zomer et al. (2015). This is because two of the three methods, placement of physical signing and sending out push notifications, only take the mass event (the amusement park) into account. The third method, sending out recommendations, does take both the characteristics of the user and the mass event into account.

Martella et al. (2017) studied the crowd management practices and how technology could play a supportive role in these practices. They examined this through interviews with crowd managers. Their study shows that there is room for more technological support at different stages of the planning and implementation of an event. This may relate to the current research in a way that one of the methods to be analysed is physical and the other two methods are technological.

Managing flow and crowds in amusement parks.

The study of managing flow and crowds in amusement parks is fairly new. The first research in this field was published in 1997 and described an early application of a model-based approach to manage the capacity and flow at amusement parks (Ahmadi 1997). This paper used the daily operations of the park and an analysis of the visitor transition patterns to find optimal settings of a ride's nominal capacity and to develop models that would suggest an optimal route.

Later research in this field focused on achieving the management of theme parks and their crowding problem using agent-based simulation approaches (Cheng et al. 2013; Yuan and Zheng 2018). However, these studies were not tested in the field. Cheng et al. (2013) quantified and modeled the behaviour of visitors. They integrated this in an agent-based simulation where visitor agents are modeled. This simulation is then used to understand the build-up of the crowd and the impacts of various control policies on visitor experience. Yuan and Zheng (2018) develops a method for predicting tourist distribution by using Markov models. To validate if this approach can improve the crowding mitigation, they used an agent-based simulation model. The results of this simulation suggest that their proposed method significantly outperforms other, existing, methods.

Zhang, Li, and Su (2017) took a different approach and studied how visitor movements were affected by certain attractions and spatial layout attributes in an amusement park. This research analysed the visit routes of their respondents in a Chinese amusement park on several days. With this, they could explore the visitors' flow and their attraction choice behaviour. This showed that 10 attraction attributes and spatial layout attributes influenced the behaviour of visitor movements in the Chinese amusement park, Wuhu Fantawild Adventure. This study paved the way for further statistical analysis in this field by transforming complicated spatial patterns into quantitative measures.

Another research studies the effect of different waiting lines on the guest satisfaction (Beloiu and Szekely 2018). There are three queuing systems modeled and evaluated based on guest satisfaction.

The latest study at the time of writing develops a personalized recommendation system for guests to use, to navigate themselves across the park (Abbaspourghomi 2020). This study examined the crowdedness of different places in amusement park the Efteling. This information about the crowdedness is then used to develop an extensive system that provides the current guests with recommendations for their visit in the park. One of the recommendations for the Efteling in response to this research is that it should study the impact of the recommendations on the behaviour and movements of different groups of visitors.

This thesis will follow up on the latest research of Abbaspourghomi (2020) by evaluating the effect of this newly developed recommendation system on the waiting times of attractions and on the crowd distribution in an actual amusement park. Next to this recommendation system method, this thesis will evaluate two other methods to manage the crowd in that same amusement park.

2.4 Evaluating intervention effects

There are several methods to evaluate the effect of an intervention, such as null hypothesis significance testing (NHST), calculating effect sizes, and computing confidence intervals. Gardner and Altman (1986) suggests that, to determine the actual size of a difference, rather than a simple indication of statistical significance, confidence intervals should be computed instead of NHST. Nakagawa and Cuthill (2007) states almost the same as Gardner and Altman (1986) by saying that NHST fails to provide two important pieces of information: the magnitude of an effect and the precision of the estimate of the magnitude of the effect. This research therefore promotes the use of effect size statistics and confidence intervals. Especially using the effect size and confidence intervals together makes it possible to more effectively interpret the relationship within the data than with the use of NHST, regardless of statistical significance.

For this research, the evaluation of the intervention of the crowd management methods will therefore be done using confidence intervals and effect sizes.

3. Methods

In this methods section, all methods that are used in this research are briefly explained in here.

3.1 Feature selection

Feature selection is in this research executed with the use of Spearman's correlation coefficient and Point-Biserial correlation coefficient.

3.1.1 Spearman's correlation coefficient.

Spearman's correlation coefficient can be used to find the relation between two variables in a nonlinear relationship. It is a non-parametric test to find the strength of the association between two numerical variables with a monotonic function, meaning that relationship can either be linear or not. Spearman's correlation coefficient varies between -1 and +1 (Charfaoui 2020).

Spearman's coefficient, r_S, can be calculated for variable u and v with Equation 1. Where n is the number of pairs of associated rankings $u_1, u_2, ..., u_n$ and $v_1, v_2, ..., v_n$ (Fieller, Hartley, and Pearson 1957).

To calculate Spearman's correlation coefficient in Python, the Pandas package (pandas development team 2020) can be applied using the pandas.DataFrame.corr function with the method paramater set to spearman.

$$r_S = 1 - \frac{\sum_{i=1}^{n} (u_i - v_i)^2}{(n^3 - 1)} \tag{1}$$

3.1.2 Point-Biserial correlation coefficient.

The point-biserial correlation coefficient method can be used to find the relation between two variables, were one of the two variables should be continuous and the other dichotomous. The correlation coefficients range from -1, a perfect negative correlation, through zero, no association at all; to +1, a perfect positive correlation. The point-biserial correlation, r_{pb}, is derived from Pearson's correlation method when one of the variables is dichotomous. The value of r_{pb} can be calculated with Equation 2.

$$r_{pb} = \frac{\bar{Y}_1 - \bar{Y}_0}{\bar{s}_y} \sqrt{\frac{N_1 N_0}{N(N-1)}}, \tag{2}$$

where $\bar{Y}_0$ and $\bar{Y}_1$ are means of the continuous observations for both classes of the dichotomous variable, coded 0 and 1 respectively; N_0 and N_1 are the number of observations for both classes coded 0 and 1; N is the total number of observations from all classes; and $\bar{s}_y$ is the standard deviation of all the continuous observations (Kornbrot 2005).

To calculate Point-Biserial correlation coefficient in Python, the Scipy package (Virtanen et al. 2020) can be applied using the scipy.stats.pointbiserialr function.

3.2 Transform data

The data are in this research transformed using a log transformation and a differencing transformation. The features are than evaluated using an error metric.

3.2.1 Log transformation.

A log transformation can be used as a method to tackle skewed data. However, a log transformation of the data is not a guarantee to make the data less skewed and make it a better approximation of the normal distribution (Changyong et al. 2014).

A log transformation is performed by taking the log of the data points. Sometimes it can help to add a constant to the data points before taking the log to account for zero values.

To perform a log transformation in Python, the NumPy package (Harris et al. 2020) is used with the numpy.log function.

3.2.2 Differencing transformation.

Differencing of data is mostly used to take out the trend out of seasonal data. Seasonal data can have a certain trend that comes back every season. This means that the seasonal data is a function over time. It is preferred to work with data that is not a function over time, called stationary data. This can be achieved by taking the seasonal trend out of the data.

Performing a differencing transformation means to subtract the value of the previous season (or data point) from the current value (Ong 2020). If y is the seasonal data with n data points, then the detrended data is given as

$$y^* = \sum_{i=1}^{n} (y_i - y_{i-1})$$

3.2.3 Error metrics.

Error metrics are used to evaluate errors. With an error being the difference between a true value, y, and a predicted value, $\hat{y}$. Summing the errors can be misleading because they may be positive and negative. Therefore, it is important to take the square or absolute value of the errors prior to summing them up. This is what RMSE and MAE do.

RMSE takes the root of the mean of the squared error, see Equation 3. Where n is the length of y and of $\hat{y}$.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2} \quad (3)$$

MAE is another metric to evaluate errors. It takes the mean of the absolute error value, see Equation 4. Where n is the length of y and of $\hat{y}$.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}| \quad (4)$$

CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4
9	2	2	4
10	2	2	4

MAE	RMSE
2.000	2.000

CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

MAE	RMSE
2.000	2.236

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325

Figure 1: An example of MAE and RMSE values where the error variance increases. Figure from (JJ 2016).

Both RMSE and MAE are negatively-oriented scores. This means that a lower value is better than a higher value. A difference between these two metrics is that RMSE gives a relatively higher weight to larger errors, because it squares the errors before averaging them. This means that the MAE value can remain the same, while the RMSE value differs. E.g., when the variance associated with the frequency distribution of the error magnitudes increases, the RMSE also increases, while MAE remains steady. Such an example is showcased in Figure 1. This means that when large errors are much more undesirable than smaller errors, the RMSE metric may be more useful than the MAE metric (JJ 2016).

3.3 Model selection

The model selection process in this research is executed using an SVR model, Ridge regression model, and a Lasso regression model with the help of a grid search algorithm.

3.3.1 Support Vector Regression.

Support Vector Regression (SVR) is a supervised learning approach for regression that is based on Support Vector Machines (SVM). The advantages of using SVR are that the computational complexity of the model does not depend on the dimensionality of the input space (Drucker et al. 1997) and also its capability to generalize is quite good, resulting in high prediction accuracies (Awad and Khanna 2015).
The main hyperparameters that are unique for SVR are ε, and C. Here, ε is used to compute the loss function. The value of ε is added around the SVR function to create a 'tube'. Values within this tube around the SVR function are deemed insenstive to the loss function, L. All values outside this tube will be used for the calculation of the loss for this SVR function. The loss function can be linear or quadratic which has the mathematical formulation as shown in Equation 5 and 6 with $f(x, w)$ being shown in Equation 7. Here, M is used for the order of the polynomial and w is the width of the margin of the SVR function. Using this ε value therefore makes the model more robust because it creates a model that is less sensitive to noisy data. The other hyperparameter mentioned before, C, is a regularization parameter which means it influences the error of the model. A

larger C will result in more weight to minimize the error. A smaller C will result in less weight to minimize the error (Awad and Khanna 2015).

$$L_\varepsilon(y, f(x,w)) = \begin{cases} 0 & |y - f(x,w)| \leq \varepsilon; \\ |y - f(x,w)| - \varepsilon & \text{otherwise,} \end{cases} \tag{5}$$

$$L_\varepsilon(y, f(x,w)) = \begin{cases} 0 & |y - f(x,w)| \leq \varepsilon; \\ (|y - f(x,w)| - \varepsilon)^2 & \text{otherwise} \end{cases} \tag{6}$$

$$f(x,w) = \sum_{i=1}^{M} w_i x^i, x \in \mathbb{R}, w \in \mathbb{R}^M \tag{7}$$

Similar to the usage in SVM, slack variables ξ_+, ξ_- can be added to SVR to take care of outliers. In SVM it is used to create what is known as the soft-margin. In SVR these variables determine how many points are tolerated outside of the tube as mentioned before. The complete SVR function can be found in Equation 8 according to Drucker et al. (1997) and Awad and Khanna (2015).

$$min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_+ + \xi_- \tag{8}$$

To perform SVR in Python, the Scikit Learn package (Pedregosa et al. 2011) can be used for the functions sklearn.svm.SVR and sklearn.svm.LinearSVR. The first function can be set with different kernels to perform a linear, polynomial, rbf, or sigmoid regression. The second function works similar to the first function with a kernel set to linear. However, it should scale better to a large number of samples to make the model run faster. Besides being able to tune the ε and C parameter in these functions, the tolerance, maximum iterations, and degree can also be set. The tolerance applies to the tolerance of the stopping criterion. The maximum iterations is a hard limit on the maximum number of iterations done within the solver. The degree represents the degree of the polynomial kernel function and can therefore only be used when the kernel is set to polynomial. The sklearn.svm.LinearSVR function has one parameter that is not shared with the other function. This is the intercept scaling parameter. This parameter can be used to lessen the effect of regularization on the intercept by increasing the intercept scaling.

3.3.2 Ridge and Lasso regression.

Both the Ridge and the Lasso regression make use of regularizing the coefficient estimates, i.e. shrink the coefficient estimates towards zero. Apparently, the variance of the coefficient estimates can be significantly reduced by shrinking them.

Ridge regression is very similar to a least squares regression, both want to minimize the residual sum of squares (RSS). A ridge regression however adds a second term to the RSS, $\lambda \sum_{i=1}^{p} \beta_i^2$, which is called a shrinkage penalty. The whole cost function is shown in Equation 9. Where $\beta_0, \beta_1, ..., \beta_p$ are the estimates and p the amount of predictors of the model. The relative impact of these two terms on the regression coefficient estimates is being controlled by the tuning parameter, λ. A λ of 0 results in a ridge regression with the same effect a least squares estimate would give. When $\lambda \to \infty$, the ridge regression coefficients estimates will approach zero because the impact of the shrinkage penalty grows (James et al. 2013).

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{i=1}^{p}\beta_i^2 = \text{RSS} + \lambda\sum_{i=1}^{p}\beta_i^2 \tag{9}$$

The lasso regression is an alternative to ridge regression. The cost function to minimize for the lasso regression is given in Equation 10. When comparing Equation 10 and 9, the only difference is that the β_j^2 term in the ridge regression penalty has been replaced with $|\beta_j|$ in the lasso regression penalty. The tuning parameter, λ, in the lasso regression operates the same as in the ridge regression. The only difference is that with a sufficiently large λ in the lasso regression, the coefficient estimates can actually become zero, where the estimates in the ridge regression only go towards zero but never actually become zero (James et al. 2013).

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{i=1}^{p}|\beta_i| = \text{RSS} + \lambda\sum_{i=1}^{p}|\beta_i| \tag{10}$$

To perform a ridge or lasso regression in Python, the Scikit Learn package (Pedregosa et al. 2011) is used with the functions sklearn.linear_model.Ridge and sklearn.linear_model.Lasso. The tuning parameter explained above as λ is in this function incorporated as alpha.

3.3.3 Grid Search.

Grid search is a method to perform hyperparameter optimization. Given the hyperparameters, this algorithm does an exhaustive search over these specified hyperparameter values to find the best estimate of a model.

To execute a grid search in Python, the Scikit Learn package (Pedregosa et al. 2011) may be used with the sklearn.model_selection.GridSearchCV function. As the function name already states, this algorithm performs a cross-validated (CV) grid search.

3.4 Crowd management method evaluation

Each crowd management method is evaluated with the help of making a cross prediction. These predictions are used to compute the confidence intervals and the effect size.

3.4.1 Cross predictions.

Comparing the outcome of an intervention that is used, to the outcome of not using that intervention during the same period, can be challenging when the data is collected during different periods. To still make this comparison, a prediction across two models can be made with one dataset to obtain an estimate of an outcome during the same period as the other outcome. This method is in this research denoted as a cross prediction.

To execute a cross prediction, two models are trained on different datasets. Model one is trained on data collected in a period where there was no intervention. Model two is trained on data collected in a period where there was an intervention. To compare the outcomes of using an intervention or not, an estimation needs to be made of what the outcomes would have been if there would not have been an intervention in period two. Such an estimation can be made by using model one to predict the outcome of data two. These predictions are the estimation of what the outcome would have looked like if there was no intervention used in period two. A diagram of this process is visualized in Figure 2.

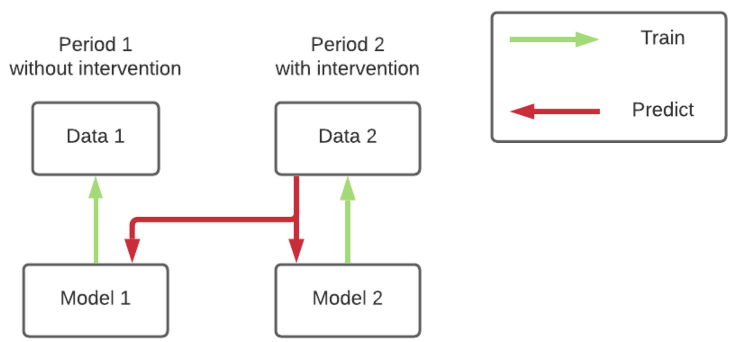


Figure 2: A diagram of a cross prediction process

3.4.2 Effect size using $\Delta\bar{R}^2$.

An effect size quantifies the difference between two groups, e.g. explained variation by two models. To quantify the additional explained variation by one of the models, taking into account all the other predictors, a ΔR^2 can be calculated (Lang 2020).

However, R^2 is an uncorrected estimate and is found to be biased regarding the explained variance in the population (Leach and Henson 2007; Roberts and Henson 2002; Yin and Fan 2001). This bias can be removed by shrinking the effect size, using an adjustment of the formula for R^2, which will result in an adjusted R^2 ($\bar{R}^2$). There are various ways to adjust the formula for R^2. The formula for an $\bar{R}^2$ in this research is based on the Ezekiel index, see Equation 11, with n samples and P predictors (Leach and Henson 2007). As can be seen in this equation, the $\bar{R}^2$ does take the number of predictors into account, resulting in a less biased effect size.

$$\bar{R}^2 = 1 - (1 - R^2)\left(\frac{n-1}{n-P-1}\right) \tag{11}$$

$\Delta\bar{R}^2$ can be used in the same way as ΔR^2 to quantify the additional explained variation of one model over the other (Lang 2020).

4. Experimental setup

This section presents how the research was set up. It discusses the data used in this research and the set up needed to answer each subquestion.

4.1 Data

There are several different datasets used to compute the analysis for this thesis. All code to run the analysis for this research will be written in the programming language Python using version 3.7.3 (van Rossum 1995). This section shall describe the datasets and illustrate what pre-processing steps need to be taken to conduct this research.

4.1.1 Dataset description.

For this research, I used data from eight different sets and sources. A brief overview of the description of all these datasets is made in Table 1 and 2. A description about what the dataset is about and what the variables are can be read below.

Name	Obtained from	Collected since	Frequency	Size (*row x column*)
Weather	KNMI database	2012-01-01	hourly	155246 x 8
Visitors	Efteling database	2011-04-01	daily	3506 x 2
Holidays etc.	Efteling database	2012-01-01	daily	3288 x 15
Push notifications	a colleague	2020-07-11	daily	7 x 1
Recommendation app	Efteling database	2020-10-20	per measurement	4670 x 6
Physical signing	a colleague	2020-08-07	daily	90 x 3
Waiting time	Efteling database	2012-08-20	half hourly	895767 x 5
Crowdedness index	Efteling database	2019-01-01	half hourly	196794 x 3

Table 1: A brief description of the datasets that are used in this research

Weather dataset.

This dataset contains the weather information. The variables consist of information about station number, date, hour of the day, wind speed in 0.1 m/s, temperature in 0.1 degree Celsius measured at a height of 1.5 m, precipitation in 0.1 mm (-1 for < 0.05mm), a binary variable for whether it rained or not, and a binary variable for whether it snowed or not.

Visitors dataset.

A dataset that contains information about the amount of visitors. One variable is the date the data has been collected and the other variable is the amount of visitors that visited the Efteling on that date.

Holidays and other special days dataset.

This dataset is about holidays and certain special periods in a year. The variables hold information about the date of the measurements, the opening and closing hours of the amusement park, if there is a business event or not, if there is a national holiday or not, if this national holiday takes place in the Netherlands, in Belgium, or in Germany, if

Name	Feature description	Feature name
Weather	wind speed in 0.1 m/s temperature in 0.1 °C precipitation in 0.1 mm	WindSpeed Temp RainHourly
Visitors	amount of visitors	cnt
Holidays etc.	business event? national holiday? national holiday in the Netherlands? national holiday in Belgium? national holiday in Germany? school holiday? school holiday in the Netherlands? school holiday in the south of the Netherlands? school holiday in the middle of the Netherlands? school holiday in the north of the Netherlands? school holiday in Belgium? school holiday in Germany? weekend day?	Business event National holiday Netherlands Belgium Germany School holiday Netherlands1 South Middle North Belgium1 Germany1 Weekend
Push notifications	push notification send?	push_send
Recommendation app	number of recommendation done number of accepted recommendations	# rec # accepted rec
Physical signing	signing placed in "Ruigrijk"? signing placed in "Reizenrijk"?	signing_ruig signing_reiz
Waiting time	waiting time corresponding to this attraction? waiting time lag of the waiting time for 1 interval	att_'attraction name' waiting_time waiting_lag1_time
Crowdedness index	crowdedness index corresponding to this squares location? crowdedness index lag of the crowdedness index for 1 interval	square_'square name' crowd_index crowd_lag1_index

Table 2: An overview of de features per dataset and their corresponding feature name. Feature description ending with a questionmark are binary features with a yes/no (1/0) answer.

there is a school holiday or not, if it is a school holiday in Belgium, Germany, and/or the Netherlands, and if this holiday takes place in the south, middle, and/or north of the Netherlands.

Push notification data.

This dataset has knowledge about the days that the push notification method has been tried out. I collected this information via a colleague who had the information in an excel file. The information was gathered between 2020-07-11 and 2020-07-17. It contains only one useful variable, the date. The other information in this file contains the text that was send out using the push-notification displayed in four different languages.

Recommendation app dataset.

This dataset is about the use of the recommendation engine. The variables are the device id, a timestamp of the day and time, the recommended location to go to, a binary variable if the recommendation has been accepted (1) or not (0), the reason for the recommendation, and if there was a request for a new recommendation.

Physical signing data.

This data is about the use of the physical signing in the park. I obtained this information via a colleague who placed these signs. After an interview to obtain the information, this dataset was custom made on the basis of this information. Each row contains the daily information. The first variable is the date. The other variables are binary and show if there was physical signing in place on location A, and if there was physical signing in place on location B. With location A being 'Ruigrijk' and location B being 'Reizenrijk'.

Waiting time dataset.

This dataset is about the waiting times of the attractions. The variables consist of the name of the attraction, the date and time of the measurement, the waiting time, the maximum waiting time of the day up until that measurement, and the minimum waiting time of the day up until that measurement.

Crowdedness index dataset.

This dataset has information on the crowdedness index per square. The variables consist of the name of the square, the date and time of the measurement, and the corresponding crowdedness index.

4.1.2 Pre-processing steps.

All this data need to be made ready for the planned analysis in this research. The following pre-processing steps need to be taken to prepare the data.

To perform the analysis in this thesis, a few extra variables are required that still needed to be created. One of them is the 'Weekend' variable for the Holidays and other special days dataset. This variable contained if a day was a weekend day (1) or not (0).

Another extra variable is the 'waiting_lag1_time' for the Waiting time dataset. This variable is created because the variable 'waiting time' is a time-series. One of the features of a time-series is that it may contain autocorrelation (correlation of a series with its own lags). An autocorrelation plot can be made for a variable, where the results of such a plot can be interpreted as shown in Figure 3. If a time-series would be autocorrelated, it could mean that a lag of this variable would be a good predictor for this variable, in this case the waiting time. To check if this is the case for the waiting time variable, an autocorrelation plot is made per attraction. These are showcased in Appendix B. These plots show that several attractions have a positive autocorrelation for their waiting times, some more than the other. This lead to the decision to create a lag feature of the waiting time with a lag of 1 interval. For this same reason, a lag feature of 1 interval for the crowdedness index is created.

There were also two categorical variables that were transformed to binary variables. These were the name of the attractions from the Waiting time dataset and the name of the squares from the Crowdedness index dataset. These variables were encoded to binary variables to more easily incorporate them into the regression model.

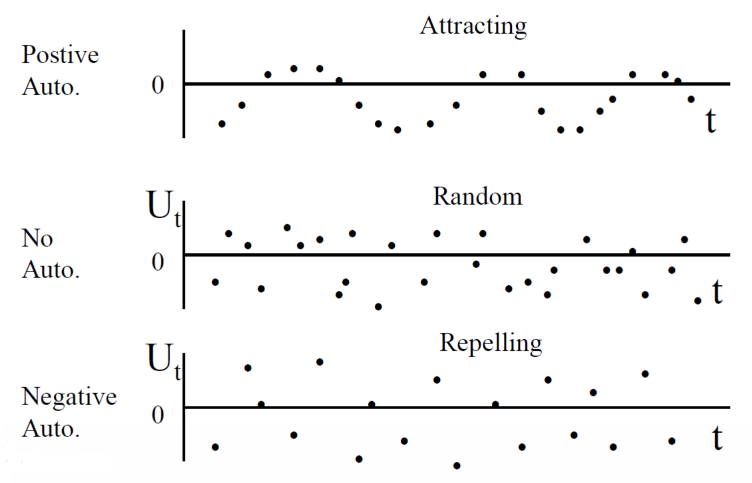


Figure 3: An example of how an autocorrelation plot should be interpreted. Figure from (Ong 2020).

Other pre-processing steps include the handling of missing data, handling of incorrect or unnecessary data, and handling of outliers. Due to the closing of the park necessary for the COVID-19 measures, there is some data missing from the Visitors dataset. These missing data points are imputed in the data set with zero values to make it compatible with the other datasets. The reason to use zero values and not the mean of the visitor count for example is because there weren't any visitors that day. It would be impossible for the park to receive visitors when it is closed. All the other data that has been collected in connection with the park, such as the waiting time, is also zero. Therefore, imputing a mean value instead of zero values would be inconsistent with the other data. Therefore, a zero value is imputed for the visitor count for all the days that the park had to close.

Some data in the Waiting time dataset is incorrect or unnecessary because attraction names are misspelled or because the attractions itself do not have much data points in the whole dataset. This led to the decision to change the misspelled names to correctly spelled names. The attractions that do not have a whole lot of data points in the dataset are deleted, which came down to 35 attractions being deleted.

The waiting time dataset also had some extreme outliers that may influence the analysis. For handling these outliers, they were deleted from the dataset. In total there were 59 data points deleted as extreme outlier.

To make all the datasets compatible with each other, the object type of the date variable needed to be changed to a timestamp for several datasets.

The dataset that had a frequency 'per measurement' (Table 1) was binned per half an hour. This means that every measurement was put in a box containing the measurements of that half hour.

The other datasets that had a daily or hourly frequency also needed to get a half hourly frequency. This was achieved by replicating the data so that it corresponded to every half hour instead of every day or hour.

As a last step, I have merged all datasets together on their dates and times to create one final dataset. Due to the set up of the three datasets containing the methods and due to the COVID-19 pandemic, this merged dataset contains only data from 2020-05-20 to 2020-11-05.

4.2 Experimental procedure

An extensive explanation of the setup for the execution of each subquestion is written down here. The sections below each correspond with the subquestions mentioned in the Introduction in the same order as they are mentioned are mentioned here.

4.2.1 Feature selection.

The features that are initially used for predicting the waiting times are the visitor counts per half an hour, the weather data per half an hour, the information about holidays and special periods per half an hour, the name of the attractions, and the lag of half an hour of the waiting times per attraction. For the prediction of the crowdedness index, the last two features differ. Instead of the name of the attractions, the name of the squares is used and instead of the lag of the waiting times, the lag of the crowdedness index of half an hour is applied. In both cases, the weather feature is actually three different features covering the temperature, wind speed, and hourly rain. The information about holidays and special periods consists of 13 features covering if it is a national holiday (NH) in general, in the Netherlands, in Belgium, and in Germany, if it is a school holiday (SH) in general, in the Netherlands, in Belgium, in Germany, and in the different region in the Netherlands (South, Middle, North), if it is a business event, and if it is a weekend day.

To do a filtering method with numerical data on a numerical predictor, a Spearman's correlation coefficient is used for ranking the features on their correlation with the dependent variable and for finding the correlation between the features with the help of a correlation matrix. Another filtering method is used for the binary features on a numerical predictor, the point-biserial correlation method. This method is used to rank the binary features on their correlation with the dependent variable. For finding the correlation between the binary features, again Spearman's correlation is used to create a correlation matrix.

The features in both rankings, Spearman's correlation and point-biserial correlation, are selected based on the measure of correlation. A threshold set at 0.1 decides if the features will stay or not, where a feature with a measure > 0.1 or a measure < -0.1 stays and a measure between -0.1 and 0.1 is discarded.

The features that are left are the features with a minimum correlation of 0.1 or a maximum of -0.1 with the dependent variable. These features are used to calculate the Spearman's correlation between them, for numerical and binary features. The reason to use Spearman's method for binary features will be discussed later. The features that are correlated with each other will be determined based on their correlation coefficient. A coefficient > 0.6 is flagged as highly correlated in this case. For all the features that are highly correlated with each other, only one is used. That one feature is picked based on its correlation measure with the dependent variable. The feature with the highest measure with the dependent variable is chosen among all features that are highly correlated.

As mentioned before, Spearman's method is also used to calculate the correlation between binary features. This method is chosen for these type of data even though it is known that the assumptions for this method are not met. The reason to calculate the correlation between all the binary features at all is because it is possible for these features that they are highly the same, e.g. the school holidays in the Netherlands will likely have much in common with the school holidays in Belgium or Germany. If multiple features all say mostly the same, it would be unnecessary to keep them all. For that reason it was necessary to find the correlation between these features.

A first thought was to calculate the correlation between binary features also with the point-biserial method, however this method is for calculation between a binary and a continuous variable and not between two binary variables. Therefore, the assumption is made that these binary features are continuous for the time being. This would make it possible to compute Spearman's correlation method on only the binary features, that are assumed to be continuous, with each other to see which binary features are alike.

4.2.2 Transform data.

To find the best set of transformed and non-transformed features, several of the features selected need to be transformed using different transformation techniques. Only the continuous features are transformed using the log transformation and the differencing transformation as explained in Section 3.2.1 and 3.2.2. Finding the best set of these features is an iterative process together with finding the best performing model. This iterative process is done for the model that predicts the waiting time and for the model that predicts the crowdedness index. This results in two different sets of transformed and non-transformed features, one each for predicting the waiting time and for predicting the crowdedness index.

For the continuous features and their transformed equivalents, all possible combinations are sought out. E.g. feature A has a log and a differencing transformation. This means that for feature A, it is possible to use the original feature, the log transformed feature, or the differencing transformed feature. In the case of having two features, A and B, with both three possibilities (original and two transformations), there are $3^2 = 9$ possible combinations for a set with transformed and non transformed features. The set of categorical features are later added to each unique combination, because the categorical features are not being transformed.

Each unique combination of features are used to train a simple Linear SVR model on. The combination of features that trains a model with the lowest RMSE is the best performing set of features for that iteration. The reason to base this choice, and further choices, on RMSE is because larger errors are much more undesirable than smaller errors, as explained in Section 3.2.3. For the next iterations, a model is picked based on the best performing model according to Section 4.2.3, to find again the combination of features that trains a model with the lowest RMSE. These iterations go on until the input model for the transformed feature selection is the same as the output model from the model selection of Section 4.2.3.

4.2.3 Model selection.

The combinations used for the grid search algorithm are divided into several categories. The model category exists of an SVR model using a linear kernel, a Ridge regression model, and a Lasso regression model. There are several more kernels for the SVR model that are not used, such as the polynomial kernel, the rbf kernel, and the sigmoid kernel.

These kernels are specifically not taken into account for the grid search model. The reason for this is that, during the experimentation phase of this research, these models did not converge. Not even when grid search was performed on them, and therefore the computation time became enormously long and the models gave error values back that where always much higher than the first mentioned three models that were used.

The category of the hyperparameters contains six hyperparameter sets. The values that these sets contain are in Table 3 displayed.

Hyperparameter	Set
Tolerance	$\langle$1e-5, 1e-4, 1e-3, 1e-2, 1e-1$\rangle$
Maximum Iterations	$\langle 1000, 2000\rangle$
ε (epsilon)	$\langle 0, 1, 2, 3, 4, 5\rangle$
C	$\langle 1, 5, 10, 100\rangle$
Intercept Scaling	$\langle 1, 2, 3, 4, 5\rangle$
α (alpha)	$\langle 1, 2, 3, 4, 5\rangle$

Table 3: An overview on the values in each hyperparameter set. Not all hyperparameters can be used for every model type.

The grid search algorithm computes for a given model all possible combinations for the hyperparameters and outputs the hyperparameter choices for the best performing model. Given three different models, this grid search outputs three best model setups, one for each model. The model setup that has the lowest RMSE is further used in this research. This process of doing grid search is done both for the model predicting the waiting time and for the model predicting the crowdedness index.

4.2.4 Crowd management method evaluation.

The models that will be used to evaluate the methods, are the models that are performing the best according to the results that will follow Section 4.2.3. These results can be found in Section 5.2. There will likely be two different models with each having a different formed set of features for the prediction of the waiting times and the prediction of the crowdedness index.

To incorporate the different methods as features, they are added to the feature set that is shaped according to the results that will follow Section 4.2.2. This results in several additional and different feature sets that are displayed in Table 4. The methods are for simplicity called method sign, push, and rec. With "sign" being the placement of physical signing across the park (sign A for signing placed in "Ruigrijk" and sign B for signing placed in "Reizenrijk"), "push" being the push notifications being send out, and "rec" being the usage of the recommendation app. The period of execution of some of these methods overlap with each other, which is why some of the features of other methods are incorporated into the feature set of a different method. The execution of the recommendation app overlaps with the signing being placed in "Ruigrijk". The signing placed in "Reizenrijk" overlaps with both the execution of the recommendation engine and the signing placed in "Ruigrijk". The implication of these overlaps is discussed further in the Discussion.

For each method, a cross prediction is made to compare its outcome to an outcome without the intervention of that method during the same period. Each method occurred in a different period and had a different time-span, with some periods even overlapping each other. This causes that the model that is trained on data without the intervention

	Feature set name				
Features	Base	sign A	sign B	push	rec
feature set	✓	✓	✓	✓	✓
signing_ruig		✓	✓		✓
signing_reiz			✓		
push_send				✓	
# rec			✓		✓
# accepted rec			✓		✓

Table 4: An overview of the features that are contained by each feature set

of a method to be different for every method. Executing this cross prediction results in a prediction with and without the use of a method in the same period for every unique method.

To compare these prediction with and without a method and check whether or not it is likely that the use of a method actually made a difference, a 95% confidence interval is calculated for the mean difference between these predictions. This confidence interval is computed for every method. The predictions or actual outcomes when using a method are in this research always subtracted from the prediction of not using a method. This would mean that a positive confidence interval indicates that not using the method has a 95% confidence of having a larger predicted outcome compared to using the method. A negative confidence interval would mean that not using the method has a smaller predicted outcome. A confidence interval that includes zero would mean that it is not expected that there is a difference between the two outcomes.

To check the outcome of the confidence intervals an effect size for each method can be computed using an adjusted R^2 ($\bar{R}^2$) value instead of the usual R^2 to take the large amount of features into account. To calculate the effect size, the $\Delta\bar{R}^2$ of the two predictions for each method is taken, where one is the prediction on a period where there was no method applied and the other was in a period where that method was applied. The models that make these predictions are fitted using a cross-validation technique using five folds. The cross-validation is used to account for the smaller amounts of data that are caused by selecting specific time periods.

5. Results

The outcomes of the experimental setups as explained in Section 4 are displayed in this section. Each result section corresponds with the subquestions as mentioned in the Introduction. One particular results section, Section 5.2, shows the results for both subquestions ii and iii.

5.1 Feature selection

The methods mentioned in Section 4.2.1 are used to create the results shown in this subsection for the two models that predict the waiting times and the crowdedness index.

Model that predicts waiting time.

Table 5 reveals the ranking order corresponding to Spearman's correlation coefficient, denoted by r_S, for all the continuous features. This table shows that the lag of the waiting time has the highest Spearman's correlation coefficient compared to all the other continuous features used. Table 6 shows the correlation based on the point-biserial method, denoted by r_{pb}, for each binary feature with the dependent variable, in descending order.

Features[1]	r_S
waiting_lag1_time	**0.9145**
cnt	**0.4801**
Temp	**0.2648**
WindSpeed	0.0792
RainHourly	-0.0697

Table 5: Spearman's rank correlation coefficients for waiting time model

The continuous features that remain, are the features with a Spearman's correlation higher than 0.1 or lower than -0.1. This means that the remaining continuous features are 'waiting_lag1_time', 'visitor count', and 'Temp'. For these three features, the Spearman's correlation between them is shown in Figure 4. None of these features were highly correlated with each other (correlation coefficient > 0.6). Therefore, none will be discarded.

The binary features that remain, are the features with a Point-Biserial correlation higher than 0.1 or lower than -0.1. According to Table 6, the features corresponding to the bold-faced r_{pb} values (they are > 0.1 or < -0.1) are the remaining binary features. For these remaining binary features, the Spearman's correlation between them is computed as explained in Section 4.2.1 and is shown in Figure 5.

Several of the remaining binary features are highly correlated with each other, because they have a correlation coefficient greater than 0.6 with each other. The feature with the highest point-biserial correlation with the dependent variable from this subset of highly correlated features, is 'South'. This feature stays, while the other highly correlated features are discarded. With all the remaining binary features, another correlation matrix is made to check whether no highly correlated features are missed. This correlation matrix is visualized in Figure 6 and shows that none of the remaining binary features are highly correlated with each other.

1 See Table 2 for an overview of the features.

Features[1]	r_{pb}	Features (continued)	r_{pb}
att_Baron 1898	**0.3521**	att_Villa Volta	0.0229
South	**0.2927**	att_Carnaval Festival	-0.0005
att_Joris en de Draak	**0.2887**	National holiday	-0.0042
att_De Vliegende Hollander	**0.2717**	att_Droomvlucht	-0.0358
Netherlands1	**0.2585**	att_Halve Maen	-0.0377
Middle	**0.2563**	att_Volk van Laaf (Monorail)	-0.0484
School holiday	**0.2462**	Germany	-0.0597
Germany1	**0.2459**	att_Fata Morgana	-0.0745
Belgium1	**0.2335**	att_Pirana	-0.078
North	**0.2058**	att_Oude Tuffer	-0.088
Weekend	**0.1175**	att_Vogelrok	-0.0991
att_Python	0.0885	att_Pagode	**-0.1806**
Belgium	0.0436	att_Polka Marina	**-0.2004**
att_Symbolica	0.0331	att_Monsieur Cannibale	**-0.2137**
Netherlands	0.0235	Business Event	NaN

Table 6: Point-Biserial correlation coefficients for waiting time model

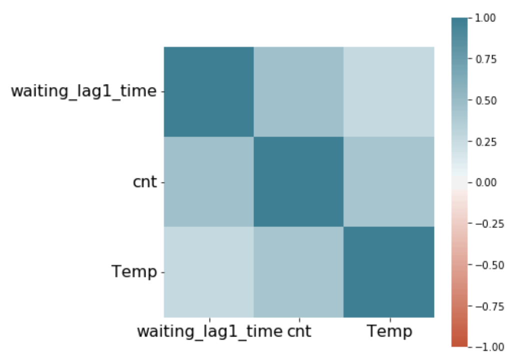


Figure 4: Spearman's correlation matrix for all remaining continuous features belonging to the waiting time model

The main results of this section are that the continuous features that have a correlation above 0.1 or below -0.1 with the dependent variable and are not correlated with each other are 'waiting_lag1_time', 'visitor count', and 'Temp'. For the binary features, these are the 'att_Baron 1898', 'att_Joris en de Draak', 'att_De Vliegende Hollander', 'Weekend', 'att_Pagode', 'att_Polka Marina', 'att_Monsieur Cannibale', and 'South'.

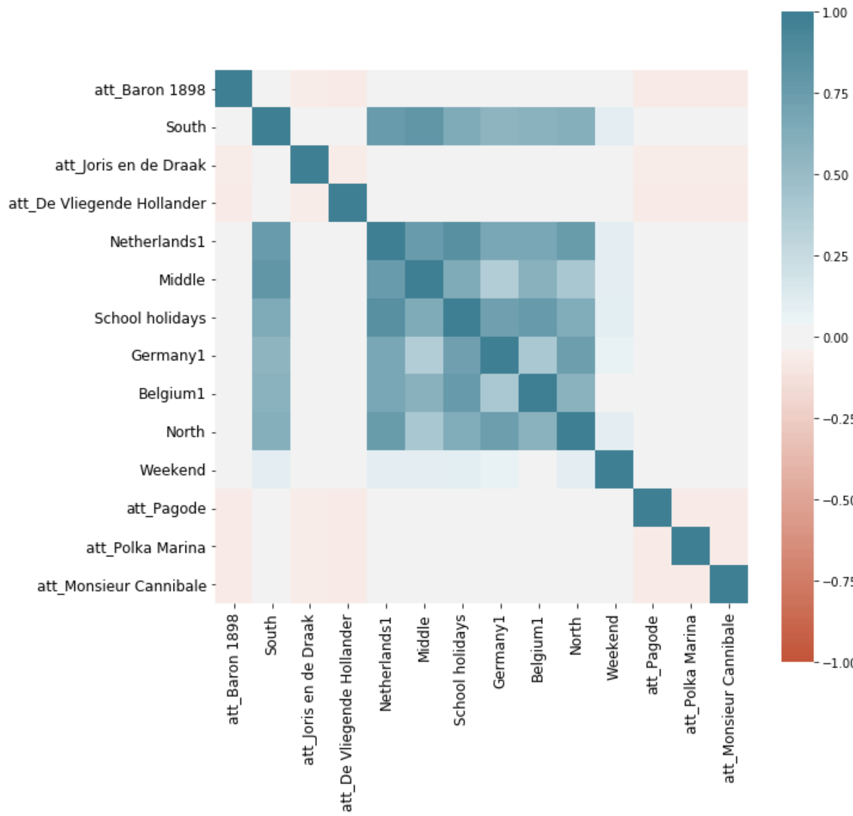


Figure 5: Spearman's correlation matrix for all remaining binary features belonging to the waiting time model

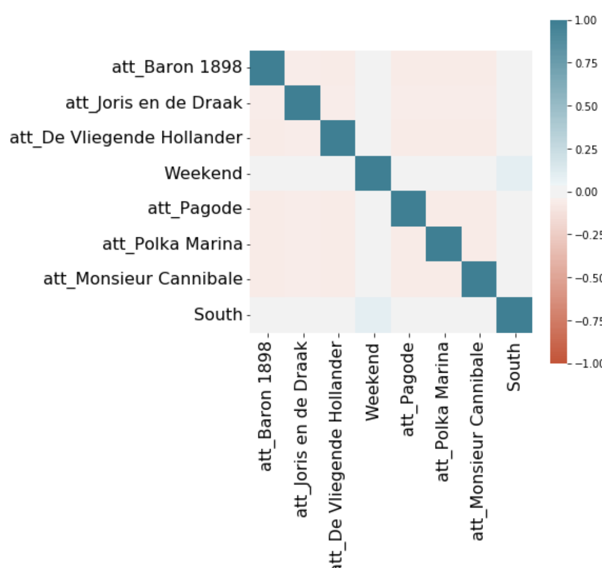


Figure 6: Spearman's correlation matrix for all, non correlated, remaining binary features belong to the waiting time model

Model that predicts crowdedness index.

Table 7 reveals the ranking order corresponding to Spearman's correlation coefficient for all continuous features. The lag of the crowdedness index has the highest Spearman's correlation coefficient according to this table. The correlation for the binary features, based on the point-biserial method, is shown in Table 8 in descending order.

Features[2]	r_S
crowd_lag1_index	**0.984**
cnt	0.0825
Temp	0.0494
WindSpeed	0.0017
RainHourly	-0.0079

Table 7: Spearman's rank correlation coefficients for crowdedness index model

Features[2]	r_{pb}	Features (continued)	r_{pb}
square_Separate location 7	**0.4599**	Netherlands	0.0086
square_Separate location 3	**0.4444**	Germany	0.0008
square_Separate location 6	**0.4087**	square_Vliegende Hollander Plein	-0.0636
Germany1	0.0608	square_Symbolica plein	-0.0681
South	0.0586	square_Python plein	**-0.2006**
Netherlands1	0.0564	square_Anton Pieckplein	**-0.2606**
School holiday	0.0526	square_Aquanura plein	**-0.2606**
North	0.0491	square_Max & Moritzplein	**-0.2606**
Belgium1	0.0452	square_Sprookjesbos	**-0.2606**
square_Droomvluchtplein	0.0433	Business Event	NaN
Middle	0.0414		
Weekend	0.0283		
square_Carnaval Festivalplein	0.0185		
Belgium	0.0164		
National holiday	0.0113		

Table 8: Point-Biserial correlation coefficients for crowdedness index model

The continuous feature that remains, is the 'crowd_lag1_index' because this is the only feature that has a Spearman's correlation higher than 0.1 (or lower than -0.1). Because there is only one continuous feature remaining, it is not possible to test the correlation between features.

The binary features that remain are the bold-faced r_{pb} values in Table 8 because they have a r_{pb} value higher than 0.1 or lower than -0.1. For these remaining binary features, the Spearman's correlation between them is computed and is shown in Figure 7.

The main results of this section are that the continuous feature that remains is 'crowd_lag1_index' and the remaining non correlated binary features are 'square_Separate location 7', 'square_Separate location 3', 'square_Separate location 6', 'square_Python plein', 'square_Anton Pieckplein', 'square_Aquanuara plein', 'square_Max & Moritzplein', and 'square_Sprookjesbos'.

2 See Table 2 for an overview of the features.

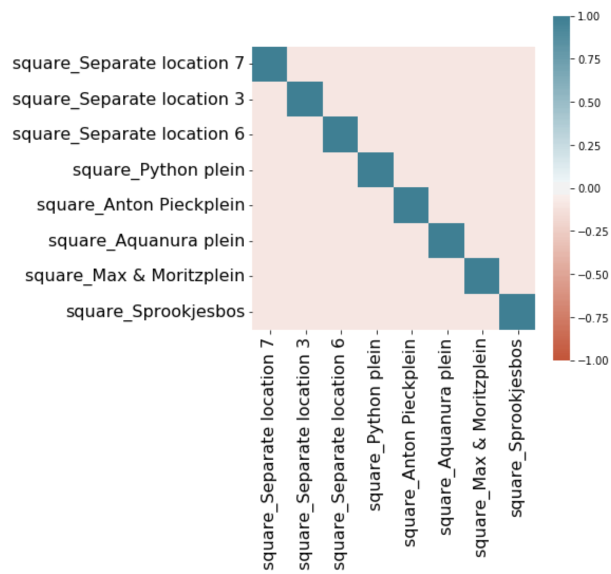


Figure 7: Spearman's correlation matrix for all remaining binary features belonging to the crowdedness index model

5.2 Selection of transformed features and model

Of the continuous features that were non correlated with each other a transformation is made. For these set of features and their transformations, all possible combinations are made

Model that predicts waiting time.

The continuous features are transformed using a log transformation and a differencing transformation. The number of combinations then comes down to 729 for the model that predicts the waiting time. After each of these combinations, the categorical non correlated features are added to create a complete feature set with a unique combination. All combinations are tested on a simple linear SVR model at the start. The outcome over the iterations is shown in Table 9. At the start, a simple Linear SVR model is used. This resulted in a set of features that are then used to select the best model using grid search. This model differs from the initial model and therefore another iteration is done were the model found with grid search is used as a model to select a set of transformed and non-transformed features. This process stopped after two iterations when the input and output model were equal. From Table 9 it can be seen that one transformed feature is selected in the process. The model that used these selected features with the lowest error metric is a ridge regression model with an alpha of 1, a maximum number of iterations of 1000 and a tolerance set to 1e-5.

Iteration	Model for feature selection	Features selected (incl. DV)	Model found with grid search
1	Linear SVR: dual=false, loss=ε insensitive	waiting_time, waiting_lag1_time, cnt, Temp, att_Baron 1898, att_Joris en de Draak, att_De Vliegende Hollander, Weekend, att_Pagode, att_Polka Marina, att_Monsieur Cannibale, South	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5
2	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5	waiting_time, waiting_lag1_time, cnt, Temp_**diff**, att_Baron 1898, att_Joris en de Draak, att_De Vliegende Hollander, Weekend, att_Pagode, att_Polka Marina, att_Monsieur Cannibale, South	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5

Table 9: An overview of the input and output of the iterations done to select the best working feature set containing transformations and selecting the best working model for waiting time

Model that predicts crowdedness index.

The model that predicts the crowdedness index has only one continuous feature. This feature is transformed using a log transformation and a differencing transformation. This comes down to only three different combinations of continuous feature sets. The categorical features are added to each of these three combinations to create a complete feature set with a unique combination. All combinations are tested on a simple linear SVR model at the start. The outcome over the iterations is shown in Table 10. This process also stopped after two iterations because the input and output model were the same for the second iteration. Table 10 shows that the only continuous feature is not performing better when transformed. Therefore, the original non-transformed feature is used in combination with the binary features. The model that used these features with the lowest error metrics is a ridge regression model with an alpha of 1, a maximum number of iterations of 1000 and a tolerance set to 1e-5.

Iteration	Model for feature selection	Features selected (incl. DV)	Model found with grid search
1	Linear SVR: dual=false, loss=ε insensitive	crowd_index, crowd_lag1_index, square_Separate location 7, square_Separate location 3, square_Separate location 6, square_Python plein, square_Anton Pieckplein, square_Aquanuara plein, square_Max & Moritzplein, square_Sprookjesbos	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5
2	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5	crowd_index, crowd_lag1_index, square_Separate location 7, square_Separate location 3, square_Separate location 6, square_Python plein, square_Anton Pieckplein, square_Aquanuara plein, square_Max & Moritzplein, square_Sprookjesbos	Ridge regression: alpha=1, max. iterations=1000, tolerance=1e-5

Table 10: An overview of the input and output of the iterations done to select the best working feature set containing transformations and selecting the best working model for crowdedness index

5.3 Crowd management method evaluation

The methods as explained in Section 4.2.4 created the results that are displayed in this section.

Model that predicts waiting time.

The confidence interval of the mean difference for each crowd management method on the waiting time in minutes is shown in Figure 8 and Table 11. Both the physical signing methods show a positive difference, the push notification method shows no difference and the recommendation app method shows a negative difference.

The predictions used for the computation of the confidence intervals is calculated from models with the following error metric found in Table 12.

The effect sizes for each model can be found in Table 13. All effect sizes are very close to zero.

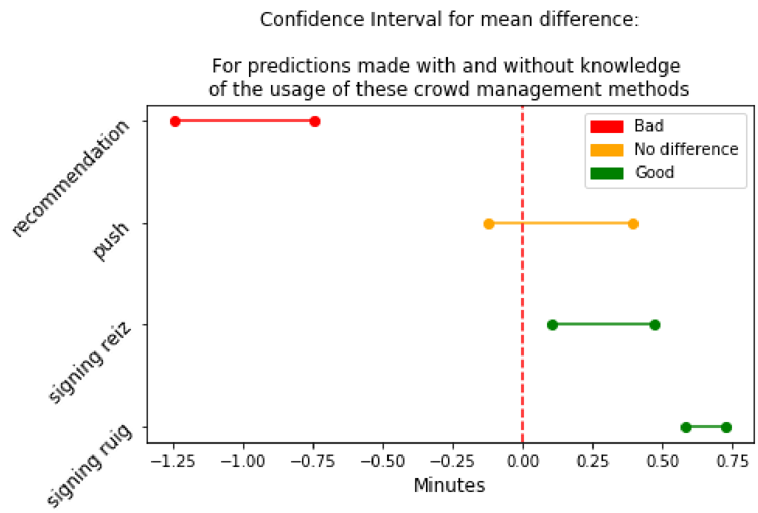


Figure 8: Confidence interval of the mean difference on the waiting time for each method in minutes. The exact numbers of the interval are given in Table 11.

Method	Lower bound	Upper bound
signing ruig	0.58	0.73
signing reiz	0.10	0.47
push notifications	-0.12	0.39
recommendation app	-1.24	-0.74

Table 11: Confidence interval of the mean difference on the waiting time for each method in minutes

Model trained without method for	RMSE
signing ruig	7.21
signing reiz	6.47
push notifications	7.21
recommendation app	6.2

Table 12: The error metric measured as RMSE for every model trained without a method, corresponding to each method for the waiting time models

Method	$\Delta \bar{R}^2$ value
signing ruig	-0.0039
signing reiz	-0.063
push notifications	-0.0116
recommendation app	0.0405

Table 13: The effect size of each method displayed in $\Delta \bar{R}^2$.

Model that predicts crowdedness index.

The mean difference confidence interval for each crowd management method on the crowdedness index is shown in Figure 9 and Table 14.

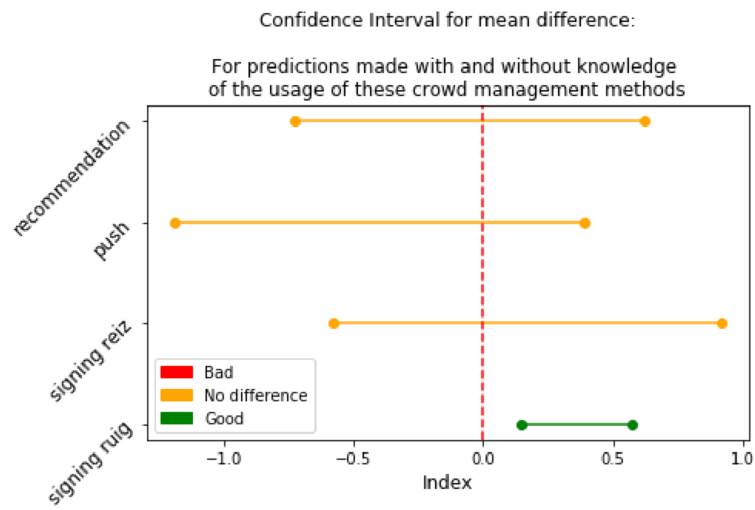


Figure 9: Confidence interval of the mean difference on the crowdedness index for each method in minutes. The exact numbers of the interval are given in Table 14.

Method	Lower bound	Upper bound
signing ruig	0.14	0.58
signing reiz	-0.58	0.91
push notifications	-1.19	0.39
recommendation app	-0.73	0.62

Table 14: Confidence interval of the mean difference on the crowdedness index for each method

The models that calculated the predictions used for the computation of the confidence intervals have an error metric, shown in Table 15, that will be taken into account for the discussion of these results.

Model trained without method for	RMSE
signing ruig	5.82
signing reiz	7.02
push notifications	5.82
recommendation app	6.54

Table 15: The error metric measured as RMSE for every model trained without a method, corresponding to each method for the crowdedness index models

The effect sizes for each model can be found in Table 16. All effect sizes are very close to zero.

Method	$\Delta\bar{R}^2$ value
signing ruig	-0.0133
signing reiz	-0.0239
push notifications	-0.0004
recommendation app	0.0006

Table 16: The effect size of each method displayed in $\Delta\bar{R}^2$.

6. Discussion

The goal of this thesis is to find the effect of three crowd management methods or combination of these methods on the waiting time of attractions and on the crowdedness in an amusement park. The findings of each subquestion and of the main questions will be discussed in Section 6.1 as well as the impact of this research in its current field. The limitations for this study and suggestions for any further research can be found in Section 6.2.

6.1 Findings

The findings of this study are divided among the four subquestions. After discussing these findings, they are all brought together to bring the answer of the research goals to the light. I will also be discussing what impact this research has in its field of study.

6.1.1 Features.

'What features can be used for the model and how important are they?' is the first subquestion. It is answered by computing a Spearman's correlation coefficient for the continuous features and a point-biserial correlation coefficient for the binary features. For the correlation between the features, a Spearman's correlation coefficient is computed for both the continuous and binary features.

For the model to predict the waiting times, the features 'waiting_lag1_time', 'visitor count', 'Temp' as continuous features and 'att_Baron 1898', 'South', 'att_Joris en de Draak', 'att_De Vliegende Hollander', 'Weekend', 'att_Pagode', 'att_Polka Marina', and 'att_Monsieur Cannibale' as binary features are the remaining features that can be used for this model. These features are in descending order of their correlation with the dependent variable, 'waiting_time', which can be interpreted as their importance for the model. The higher their correlation, the better the model probably predicts with these features in the future.

For the model to predict the crowdedness index, the continuous feature 'crowd_lag1_index' and binary features 'square_Separate location 7', 'square_Separate location 3', 'square_Separate location 6', 'square_Python plein', 'square_Anton Pieckplein', 'square_Aquanuara plein', 'square_Max & Moritzplein', and 'square_Sprookjesbos' are the remaining features to be used for this model. It was surprising that the features for visitor count and temperature were not highly correlated with the dependent variable, 'crowd_index'. In Abbaspourghomi (2020) it is explained that the crowdedness index is partially computed by knowing the waiting time of the attractions within that location. I was expecting these features to also be of some importance for the prediction of the crowdedness distribution index because these features were also important for the prediction of the waiting time. However, this was not the case and therefore the only continuous feature that remains is the 'crowd_lag1_index' feature.

6.1.2 Transforming data.

'Which set of transformed and non transformed features performs best in terms of error?' is the second subquestion. It is answered by transforming all continuous features and creating feature sets with all possible combinations of transformed and non-transformed features. The binary features are added to all these unique combinations to create a complete set of features. These sets are tested on a simple linear SVR model at first to find the best performing combination. The set of features that is outputted by this process is used to find the best performing model. That best performing model is than used to again find the best combination of features. This process went on until the input and output model were the same. This subsection solely discusses the sets of transformed features.

The model that predicts the waiting times outputted after two iterations a set of features. This set contained one transformed feature, the temperature feature that was transformed using differencing, meaning that the seasonal trend has been removed from this feature. It is interesting that the visitor count feature did not need to remove the seasonal trend to create a better performing model, because this feature is also heavily dependent on the seasons, just as the temperature feature.

The model that predicts the crowdedness index also outputted after two iterations a set of features that was further used in this research. This set did not change from the non-transformed set of features and therefore does not contain a feature that is transformed. Not even the dependent variable, 'crowd_index', needed a transformation to get a better performance.

6.1.3 Hyperparameters and model selection.

'What regression model with which parameters performs best in terms of error?' is the third subquestion. This question is answered by performing a grid search on several models and hyperparameter sets. The best model fit of each grid search is compared with each other based on RMSE. The model found with this grid search is used to again find the best set of features as explained in Section 6.1.2. This new set of features is then used to once more perform grid search and find the best model. This process went on until a model was found that was also found in the previous iteration. This subsection solely discusses the models selected in the process.

Both prediction models, for waiting time and for crowdedness index, found the same model that performed best in the grid search in the last iteration. This was a ridge regression model with alpha set to 1, a maximum iterations of 1000 and a tolerance of 1e-5. I was not surprised that for predicting the waiting time and the crowdedness index both found the same regression model that performed best in their case. I was expecting a similar model because both cases were very similar too. This also shows promising options to make a generalised process for amusement park the 'Efteling' to test their new methods on something other than waiting time or crowdedness distribution.

6.1.4 Effects of methods.

'What crowd management method or combination of methods has an effect based on confidence intervals, and what is the magnitude of this effect, based on explained variation?' is the fourth and last subquestion. It is answered by computing the mean difference confidence intervals of the dependent variable distribution when a method has been applied and the prediction distribution of the dependent variable of what that same period would have looked like if that method had not been applied and by computing the effect size. However, when interpreting these confidence intervals, the errors of the predictions should also be taken into account because this is the uncertainty of all the predictions made.

When evaluating the crowd management methods on the waiting time, using the confidence intervals, it can be seen that both placements of physical signing have a positive effect, meaning that with a 95% confidence it can be said that the waiting times are shorter if these methods are applied. The usage of the recommendation app shows a negative effect, meaning that with a 95% confidence it can be said that the waiting times are longer. Sending out the push notifications did not show any effect. However, the RMSE value of all the models to compute the prediction of what it should be like when a method was not used are high compared with the confidence interval values. These errors can be taken into account for the evaluation by including them in the confidence interval. By doing that, all confidence intervals would definitely include zero in this case. This would mean that, with the current predictions and corresponding error, none of the methods have an effect on the waiting time. The effect size also confirms this. Every method has an effect size close to zero, meaning there is no effect in these cases.

When evaluating the crowd management methods on the crowdedness index, only the placement of signing in 'Ruigrijk' had a positive effect. The other methods all showed no effect according to the confidence intervals. The models that computed the predictions for the crowdedness index also had a high RMSE value compared to the values of the confidence intervals. Again, including these error values into the intervals would result in all intervals including zero. This means that, with a 95% confidence, it can be said that the true value of the difference of these two distributions is between this interval, which could thus also be zero since the interval includes zero. And a difference of zero obviously means that there is no difference. The effect size for these models were almost zero, suggesting that there is also no effect in these cases.

The use of confidence intervals together with effect sizes did give a good representation of the relationships of the different distributions as was suggested by Nakagawa and Cuthill (2007). Because when only significance testing would have been done, it would not have become clear that the prediction model was not accurate enough. This is however made clear by using the confidence intervals to evaluate the crowd management methods.

6.1.5 Findings for main thesis goals.

The main goal of this thesis is to find the effect of three crowd management methods or combinations of these methods on the waiting time of attractions and on the crowd distribution in an amusement park. To get an answer for this, a model needed to be created to compare the effect of the methods with a baseline model (using no methods). To construct such a model, it is important to know which data to use as features, how to use this data combined with transformations, and which regression model and hyperparameters to choose. After finding the correct settings for all this, it can be determined if each method had an effect and what these effect sizes are.

The results of the confidence intervals with the error taken into account show that none of the methods is seen to be making a difference when using the models created in this research. Not for the waiting times of the attractions nor the crowd distribution in the park. The way of evaluating however seems to be a reliable method that could show very accurate results in the difference a method brings when looking at minutes of the waiting time or index for the crowd distribution. One thing this evaluation method needs, to create such accurate results, is a prediction model that also makes accurate predictions. If the prediction error is too high compared to the confidence intervals, nothing much can be said about the effect of a method. This is the case in the current situation, were the prediction error of all models is high compared to the confidence interval values.

Nevertheless, it could very well also be the case that these crowd management methods are just not effective enough. Which is, as Hummel and Maedche (2019) stated that the effectiveness of such a small environmental change, called a nudge, is related to the category and context of it. Which could mean that implementing such a method should not always be as effective as is thought. Relating to the non effectiveness of each method, the study of Zomer et al. (2015) proposed a crowd management system that takes both the characteristics of the crowd as well as of the urban mass event, in this case the amusement park, into account. The recommendation app evaluated in this research takes both these characteristics into account, however it was not proven that this made a difference. This study therefore does not support the research done by Zomer et al. (2015).

6.1.6 Impact on this field.

Although the main goal of this research is not fully completed due to the model error being to high, it still has an important role in the field of studying the managing of flow and crowds in amusement parks. This research is one of the few studies in this field that analysed the consequences of several methods in real life, and thus not in a simulation. Making inferences on analysis that has been done with real life data could give better insight on the effect of these methods in amusement parks. This research also shows an approach on how to evaluate crowd management methods (or other methods) when they have already been implemented in the past, which could give the option to do research in this field on data that has already been collected.

This research also impacted this field by studying how to measure the impact of the recommendation app on the behaviour and movements of visitors and therefore elaborating on the study done by Abbaspourghomi (2020).

6.2 Limitations and further research

Naturally, this study also brings limitations with it which could be taken into account when doing further research. Both the limitations and the further research will be discussed here.

The most important limitations of this study is the inaccuracy of the prediction model. Because the prediction model, that should predict what a period should have looked like when a method was not applied, is not accurate enough. The errors are too high, which means that the results of the prediction model could be very imprecise. An imprecise prediction that is used to compute the confidence intervals may follow in inaccurate results for the evaluation.

As mentioned in the introduction, on of the limitations of this research is that searching for the best set of transformed and non-transformed features and searching for the best performing model is done separately. The reason for this separation was to save computing time because this research had to be constructed within a restricted time frame. However, this way of searching for the best outcomes separately could result in a different outcome compared to searching for the best outcomes together. This has certainly the means to change the outcome of the whole research. When the set of features and the model would be different, the performance of the model could very well also be different, meaning that the error could perhaps be lower than is the case now. In further research it would be wise to take into consideration if this step should be done separately or together.

An additional limitation was that in the collected data for this research, the periods of execution for some of the crowd management methods overlapped with each other, making it more challenging to make correct inferences from the results. Future research could focus on executing just one method or executing the methods separate from each other.

Another limitation is that this research is very specific for one amusement park, the Efteling. This study could be improved by generalizing it more. The same analysis could for example be done in several different amusement parks to find if the results generalize to different amusement parks. E.g. Cheng et al. (2013) did research on data of several amusement parks by using massive agent-based simulations. This way, the research is not entirely dependent on the willingness of amusement parks to cooperate with applying crowd management methods.

The last obstacle was that this research is carried out during the COVID-19 pandemic in 2020. The measurements installed to keep this pandemic in check greatly influenced the data during that time. This is due to less people being allowed in the park and the attractions having less capacity which in turn influences the waiting time and crowdedness in general. All this makes it unsure if the results of this study will apply to other time periods, e.g. when there is no pandemic and restricted measurements. Therefore, it would be constructive to replicate this research for a time period that is not within this pandemic.

7. Conclusion

The goal of this study was to find the effects of three different crowd management methods on the waiting time of attractions and on the crowd distribution in an amusement park. This research attempted to find an answer to this goal by creating a prediction model and making predictions on a period where a crowd management method has been applied. The predictions that are made are to estimate what that period would have looked like if that method has not been applied at that time. These predictions are compared to the actual data using confidence intervals of the mean difference of these distributions. The process of creating a predictions model for this problem is also explored in this research. The confidence intervals, with the prediction error taken into account, did, in all cases, not show a difference. However, the predictions errors were high in comparison to the initial confidence intervals. To strengthen the results of the confidence intervals, an effect size is calculated per method. The effect sizes were all very close to zero, which would mean that none of the crowd management methods had an effect on the waiting time nor the crowd distribution in the amusement park. Eventually, the main research questions could not be fully answered since the model that computes the predictions was not performing optimally.

In conclusion, this study found that none of the crowd management methods showed an effect on the waiting time of the attractions nor the crowd distribution in amusement park the Efteling. The accuracy of these results could be improved by optimising the prediction model.

References

Abbaspourghomi, Abouzar. 2020. *A Personalized Recommendation System for Efteling using Crowdedness and Guest Behaviour*. Unpublished pdeng thesis.

Ahmadi, Reza H. 1997. Managing capacity and flow at theme parks. *Operations research*, 45(1):1–13.

Awad, Mariette and Rahul Khanna. 2015. Support vector regression. In *Efficient learning machines*. Springer, pages 67–80.

Beloiu, Iulian and Gergely Szekely. 2018. Theme park queuing systems: Guest satisfaction, a comparative study. Master's thesis, blekinge institute of technology, karlshamn, sweden.

Changyong, FENG, WANG Hongyue, LU Naiji, CHEN Tian, HE Hua, LU Ying, et al. 2014. Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, 26(2):105.

Charfaoui, Y. 2020. Hands-on with feature selection techniques: Filter methods. https://heartbeat.fritz.ai/hands-on-with-feature-selection-techniques-filter-methods-f248e0436ce5.

Cheng, Shih-Fen, Larry Lin, Jiali Du, Hoong Chuin Lau, and Pradeep Varakantham. 2013. An agent-based simulation approach to experience management in theme parks. In *2013 Winter Simulations Conference (WSC)*, pages 1527–1538, IEEE.

Davis, Mark M and Janelle Heineke. 1998. How disconfirmation, perception and actual waiting times impact customer satisfaction. *international Journal of Service industry Management*.

Drucker, Harris, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Fieller, Edgar C, Herman O Hartley, and Egon S Pearson. 1957. Tests for rank correlation coefficients. i. *Biometrika*, 44(3/4):470.

Fink, Ross and John Gillett. 2006. Queuing theory and the taguchi loss function: The cost of customer dissatisfaction in waiting lines. *International Journal*, 17.

Furnham, Adrian, Luke Treglown, and George Horne. 2020. The psychology of queuing. *Psychology*, 11(3):480–498.

Gardner, Martin J and Douglas G Altman. 1986. Confidence intervals rather than p values: estimation rather than hypothesis testing. *Br Med J (Clin Res Ed)*, 292(6522):746–750.

Harris, Charles R., K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Hummel, Dennis and Alexander Maedche. 2019. How effective is nudging? a quantitative review on the effect sizes and limits of empirical nudging studies. *Journal of Behavioral and Experimental Economics*, 80:47–58.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning*, volume 112. Springer.

JJ. 2016. Mae and rmse - which metric is better? https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d.

Kolli, Sindhu and Kamalakar Karlapalem. 2013. Mama: multi-agent management of crowds to avoid stampedes in long queues. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1203–1204.

Kornbrot, Diana. 2005. Point biserial correlation. *Encyclopedia of Statistics in Behavioral Science*.

Lang, K. 2020. Multiple linear regression. Tilburg University.

Leach, Lesley F and Robin K Henson. 2007. The use and impact of adjusted r2 effects in published regression research. *Multiple Linear Regression Viewpoints*, 33(1):1–11.

Lee, Kun Chang and Soonjae Kwon. 2008. Online shopping recommendation mechanism and its influence on consumer decisions and behaviors: A causal map approach. *Expert Systems with Applications*, 35(4):1567–1574.

Lin, Yiling, Magda Osman, and Richard Ashcroft. 2017. Nudge: concept, effectiveness, and ethics. *Basic and Applied Social Psychology*, 39(6):293–306.

Lu, Yina, Andrés Musalem, Marcelo Olivares, and Ariel Schilkrut. 2013. Measuring the effect of queues on customer purchases. *Management Science*, 59(8):1743–1763.

Martella, C, J Li, C Conrado, and A Vermeeren. 2017. On current crowd management practices and the need for increased situation awareness, prediction, and intervention. *Safety science*, 91:381–393.
Nakagawa, Shinichi and Innes C Cuthill. 2007. Effect size, confidence interval and statistical significance: a practical guide for biologists. *Biological reviews*, 82(4):591–605.
Ong, S. 2020. Rs: Spatiotemporal data analysis. Tilburg University.
Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
Pruyn, A Th H and Ale Smidts. 1993. Customers' evaluations of queues: Three exploratory studies. *ACR European Advances*.
Roberts, J Kyle and Robin K Henson. 2002. Correction for bias in estimating effect sizes. *Educational and Psychological Measurement*, 62(2):241–253.
van Rossum, G. 1995. Python tutorial, technical report cs-r9526, centrum voor wiskunde en informatica (cwi), amsterdam.".
Senecal, Sylvain and Jacques Nantel. 2004. The influence of online product recommendations on consumers' online choices. *Journal of retailing*, 80(2):159–169.
Singh, Harkiranpal. 2006. The importance of customer satisfaction in relation to customer loyalty and retention. *Academy of Marketing Science*, 60(193-225):46.
pandas development team, The. 2020. pandas-dev/pandas: Pandas.
Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
Yin, Ping and Xitao Fan. 2001. Estimating r 2 shrinkage in multiple regression: A comparison of different analytical methods. *The Journal of Experimental Education*, 69(2):203–224.
Yuan, Yuguo and Weimin Zheng. 2018. How to mitigate theme park crowding? a prospective coordination approach. *Mathematical Problems in Engineering*, 2018.
Zhang, Yingsha, Xiang Robert Li, and Qin Su. 2017. Does spatial layout matter to theme park tourism carrying capacity? *Tourism Management*, 61:82–95.
Zomer, Lara Britt, Winnie Daamen, Sebastiaan Meijer, and Serge Paul Hoogendoorn. 2015. Managing crowds: The possibilities and limitations of crowd information during urban mass events. In *Planning Support Systems and Smart Cities*. Springer, pages 77–97.

Appendices

A. Flowchart of this research

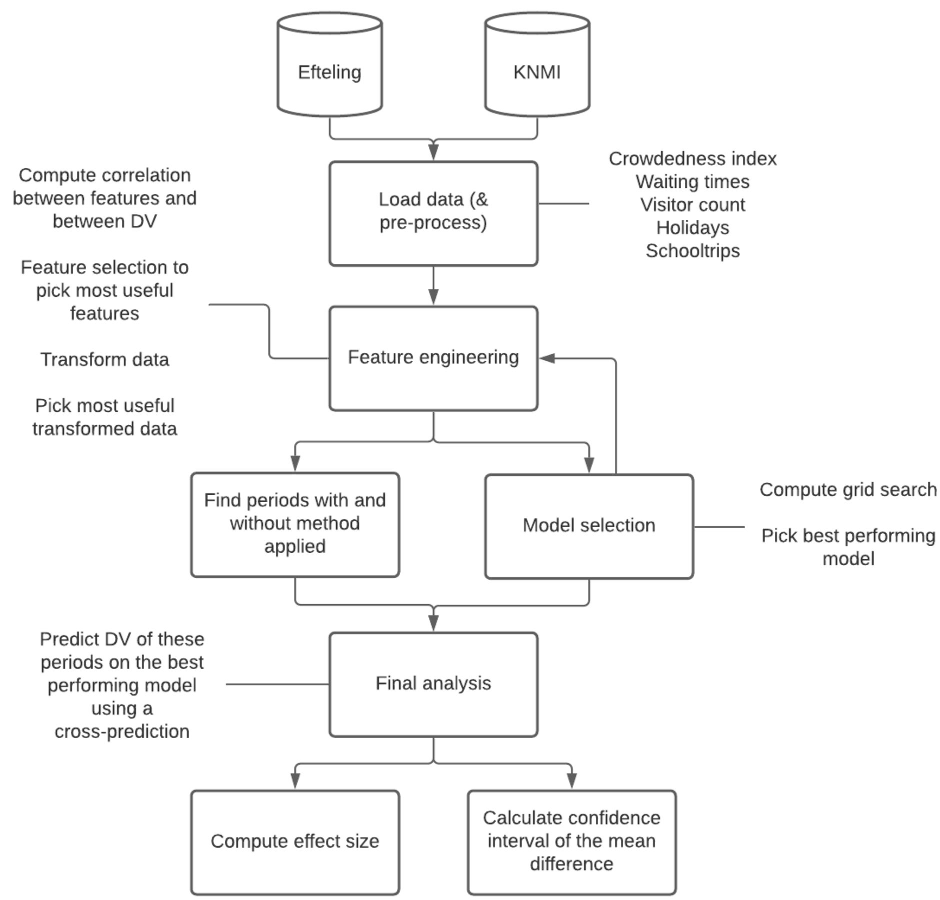
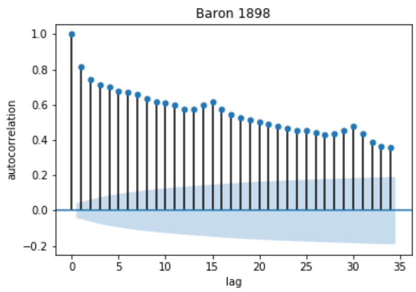
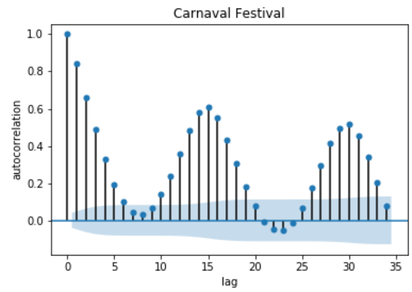


Figure 10: A flowchart of the processes for this research

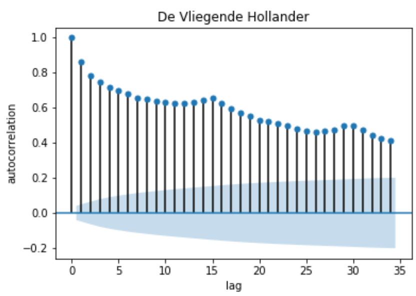
B. Autocorrelation plot per attraction



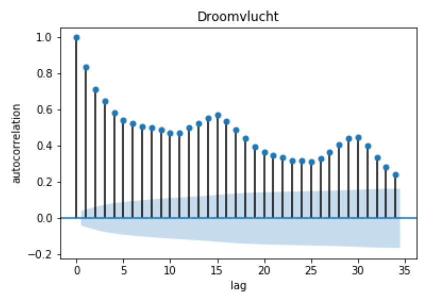
(a)



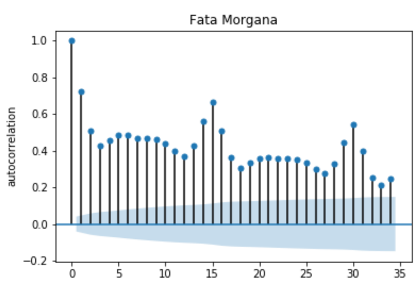
(b)



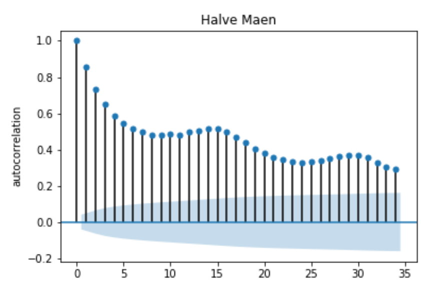
(c)



(d)

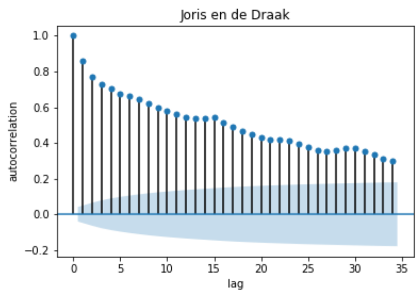


(e)

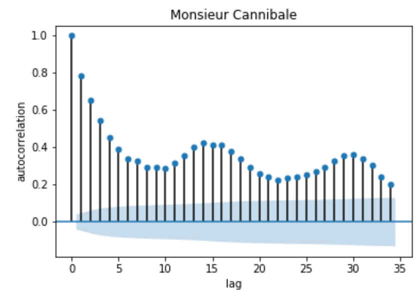


(f)

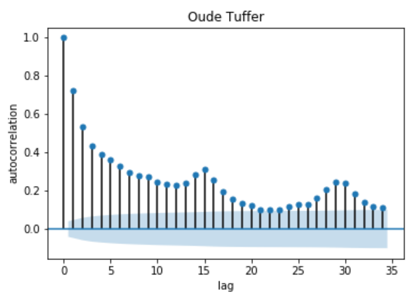
Figure 11: Autocorrelation plot per attraction



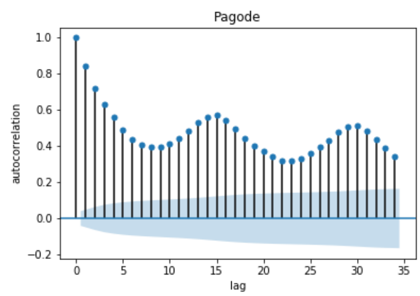
(g)



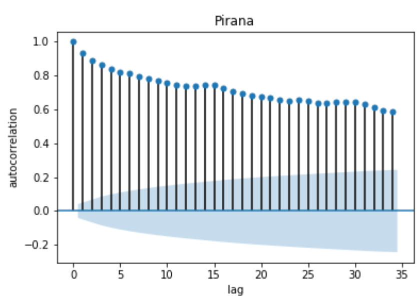
(h)



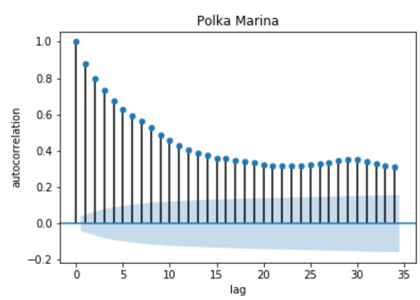
(i)



(j)

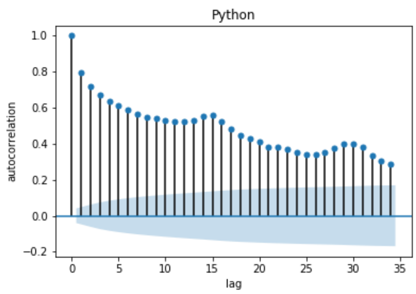


(k)

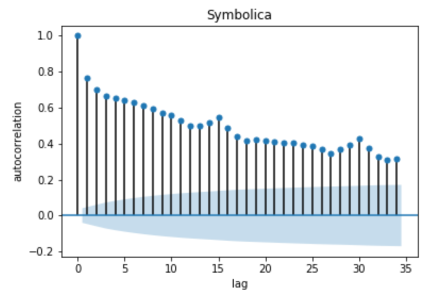


(l)

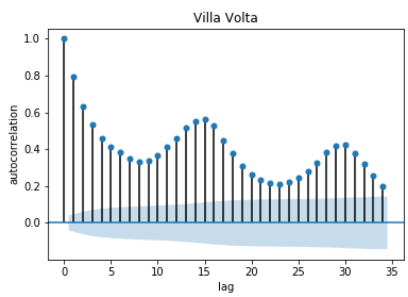
Figure 11: Autocorrelation plot per attraction (cont.)



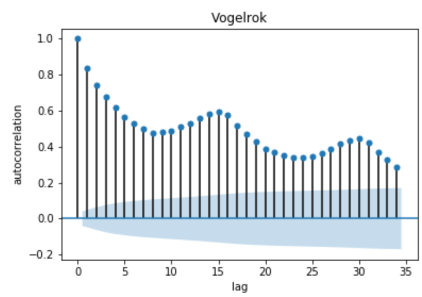
(m)



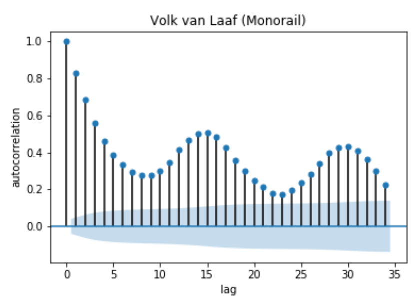
(n)



(o)



(p)



(q)

Figure 11: Autocorrelation plot per attraction (cont.)