ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Econometrics and Management Science

Master Thesis in Operations Research and Quantitative Logistics

Application of Machine Learning Techniques in Humanitarian Aid Forecasts

Student name
Hein
Tegelberg

Student ID number
385981

Supervisor
Albert
Wagelmans

Second assessor
Ilker
Birbil

Internship company
Argusi B.V.

Internship supervisor
Bas
Groothedde

January 19, 2021

Abstract

Humanitarian logistics, the practice of distributing aid goods to people in need, is one of the greatest contributors to costs in humanitarian operations. Even though its commercial counterpart has been continuously developed itself over the years, humanitarian logistics suffers from a "fire fighting" mindset in which data collection is seen as a luxury rather than a goal, with incomplete datasets and underdeveloped logistic methods as a result. In this paper we investigate several commonly used machine learning methods and explore their relevance to the process of creating a forecast model for the demand of aid goods as a result of an earthquake. Spectral clustering is used to reliably fill in missing data on countries in which earthquakes occur. As an alternative to traditional linear regression models, neural networks are introduced with which nonlinearity can be modelled. We introduce a variable-selection algorithm utilizing K-fold cross-validation for both neural networks and linear models, and a parameter-selection algorithm for neural networks specifically. Using several design choices, we create sixteen forecast models and evaluate their performance on new observations. We conclude that the machine learning methods outperform traditional methods in their respective applications. Additionally, we introduce a method of measuring the impact of forecast quality based on a real-life humanitarian operation. We use this method to compare the forecast of demand for a high-profile earthquake made by our best model to the official forecasts. Finally, we make suggestions for further research with which humanitarian logistics can learn from its commercial counterpart and become a thriving part of the future of humanitarian operations.

Keywords: *Humanitarian logistics, machine learning, neural network, demand forecast, K-fold cross-validation, impact measurement*

Contents

1 Introduction

Around the world, countless communities live at risk of being struck by circumstances that they themselves cannot cope with. When this occurs, mankind responds by helping those less fortunate, alleviating individual suffering and rebuilding damaged houses and infrastructure. The aid that is given to those in need is called humanitarian aid, generally provided by governments, NGO's, and private donors. It is of paramount importance that the required aid - be it food, medicine or shelter - is distributed to disaster victims in a timely manner, as delays result in prolonged suffering. However, the act of getting goods from a donor or depot to the affected area - also known as humanitarian logistics - is no trivial task. In fact, 80% of the costs associated with disaster relief are related to logistics (Van Wassenhove, 2006).

These relatively high costs are in part due to the effects of the high level of uncertainty associated with humanitarian work. Currently, a "fire fighting" mentality prevails, where the delivery of immediate aid is prioritized over planning and long-term strategy development. Even though this mentality is suitable for such urgent disaster situations, as a result, aid goods are hastily flown in from around the world with little regard to whether those goods actually need to be there at that moment. Not only does this lead to the use of quick but costly air traffic over more patient and cheaper ocean shipping, but also to an environment in which data collection is disregarded. Uncertainty in the volume of demand over time could be tackled by means of forecasting models, but limited data availability has rendered traditional forecasting methods to be less effective in humanitarian logistics than in commercial logistics.

Without the data that could be collected during a humanitarian response, effective applications of this data can not be developed. Without effective applications, there is no need to collect any data on humanitarian logistics. In this thesis we attempt to break this vicious circle in which humanitarian logistics is trapped. This research could initiate the use of modern forecast methods and higher quality data in the field of humanitarian logistics, bridging the gap between it and its commercial counterpart. Specifically, we will investigate in what way machine learning methods can be used to forecast demand for humanitarian aid in the wake of a sudden-onset natural disaster.

In our investigation, we apply machine learning in both our data collection and in the methodology used. Using those methods and data, we create forecast models for the demand for aid goods after an earthquake, comparing the impact of several machine learning techniques to their traditional counterparts. We select earthquakes specifically for their quantifiability, providing an exact date, epicentre, magnitude and depth. We collect data on recent earthquakes worldwide, high-accuracy population data that allows us to estimate the number of people living in the vicinity of a disaster, and data on several indicators on a national level that could be of significance to the required aid after a disaster. As these indicators are incomplete, we use spectral clustering to cluster countries together, filling in blanks in the dataset based on cluster averages. Based on the available data, we select the total number of affected people as the dependent variable in our forecasting models. In addition, we create a dataset using the natural logarithm of the number of affected people as the dependent variable in order to reduce the risk of overfitting to observations with a very large number of affected people.

In order to build forecasting models, we use linear regression for the traditional models. neural networks We use linear regression models and neural networks as the structures for our traditional and modern forecasting models, respectively. We create a variable-selection algorithm with which we select

the optimal variable set for each of the models. Neural networks also go through a parameter-selection algorithm to determine their hidden layer composition. Both of these algorithms are based on 5-fold cross-validation, and use either the Root Mean Squared Error (RMSE) or the Mean Absolute Error (MAE) in measuring performance. We also create models using a set of pre-selected variables rather than letting the variable-selection algorithm do its work.

With four different design choices - between linear regression model and neural network, between regular and logarithmic dataset, between RMSE and MAE, and between pre-selecting variables or not - we create 16 different models. We evaluate the performance of these models, with which we comment on the effectiveness of the machine learning techniques over traditional methods. In addition, we give direction to the development of forecast models in humanitarian logistics in the future. Finally, we put our best model to the test against official estimates of the demand for humanitarian aid after an earthquake, using an original method of measuring the impact of the quality of forecasts on costs of a humanitarian operation.

In our investigation, we find that the dataset filled by means of a spectral clustering of countries outperformed the dataset filled by means of a traditional regional clustering. We also find that models whose variables have been selected by means of our variable-selection algorithm outperform those created with pre-selected variables, showing the efficacy of the algorithm based on machine learning techniques. Most notably, we see that neural networks are more consistent and generally perform better than the linear models. These three findings, and other findings not discussed here, show that the application of machine learning techniques in the forecast of demand for humanitarian aid is very promising.

In addition, the best model we created is pitted against a perfect forecast and the official forecast of a high-profile earthquake in the past decade. Using our impact measurement method, we find that application of our forecast results in approximately 40 % higher costs than a perfect forecast. Considering the current vicious circle in humanitarian logistics and the problems that entails, this performance is quite good. When compared with the official forecast, our models incurs between 17 and 70 % lower, costs depending on the scenario used. Machine learning techniques do not only perform well when compared to theoretical traditional models, but also when compared to the models that were used in real life.

Having explored the ways in which machine learning techniques can be used to forecast demand in humanitarian logistics, we give directions for future research into this topic. We address both sides of the vicious circle in order that needs to be resolved before logistics can be a thriving part of humanitarian operations. We give ideas for the type of data that is required to further develop promising models. Especially time-series data on the demand after a disaster helps reduce costs by optimizing the transport of goods to an affected area. Furthermore, we give recommendations into the methods that can be expanded upon given the data that we collected. The machine learning techniques that we introduce is explored for their potential usefulness in humanitarian logistics, but not fully optimized. Optimizing other parameters than their hidden layers should further increase the performance of the neural networks. With these results and recommendations, this thesis should pave the way for the future of humanitarian logistics, in which machine learning and greater data proficiency bridge the gap between the humanitarian and the commercial world.

1.1 Problem Definition

Before we can start to perform our research, we would like to further elaborate on the question that this thesis answers. In this thesis we ask and answer the following main research question:

In what way can machine learning techniques be used to forecast demand for humanitarian aid in the wake of a sudden-onset natural disaster?

In this main question, we touch upon several domains. To further elaborate on these domains, as well as give some structure to this paper, we define several subquestions as follows:

1. **In what way can humanitarian logistics benefit from machine learning in demand forecasting?**
   (a) What is humanitarian logistics?
   (b) How is machine learning applied to demand forecasting?
   (c) When are either demand forecasting or machine learning employed in humanitarian logistics?
2. **How can we use machine learning to forecast demand for humanitarian aid?**
   (a) What relevant data and predictors are available?
   (b) What machine learning techniques can we apply?
   (c) How can we develop a forecasting model?
3. **How well can we apply forecasting models to real-life humanitarian situations?**
   (a) How do our models perform on our data?
   (b) How can forecasts help solve real-life problems?
   (c) What is the impact of our forecast models on real-life scenarios?

Subquestion 1 builds the foundation for the rest of this thesis, summarizing what has been researched in the past and how the fields of machine learning in demand forecasting and humanitarian logistics can be brought together. This subquestion is answered in Section 1.2, where relevant recent literature from the fields of humanitarian logistics and machine learning in demand forecasting is discussed. In subquestion 1a we look at what the field of humanitarian logistics entails and define the scope of this field. For subquestion 1b we investigate the application of machine learning in demand forecasting in commercial logistics. In subquestion 1c we list to what extent these practices have already been incorporated separately into the field of humanitarian logistics.

In subquestion 2, we investigate what kind of forecasting model we can create and how we can develop it. We answer this question in Sections 2 and 3, in which we discuss the data and methodology used in this thesis. The choice for the kind of forecasting model largely depends on the available data and predictors, which we discuss in subquestion 2a and combine into a dataset. This is not a trivial question as there are many elements in humanitarian aid that are subject to uncertainty and could therefore use forecasting models. In subquestion 2b we introduce the machine learning techniques are compatible with the created dataset. As there are many choices in modelling, we list some of them in subquestion 2c. With our data, machine learning techniques and modelling choices we create a set of forecasting models.

Finally, this thesis will discuss the results of this research and its relevance in further work both academical and practical. We do this by means of subquestion 3, which is discussed in Section 4. We begin by validating our models for observations that had not been used in the creation of the models.

Subquestion 3a corresponds to this validation, for which we comment on the accuracy of the created models. From these results, we select the model that is most likely to reliably forecast the demand for unknown observations as our final model. To evaluate the impact of forecasts on logistic problems, we develop an environment that simulates a simplified humanitarian operation in subquestion 3b. Using this environment, we measure the impact of the forecasts made by our best model in subquestion 3c. We analyse the expected benefit of using the forecast model in a real-life scenario. To showcase this, we compare historical results of the decisions made during the event itself with the potential results of the decisions that would be made using the new information provided by the forecast model.

Together, these subquestions form the foundation of the answer to the main research question. Through these questions we create an image of the current state of affairs in the world of humanitarian logistics, we take stock of available data and machine learning techniques in order to make a forecast model, and investigate to what extent our model could influence the response to real-life disaster situations.

1.2 Literature Review

There are two distinct academic fields upon which we expand this thesis. The rest of the research will be based on relevant literature found in these fields. These are the fields of humanitarian logistics and of demand forecast using machine learning. In this thesis, we will be combining these two fields. Before we can do that, however, it is important to learn how these fields have been (jointly) researched in the past. To this end, we will investigate the following topics; humanitarian logistics in general, the application of machine learning in humanitarian logistics, demand forecast in the field of humanitarian logistics, and demand forecast using machine learning techniques in other fields.

1.2.1 Humanitarian Logistics

In humanitarian logistics, one of the leading papers is Van Wassenhove (2006). This paper gives a good recapitulation of the state of affairs in the world of humanitarian logistics and discusses where its challenges and opportunities lie. It identifies several types of humanitarian crises. First of all, a distinction is made between man-made and natural disasters. Secondly, there is a difference between slow-onset and sudden-onset disasters. Man-made disasters often flow from political situations or are related to failure of man-made structures. Natural disasters are related to everything outside of man's influence, such as tectonic movements or meteorological circumstances such as drought and rain. Then there are disasters that occur suddenly such as landslides or chemical plant failures, and longer-expected disasters such as famine or a refugee crisis. These disaster types require different approaches with regards to prevention and response. For this research we chose the segment of natural sudden-onset disasters, as the impact of proper data analysis in this field is large whilst also having clear predictors such that forecasting is possible.

Van Wassenhove (2006) also emphasizes the fact that humanitarian organisations are under pressure from donors to prove that their donations are effectively reaching those in need. This accountability has increased the importance of transparency in costs, bringing the large impact of logistics in the humanitarian sector to light. A slow rise in the attention given to logistics in the sector had been developing, but the paper states that, at the time, humanitarian logistics lagged about 15 years behind their commercial

counterparts. This is due to a vicious circle in the humanitarian world where a lack of understanding of the importance and workings of logistics causes logistics to be disregarded in planning, which in turn causes a "fire fighting" mentality where the quick delivery of aid has priority over long-term planning and reflection on past operations. This causes logistics to fail to learn from the past, causing logistics to be once again disregarded when reviewing an operation, ultimately furthering the lack of understanding of logistics.

A recent overview of the research that has been performed since is given in Behl and Dutta (2019). This paper summarizes what developments have been made in different fields of humanitarian logistics in order to identify where further research would be necessary. Here, many usages of big data in the field of humanitarian logistics are discussed, such as the application of machine learning to use aerial data for disaster response in Ofli et al. (2016). In this research, human input would train a machine learning algorithm to recognise objects of interest such as collapsed buildings or damaged infrastructure. Other uses of big data and machine learning include analysis of publicly available social media messages to evaluate the scale of natural disasters. Despite the research already done, there is still a call for further research in predictive analysis for post-disaster relief operations.

Yu, Yang, and Yun Li (2018) presents a review on the application of big data in natural disaster management. In all stages of disaster management - mitigation, preparedness, response and recovery - big data has seen a rise in applicability. Below we will discuss the application of big data and machine learning in these four phases as described in Yu et al (2018).

In the mitigation phase, adequate use of big data has had a positive impact on the reliability of predictions of the occurrence and magnitude of meteorological events such as typhoons, storms and weather in general. An example of this is Zhang and Weng (2015) where a system based on radar observations managed to predict hurricane track, intensity and other characteristics up to five days before landfall.

In preparedness, the phase just before disaster strikes, previously untapped sources of data are employed. Satellite data and publicly available social media activity are used to follow the course of a disaster, and reliably give early warnings to those in the path of the disaster. For instance, social media coverage of a flooding in a village further upriver combined with supporting satellite data might warn a city downriver just in time to mobilise an evacuation or begin deploying sandbags earlier than they normally could.

In the phase this research would like to focus on, the response phase, the focus has thus far been on damage assessment. This is of great importance as search and rescue missions should be able to get to areas that are likely to have trapped people as quickly as possible. These damage assessments are based on visual data collected by unmanned aerial vehicles such as drones, from which worrisome features such as rubble piles are identified by human eyes. In addition, location data provided by smartphones and other devices were used to identify possible trapped persons. However, in this phase we do not yet see the use of modern problem solving techniques such as the application of machine learning to more efficiently recognise landmarks associated with collapsed buildings. Also, there has not yet been any application of big data concerned with forecast of aid demand. This is consistent with the observations regarding the vicious circle in Van Wassenhove (2006).

The recovery phase, which takes place after the resolution of the disaster, is the phase where big data

are least utilised. In this phase, the urgency of the situation has diminished significantly and normal life is slowly returning for the victims of the disaster. Because of this, analyses related to the disaster are not as needed as before. The only reported occurrence of big data analysis is in the observation of aerial and satellite data with regards to the reconstruction of a region in the wake of a disaster.

Demand forecasting in humanitarian logistics has historically been a difficult topic. Many in the field of humanitarian logistics claim that unpredictable demand is an insurmountable obstacle in disaster relief supply chains. Admittedly, predicting when and where disasters will occur far in advance is nigh impossible. In response, Jahre et al. (2011) states that demand is only unpredictable because there is not enough data collected, while there is not enough data collection because its effectiveness is disputed. This is in accordance with the statements in Van Wassenhove (2006). Jahre et al. (2011) then presents a method in which demand can be reasonably forecast based on empirical data. For this, an estimation was made based on disaster type and region by aggregating demand for those types and regions each year and averaging those based on prevalence of the specific disaster type. These estimations only concern the yearly volume of aid required which is certainly useful for humanitarian organizations. Even though its findings are not directly applicable to individual disasters, we can use them in order to convert the number of affected people by a disaster to the volume of aid that is required in response.

Humanitarian aid also concerns slow-onset disasters, as many NGO's operate to combat poverty and hunger in developing countries. In Laan et al. (2016), the demand forecast for goods in long-term aid projects of Médecins Sans Frontières is analysed. According to this paper, consumption forecast is biased to be higher than reality. This is partly due to incentives for medical personnel, as a high consumption forecast will cause goods with high criticality and demand variability to be delivered more urgently. This can be the cause of bullwhip effects further up the supply chain, leading to unnecessarily high costs.

Laan et al. (2016) also identified several situational factors that are of influence on the consumption of aid goods that might also be relevant for our own research. These relate firstly to population setting, whether a project is set in a rural or urban region, or a combination of both. Secondly, the socio-economic context is of importance, whether a region is (in)stable or (has been) in state of conflict. Lastly, the population type is of importance, which indicates whether a native population or refugee population is targeted by the project.

In conclusion, much of the research in the field of humanitarian logistics is built on the findings of Van Wassenhove (2006). There, an issue is raised regarding data proficiency in the humanitarian world. In addition, some key definitions are made regarding disaster types and phases in disaster response. We find that big data and machine learning were becoming more popular amongst researchers in this field since 2006, with several promising developments being made recently. However, in the response phase there was still opportunity with regards to demand forecast, a research gap in which we position this paper. In demand forecast in the humanitarian world, much work has been done to challenge the belief that forecasting is impossible. Recent research has tried to reliably forecast demand, but has until now not succeeded due to numerous reasons. The application of machine learning to forecast demand in humanitarian logistics, however, has not yet been reported on.

1.2.2 Machine Learning in Demand Forecasting

Here we will discuss recent literature on the application of machine learning in demand forecasting. Some of the techniques that we discuss here will be elaborated on in Section 3 as those will be applied in this research. Other techniques mentioned below will not be part of this research, but help us in gaining insight in previous research.

In supply chains, the underlying demand structure of other participants in the chain is often unknown. Therefore, forecasts historically respond to actual demand rather than expected demand. This leads to the so-called bullwhip effect, causing large forecast errors further up the supply chain. Carbonneau, Laframboise, and Vahidov (2008) uses machine learning techniques in order to streamline supply chains, especially at the upstream end of the chain. Their research employs both machine learning techniques as well as traditional forecasting methods. Neural Networks (NN), Recurrent Neural Networks (RNN) and Support Vector Machines (SVM) are introduced as machine learning techniques. These techniques allow for non-linearity, whereas traditional forecast methods only take linear models into account. For the NN, a three-layer (one hidden layer) feed-forward back-propagation is used. RNN used back-propagation through time, which is especially useful when time series data is concerned, in addition to the same three layers as NN. The methods were tested on both simulated and real data. RNN and SVM were both good performers, whereas NN suffered from overfitting. Ultimately, the machine learning techniques performed better than the best traditional technique.

Yildiz, Bilbao, and Sproul (2017) applies machine learning techniques in the field of electricity load forecasting, where energy consumption of buildings is forecast based on regression models. This paper models the hourly demand one day ahead using NN, SVM and Nonlinear Autoregressive Networks with Exogenous inputs (NARX). This research also includes simulated data next to historical data for forecasting energy consumption of buildings that did not have historical data. This was possible due to previous simulation methods having been developed in the field. The research concluded that Multivariate Regression Models (MLR) were useful in the specific application because of better user engagement, clearly showing which parameters had what influence. It should be noted that this conclusion was made for an audience that was unfamiliar with machine learning models. With regards to forecast accuracy, however, NARX outperformed all other forecast methods because it incorporated some of the unique features of the field.

Another paper in the field, Shi, Xu, and R. Li (2017), uses a pooling-based deep recurrent neural network where groups of consumers' energy consumption was pooled to overcome overfitting issues. This outperformed both traditional and machine learning forecast methods with regards to forecast error. This application differs quite from humanitarian logistics, however, where the pooling of disasters seems impossible due to different external variables.

Debnath and Mourshed (2018) analyses the developments in energy planning models over the period 1985-2018. Fifty different forecasting methods are identified, of which the most widely used machine learning methods were NN and SVM. In traditional forecast methods, Autoregressive Integrated Moving Average Models (ARIMA), Linear Regression Models (LR) and Autoregressive Moving Average Models (ARMA) were most widely used. This paper concluded that machine learning methods were more

accurate in their forecasts than their traditional counterparts.

Yanting Li, Su, and Shu (2014) introduces a method not yet mentioned before in the field of electric load forecasting which combines traditional time series models with unique properties of the field. In a so-called ARMAX model, the traditional ARIMA model is extended with the use of exogenous factors. The paper presented a one-day ahead forecast, for which the ARMAX performed better than NN methods previously used in the field of electric load forecasting.

Seeing that machine learning has often been applied in demand forecasting in other fields, we expect to do so too when forecasting demand in humanitarian logistics. From the literature we see machine learning techniques such as (recurrent) neural networks often outperform their traditional counterparts. There is, however, no absolute best method for forecasting demand as performances depend on data structure and other demand characteristics. We elaborate on the selected methods and other methodology employed in this thesis in Section 3. The methods that we select for forecasting demand depend on the available demand data, as certain machine learning techniques only work with certain data structures. The available data is discussed in the next section.

2 Data

Accuracy is of great importance for any kind of research. This is especially the case in data science, where overlooked inaccuracies can lead to vastly different conclusions. In addition, data should be easily understandable for those who might replicate or further elaborate on this work. It is therefore important in this thesis that our data is both accurate and easily understandable for people from different academic backgrounds. In this section we explain why we choose our specific data and from which sources these datasets come, in addition to how we build up final the dataset. We discuss three data types; disasters in Section 2.1, population in Section 2.2 and local indicators in Section 2.3. Finally, the dependent variable we want to forecast is discussed and analysed in Section 2.4.

2.1 Disaster Data

As mentioned above, our data needs to be accurate and consistent. When considering different types of natural disasters, characteristics are usually quite different. A flood can cover a certain area, and can then also remain there for a certain time. A typhoon can reach a certain maximal wind speed, and can travel a certain distance over a certain time over land. As a result, properly quantifying these disasters is very difficult. We have therefore chosen to only make use of data regarding earthquakes, as they have an accurate magnitude on the scale of Richter and have exact recordings of where and when an earthquake occurs. We feel that this choice makes the research achievable without compromising the quality of the answers to the question this thesis poses, which is how machine learning can be applied in the context of demand forecasting in humanitarian logistics.

We found a reliable source for significant earthquakes, provided by NCEI/WDS Global Significant Earthquake Database (SED). Here, significant earthquakes are defined as earthquakes with at least moderate damage of 1 million U.S. Dollars (USD) or more, 10 or more deaths, a magnitude of at least 7.5, a Modified Mercalli Intensity of at least X, or with a tsunami associated with it. An earthquake with Modified Mercalli Index of X is described as extreme by Wood and Neumann (1931), with "some well-built wooden structures being destroyed", amongst other descriptions. From this dataset, we select only earthquakes that occurred without secondary effects such as tsunamis or volcanic activity as we expect these secondary effects cannot be quantified in this thesis. In addition, we only select earthquakes from years for which regional data was available, which led to the choice of them happening between 2000 and 2019. Finally, a few data points that did not have complete information on magnitude or focal depth are removed from the dataset, leaving us with 930 observations. Each observation consists of an ID, coordinates, magnitude and focal depth of the earthquake, number of dead/injured/missing people, and the number of damaged or destroyed buildings as a result of the earthquake. Data on the number of dead, injured and missing people, as well as data on the number of damaged and destroyed buildings are not always fully present. As these variables are the only candidates for dependent variables, we devise a method to generate the dependent variable in Section 2.4.

Having decided on which disasters to be researched, we also require further data to connect event information on date and location to indicators on population, environment and healthcare.

2.2 Population Data

In addition to data on earthquakes, we require information on population levels near the epicentre at the time of the earthquake. To this end, we would like to determine the population within 25, 100 and 250 kilometres of the given coordinates. An estimation per grid of the world was found at NASA's Socioeconomic Data and Applications Center, with estimations for the years 2000, 2005, 2010, 2015 and 2020. This dataset goes by the name of Gridded Population of the World (GPW).

For precision, we choose a grid of 2.5 minutes, equal to 1/24 of a degree. This size corresponds to approximately 5km at the equator, meaning that the grid data gives the estimated population for each box of 25 km^2 at the equator with a diminishing size as you get nearer to the poles. We choose this resolution for its computational tractability whilst maintaining precision. We develop a script that translates coordinates into an ellipsis in degrees around the coordinates that corresponds to a circle in kilometres around a point on Earth. Using this script, we are able to estimate the population within 25, 100 and 250 kilometres of the epicenter at the time of the earthquake from the GPW.

2.3 Local Indicator Data

As we would expect that certain variables related to development of a country are of influence on the scale of the effects of an earthquake, we would like to include local indicators in our model. Here we have to make a decision between geographical precision and completeness of the dataset. As most data are only collected on a national level rather than on a regional level, in addition to the fact that the SED does not specify regions of countries, we decide to search for national data only. For this, we make use of the World Development Indicators (WDI) database of The World Bank, which collects and stores data on a large number of development-related variables.

As the WDI consists of 1431 indicators, we make a selection of the variables we want to include. We include these for the years between 2000 and 2019, as this corresponds to the years of the earthquakes in the SED. From this list of indicators, we choose 50 based on their completeness and their likely relevance to aid goods consumption, drawing inspiration from Laan et al. (2016). The complete list of selected indicators is found in Appendix A. To deal with incomplete values in the WDI, we devise a procedure.

Firstly, for all missing values where there is a value present at another moment of time, a logarithmic prediction model is fit. This is done by means of a linear prediction model using Ordinary Least Squares (OLS) on the logarithmic values of the values in the dataset and the corresponding years. The exponential values of the predicted values are inserted in the missing places of the dataset and truncated where necessary.

Secondly, values that are missing completely have to be filled in. For this, a clustering of countries needs to be made such that each cluster has at least one value for each indicator. This way, countries without values for an indicator can copy the average of their cluster without fail. Several countries without enough data are omitted from this clustering, as they will make it harder to fulfill our requirements to fill all missing data. None of the omitted countries have any earthquakes associated with them in SED. The results of our clustering, as well as the performance of different clustering methods can be found in Section 4.2.

Finally, the average values of the relevant cluster for that year are filled in where data was missing.

2.4 Dependent Variable

With the three types of data present, we combine all information into one dataset. This dataset contains 930 observations, each with data on the earthquake itself, the number of people living in the vicinity of the earthquake at the time, and data concerning the country in which the earthquake occurred. However, we have yet to decide on how we want to forecast the demand for humanitarian aid. Given the available data, it is impossible to reliably generate time-series data on the demand. We can, however, generate data on the total demand for humanitarian aid after an earthquake. For this, we make use of Jahre et al. (2011), where it is concluded that demand for humanitarian aid goods is linear with the number of affected people. In fact, it is stated that in case of an earthquake, on average each person affected requires a total of 1017kg of aid goods. 60% of those goods are required in the first three months, 30% in the second three months, and the last 10% is required in the period between 7 and 12 months after the disaster. This information will prove useful when applying the results of our model to a real-life situation in Section 4.1.

We therefore decide that our model will forecast the number of people affected by a disaster, even though not all observations had that information readily available. As a result, we have to estimate the number of affected people by means of the number of buildings that were damaged, if direct information on the number of affected people is absent. Using historical data on household size, we can estimate the number of people that were made homeless and were therefore affected based on the number of damaged buildings. We combine household size data from the United Nations Department of Economic and Social Affairs and from the Global Data Lab in order to estimate the household size for each country for each year.

From these two distinct methods of estimating the number of affected people, we choose the highest number of affected people. We do this due to the observation that disasters often only recorded one of the two statistics which excludes the possibility of a minimum or a linear combination of the two. However, we notice that sometimes the estimated number of affected people was far larger than the estimated number of people within a 250km radius, even for earthquakes with low magnitudes. Due to the unreliability of these population statistics, final models that are created using the full dataset perform poorly. Therefore, we decide to remove all observations with erratic estimations of the number of affected people from our dataset. After this decision, we are left with 656 observations, each with reliable data on the disaster, on the nearby population, on the situation in the country and on the number of people affected.

From the total dataset of 656 observations we form a training set and a testing set of 525 and 131 observations respectively. In short, the training set is used to create forecast models and the testing set is used to evaluate the performance of those models. These sets are formed chronologically, where the earthquakes in the training set occurred before those in the testing set, as is realistic when performing forecasts. This division, however, causes the variance of the dependent variable in the training set to be more than 30 times the variance of the dependent variable in the testing set. This poses no problem in itself, but could have implications for the performance of our forecasts.

We would like to take a better look at the structure of our dataset, in order to preemptively identify issues that might else come forward in modelling. The distribution of the values of the dependent variable is presented in Figure 1. We sort the observations based on the value of the dependent variable,

and each bar in the chart shows that value. In Figure 1, we see that the distribution of these variables is distorted by a few of the observations with the largest dependent variable value. This could severely skew our modelling, causing premature termination of algorithms or overfitting to the largest observations. Therefore, we would like to not only use the regular dependent variable in the development of forecasting models, but also something less vulnerable to overfitting. We find this in the natural logarithm of the regular dependent variable, the distribution of which is presented in Figure 2. We see a much more even distribution once the natural logarithm has been taken, which should iron out some of the issues that a more skewed distribution causes. There is also a cost associated with the natural logarithm as information on the actual scale of disasters is unknown to models that use it. We will be using both datasets - one with regular dependent variable and one with the logarithm of the regular dependent variable - to test, train and tune our models.

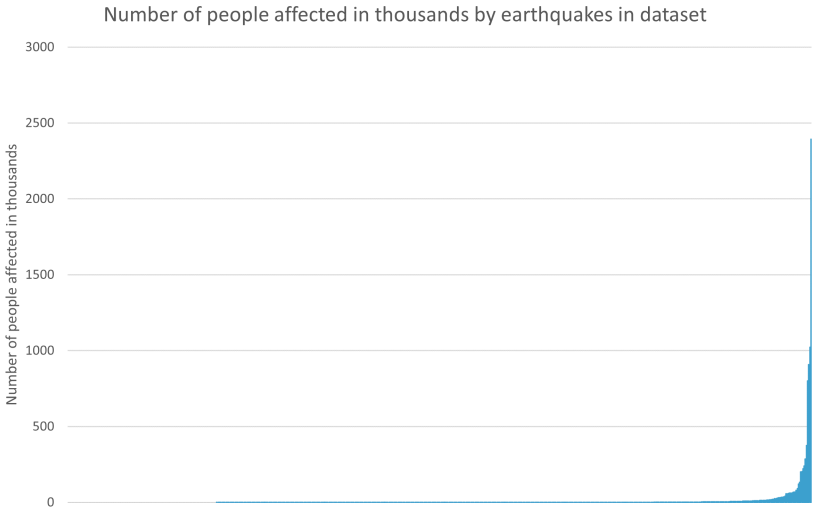

Figure 1: Distribution of regular dependent variable

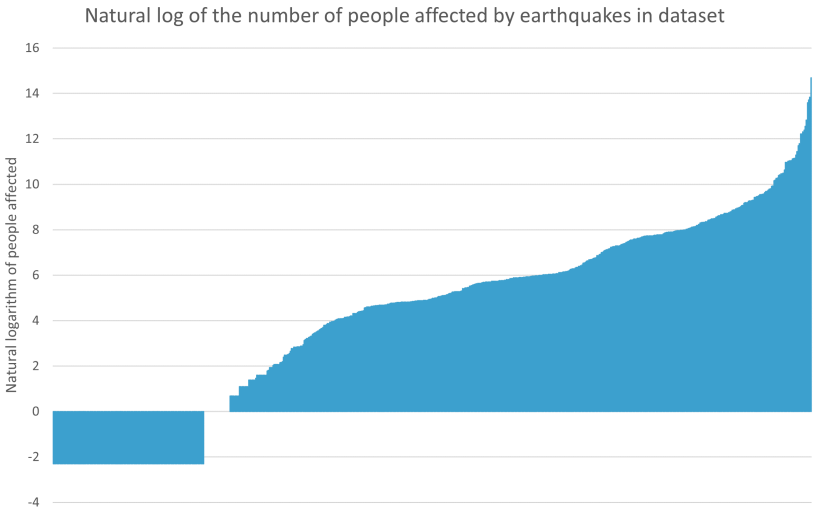

Figure 2: Distribution of the natural logarithm of the regular dependent variable

In all types of data, save for seismic data concerning earthquakes, we have been required to make assumptions, generalizations and estimations. By increasing the accuracy of our data through better recording or availability, we are able to improve our models. On the one hand, having more complete observations gives models more information to learn from once we no longer have to exclude certain observations for being questionable. On the other hand, having values of indicators and dependent variables that are closer to their real-life values lets our models capture relations that are present in reality, rather than in our dataset. Having more data being collected in general, such as exact data on the daily demand of aid goods after a disaster, will also greatly expand the scope of research that can be performed in the field of humanitarian logistics. For now, however, we will use the number of affected people as the dependent variable with which we develop our models.

3 Methodology

In order to build a working model, we delve into some of the methodology that is commonly used in machine learning and how we apply these methods in creating our model. Before we introduce these methods, we should explain an important characteristic of datasets as well as give some background to how forecasting models work.

In machine learning, both supervised and unsupervised learning methods are employed based on the type of data that is subject to research. In supervised learning, a data point consists of a number of explanatory variables which we will call X and a dependent variable called y. Ideally, these X have some significance in explaining the value of y. Using the supervised data points, the influence of these X on the value of y is assessed, and stored in a parameter vector β. Then, when a new data point i is introduced for which we want to estimate y_i, we use the values of X_i for the new data point and the learned parameters in β to come up with a estimated value $\hat{y}_i$. Because we can supervise the actual value of y_i, we can derive the error of our estimate using the values of y_i and $\hat{y}_i$. An example of this is linear regression, a well-known method that we will employ further on in this thesis.

In unsupervised learning, we only know the values for the variables X for each of the data points. Here, the goal is to find patterns or groups in datasets. For instance, a set of data points can be subdivided into clusters based on their values of X. Instead of building a model based on minimizing of the error between estimated and actual values of the independent variable, here we minimize the distance between the data points within each cluster.

In this section, we will be addressing different methods in the order that they will be applied for our computational experiments described in Section 4. First, we describe how a forecast can be applied to real life in Section 3.1. Secondly, we describe the spectral clustering algorithm used to fill in missing values in our data in Section 3.2. In Section 3.3, the concept and workings of neural networks are explained. Finally, we need to select variables to include in our models and the value of parameters as described in Section 3.4.

3.1 Impact Measurement

In order to measure the impact of a forecast model, we first need to know in what way such a forecast is generally employed. As our thesis concerns the forecast of demand for humanitarian aid goods, we will focus on the impact with regards to the logistics behind the delivery of aid goods. To show this impact, we introduce a simplified version of the mechanisms at work in a real-life humanitarian operation. We assume that all goods are present at the supply depot from the start, that there is only a single commodity to be demanded and shipped, that transportation methods have no initial costs, availability or capacity limit, and that surplus goods cannot be resold.

Using this simplified version, we can calculate the impact of the quality of a forecast on the costs associated with transporting aid goods to their destination. In this problem, there is uncertain demand without prior knowledge on its distribution, making an optimization impossible. By means of a forecast for the demand in each period, however, the problem becomes easily solvable. The application we create here is the result of a humanitarian organization solving the problem of selecting shipping methods when

faced with uncertain demand from a single supply depot. Mathematically, the problem becomes:

$$\min \sum_{t=1}^{T} \big(c_a(x_{at} + f \cdot y_t) + c_o x_{ot}\big) + b\big(x_{at} + x_{ot} + y_t\big), \tag{1}$$

subject to:

$$x_{at} + x_{ot} + y_t \geq d_t, \qquad t = 1, \ \dots \ , T, \ d_t \in D, \tag{2}$$

$$x_{at} \in \mathbb{R}^+, \qquad t = \lambda_a, \ \dots \ , T, \tag{3}$$

$$x_{ot} \in \mathbb{R}^+, \qquad t = \lambda_o, \ \dots \ , T, \tag{4}$$

$$y_t \in \mathbb{R}^+, \qquad t = 1, \ \dots \ , T. \tag{5}$$

We will go through this optimization problem globally. The goal is to minimize the cost function given in Equation 1, whilst satisfying the uncertain demand for aid goods d_t in each period $t = 1, \ \dots \ , T$ in Equation 2. These periods are counted after a disaster occurs, such that period 0 represents the moment of the disaster.

We have three decision variables; x_{at} represents the volume of goods arriving at moment t by air, x_{ot} represents the volume of goods arriving at moment t by ocean, and y_t represents the volume of goods shipped using an emergency shipment arriving at moment t. Emergency shipments can be used to fly in additional aid goods when the demand is higher than expected and can therefore not be satisfied by the regular shipments coming in by air or ocean. The costs for shipping aid goods given in Equation 1 are built up out of shipping costs and unit costs. Shipping costs are $c_a > 0$ and $c_o > 0$ for shipping by air and ocean, respectively. Emergency shipments are always shipped by air and receive an additional cost multiplier to them of f, where $f > 1$. Regardless of shipping method, all goods are procured against unit cost $b > 0$. Demand is satisfied in Equation 2 by means of the three shipping methods, where we make the assumption that any leftover goods are not carried over through time to other moments. We justify this assumption by referring back to the "fire fighting" mentality present in the humanitarian world, where delivering aid as soon as possible to those in need is prioritised over careful consideration whether demand has actually been satisfied already. This would result in benefactors of aid either consuming more goods or storing goods in their own shelters whenever surplus aid is available. The demand for aid goods is defined for the total demand over period t, so that d_t when $t = 1$ represents all demand for aid during the first week after the disaster. If we were to allow the carrying over of surplus aid goods to the next period, we should also include some sort of holding costs for keeping the goods in stock at the disaster location. However, as we have no available data on holding costs, we are further discouraged from changing our assumption. Shipping methods have lead times λ_a and λ_o for air transport and ocean transport, respectively. As shipments can only be sent after a disaster occurs at time period 0, the decision variables are only defined for the periods after their respective lead time in Equations 3 and 4. We assume that $\lambda_a \leq \lambda_o$, and that $c_a \geq c_o$, which means that transport via air is quicker, but also more expensive than transport via ocean.

The optimal solution to this problem would be very straightforward save for the uncertain demand d_t, which is drawn from an unknown uncertainty set D. Sampling from an unknown uncertainty set means that we do not only not know the value of each d_t, but we also do not know what values d_t could attain and with which probability. This problem is therefore solved using forecasts for the periodical demand.

Assuming that forecasts change over time as more information about the disaster becomes available, we will denote the forecast at a certain moment s for the demand in period t by $p\big(s,t\big)$. This means that for instance at moment 0, when the disaster occurs, we forecast that the demand in the following period will be $p\big(0,1\big)$. The best use of these forecasts, given that we do not know the distribution of d_t, is by making sure that shipments leaving at moment s for period t are sized such that they fill the gap between what is already going to arrive for period t and what is estimated to be needed. This means that $x_{ot} = p\big((t-\lambda_o),t\big)$ so that the expected demand for period t is fully satisfied via ocean transport. In the case that the forecast for the demand for period t has changed since the ocean shipment left, the expected gap is filled using air transport, following $x_{at} \geq p\big((t-\lambda_a),t\big) - x_{ot} = p\big((t-\lambda_a),t\big) - p\big((t-\lambda_o),t\big)$. We use a greater-than-or-equal-to sign to allow x_{at} to be zero when the right side of the equation is negative. This means that any $p_{s,t}$ where s is negative is equal to zero as there can be no forecast for demand for a disaster that has not yet occurred. The third decision variable remains $y_t \geq d_t - x_{ot} - x_{at}$. If we were to allow for the retention of surplus goods over periods, we could adapt the shipments leaving after each period to take that surplus into account. This should decrease costs as fewer goods are shipped to disasters that are already stockpiling goods locally. As stated previously, however, we assume that there is too little data collection performed during aid distribution to do this effectively.

If we were to allow for the retention of surplus goods over period t to $t+1$, with a certain $z_t = x_{at} + x_{ot} + y_t + z_{(t-1)} - d_t$, we could adapt the shipments leaving after period t such that $x_{at} \geq p\big((t-\lambda_a),t\big) - x_{ot} - z_{(t-\lambda_a)}$ and $x_{ot} \geq p\big((t-\lambda_o),t\big) - z_{(t-\lambda_o)}$.

With this application of the demand forecasts, we can easily solve the problem in Equations 1 - 5 by adding the following inequalities:

$$x_{ot} \geq p\big((t-\lambda_o),t\big), \qquad t = \lambda_o,\ \ldots\ , T, \tag{6}$$

$$x_{at} \geq p\big((t-\lambda_o),t\big) - p\big((t-\lambda_o),t\big), \qquad t = \lambda_a,\ \ldots\ , T. \tag{7}$$

Using Equations 1 through 7, we will calculate the impact of accurate forecasts on transportation costs after a disaster in Section 4.1.

3.2 Spectral Clustering

Clustering is one of the applications of unsupervised learning in machine learning. When clustering, we assign data points to a given number of clusters in such a way that minimizes the total distance between the data points in each cluster. This is done by iteratively assigning data points to the cluster with the mean that is closest, and updating this mean with the newly assigned data points. It is therefore crucial to determine what distance, or dissimilarity, actually entails.

Spectral clustering algorithms (SCAs) make use of dissimilarity as a broader concept than just distance in order to cluster data points. As a result, spectral clustering algorithms are easy to implement, solved efficiently, and generally outperform their more traditional counterparts, according to Von Luxburg (2007). Notable for a SCA is that it captures non-convex data structures much better than traditional methods can. We will shortly walk through the mathematical concepts behind these kind of algorithms, as well as discuss the choices made in this thesis regarding SCAs.

In SCA, we decide to represent the dataset as an undirected weighted graph $G = (V, E)$, where V is

the set of vertices, or data points. The set E contains the edges between the vertices in V, and are used to represent the similarity between two vertices. The edge weight w_{ij} represents the similarity between data points i and j, where we must remember that $i, j \in V$ and $w_{ij} = w_{ji}$. For these similarity weights we use the Euclidian distance, however,as we want to mainly include information on clusters of data points, we only let edges exist between closely related vertices. To this end, we transform the graph into a similarity graph using the k-nearest neighbours method. What this does is only include the edge between two vertices i and j if either i is in the set of k-nearest neighbours of j, or vice versa. This k is set to 10 for our application. Nearest neighbours are decided upon based on Euclidian distance from a vertex. This helps us in keeping the weights within groups of related data points high, but between those groups low. Having defined the graph with vertices, edges and their associated weights, we move on to some of the mathematics that will eventually lead to our spectral clustering.

Ultimately, we want to create a normalized graph Laplacian matrix L on which we base our clustering. We define d_i as the degree of vertex i, or as the sum of the weights of the edges that are adjacent to that vertex, so $d_i = \sum_{j=1}^{n} w_{ij}$. This means that this definition of the degree of a vertex is different than is traditionally the case in graph theory. Additionally, we define the matrix D as the matrix with these degrees on the diagonal. W is the weight matrix, consisting of the elements w_{ij} defined earlier. Now we can combine these elements into the definition of a symmetric graph Laplacian matrix:

$$L_{sym} = I_n - D^{-1/2} W D^{-1/2} \tag{8}$$

I_n here is the identity matrix of size n, the size of our dataset. The matrix $D^{-1/2}$ exists as D only has non-zero elements on the diagonal. We further know that L_{sym} is positive semi-definite, which means that the smallest possible eigenvalue is 0. Each of the eigenvectors of L_{sym} consists of n elements, each element corresponding to one of the n data points in our dataset. We take the first m eigenvectors of L_{sym}, the eigenvectors for which the eigenvalues are smallest. These eigenvectors are collected columnwise into a matrix Z of size $m \times n$, where each row corresponds to a data point. We finally apply k-means clustering on each of the n rows of Z. This k corresponds to the number of clusters we would like to generate. This clustering of the n data points is presented as the final result of the SCA.

3.3 Neural Networks

Neural networks are examples of supervised learning, making use of many inputs and combining those both linearly and non-linearly to estimate values of the output. They can be used for purposes such as regression models, which we will be doing in this paper, or classification models. Within each type of model, many different implementations have been researched based on the type and availability of data, as well as on expected underlying structures in the data.

In this section, we discuss the type of neural network that we will be using during this thesis, to which we shall further on simply refer to as 'neural network'. We make use of a (deep) feed forward network, a fairly straightforward neural network when compared to other commonly used neural networks. The reason why we chose this network is because we do not yet have any underlying structure such as demand over time. In addition, the simplicity is useful considering that this is only an investigative approach into the application of neural networks in humanitarian logistics, where application-specific tricks are possibly unneeded. We introduce the structure of our neural network in Section 3.3.1, followed by an elaboration

on the learning component in Section 3.3.2.

3.3.1 Structure

While often seen as magical black boxes that somehow churn out models based on some data and input parameters, our neural network is actually quite logical, as we will hopefully show here. A small diagram of what our type of neural network generally looks like schematically is given in Figure 3. In this figure, we have an $n-$dimensional input, a single hidden layer consisting of m nodes, and a $l-$dimensional output. Often, we will also have a X_0 and a z_0 that function as an enabler for some form of bias. In the figure, however, they are not shown.

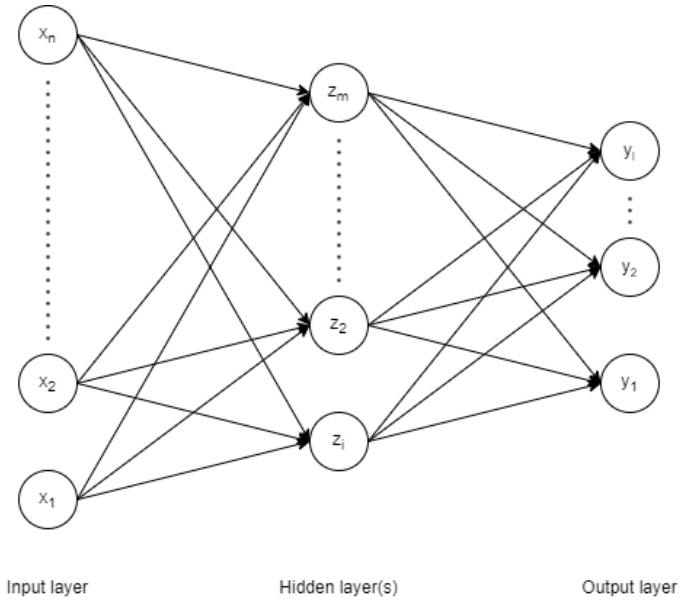

Figure 3: Schematic overview of a feed forward network

In the figure above, we see a network of nodes and directed arcs going from left to right. Each of the nodes is either an input node, a hidden node, or an output node. The input nodes are found in the input layer, and contain the value of a certain variable in a data point.

The hidden layers consist of hidden nodes, where data from the input layer is collected using some function. There is no golden rule for the number of hidden layers or hidden nodes in each of those layers. Appropriate values of such parameters are often found using algorithms, as will be explained in Section 3.4. The value of a node in a hidden layer is calculated using a function of the values that are fed through from the previous layer. This function uses separate weights for each node in the previous layer. For the value of a certain hidden node z_j in the first hidden layer, its value is calculated using

$$a_j = \sum_{k=1}^{n} \alpha_{jk} X_k + \alpha_{j0}, \qquad z_j = h(a_j), \tag{9}$$

where α_{jk} is the weight from node X_k in the input layer to z_j in the first hidden layer. The function $h(a_j)$ can be one of many, depending on the data structure in the input layer. In this thesis, we use the

logistic activation function, also known as the sigmoid function, given by

$$h(a_j) = \frac{1}{1 + e^{-a_j}}, j = 1, \ldots, m. \tag{10}$$

For other hidden nodes and subsequent hidden layers, the same functions as Functions 9 and 10 are used to calculate their value using different weights.

Finally, we find the output of a neural network in the values of its output nodes. In our thesis, we have a one-dimensional output as we look only to forecast a single variable. Similar to the calculation of the value of a hidden node, we calculate the value of our single output node as

$$y = \sum_{j=1}^{m} \omega_j z_j + \omega_0. \tag{11}$$

Following this structure, our neural network provides us with output based on an input vector. In our thesis, this means we get an estimation of the number of people in need of aid after a disaster, based on the data discussed in Section 2.

3.3.2 Backpropagation

The big question, however, is how we end up with the value of the weights used in the previous paragraphs. This is where the learning aspect of the neural network comes into play. First, we will need some definitions. We denote the set consisting of all weights by means of β. The training error within a neural network is given as

$$E(\beta) = \sum_{i=1}^{n} E_i(\beta) = \frac{1}{2} \sum_{i=1}^{n} (\hat{y}(X_i, \beta) - y_i)^2, \tag{12}$$

where y_i is the supervised value of the dependent variable of observation $i \in \{1, \ldots, n\}$, and $\hat{y}(X_i, \beta)$ is the estimated value of the dependent variable, based on the current set of weights β and the independent variables X_i corresponding to observation i. $E_i(\beta)$ is therefore the error of the neural network for a specific observation i. The way a neural network learns is by adapting β in order to minimize the total error $E(\beta)$. In this thesis, we perform this minimization by means of backpropagation.

Backpropagation searches for the local minimum of the error function by means of a gradient descent. A gradient descent updates the weights each iteration using the following relation:

$$\beta^{t+1} = \beta^t - \gamma \sum_{i=1}^{n} \nabla E_i(\beta^t), \tag{13}$$

where t is the current iteration, and γ is a learning rate. The minimum is local, as initial values for the weights β^0 are chosen randomly, meaning that certain neighbourhoods will not be reached. This is not too much of a problem, as a global minimum is likely to be an overfit to the specific data in the set of training observations.

What we need to do, according to Equation 13, is compute $\nabla E_i(\beta^t)$ at each iteration, until we either run out of iterations or have $\nabla E_i(\beta^t)$ close enough to zero to stop the algorithm. In order to calculate this, we feed our input vector X_i through the network using Equations 9 - 11. We evaluate the error at the output unit as $\delta_y = \hat{y}(X_i, \beta^t) - y_i$. Now we can move a layer back through the network, and evaluate the error at each of the nodes in that layer. We take a specific node s, for which we evaluate the error as

follows:

$$\delta_s = \frac{\partial E_i}{\partial a_s} = \sum_u \frac{\partial E_i}{\partial a_u} \frac{\partial a_u}{\partial a_s} = h'(a_s) \sum_u \delta_u \beta_{us}. \tag{14}$$

Here, nodes u are all the nodes in the layer from which we came. Because we are moving back through the network, this is the layer to the right of the current layer in Figure 3. a_s and a_u are found in Equation 9, and the function $h(a_s)$ is found in Equation 10. Finally, the β_{us} is the weight from node s to node u at iteration t. With the relation given in Equation 14, we can backpropagate through all hidden layers in the network until we come to the input layer. With errors derived for all nodes except for the input layer, we can now evaluate the components of the gradient corresponding to the weight from node j to node s as follows:

$$\frac{\partial E_i}{\partial \beta_{sj}} = \frac{\partial E_i}{\partial a_s} \frac{\partial a_s}{\partial \beta_{sj}} = \delta_s z_j, \tag{15}$$

with z_j as defined in Equations 9 and 10. Now we can update the weights in β each iteration, until the training of the model is completed by reaching a certain number of iterations or by getting all the components of the gradient under a certain threshold.

3.4 Variable and Parameter Selection

In order to train a neural network optimally, many choices have to be made. We already touched upon some of these choices in the previous subsection, such as learning rate, maximum number of iterations, gradient threshold, and the starting values of all the weights. While these choices are interesting and meaningful, the focus of this thesis is to determine if and how machine learning can be used to forecast the demand for humanitarian aid after a disaster, and not how to perfect those machine learning models. Therefore we will not be experimenting with the values mentioned above, but we will set them to values that have often been used in the world of machine learning.

Instead, we would like to focus on selecting the best variables and - in the case of neural networks - the best composition of the hidden layers. Variables should be selected carefully as we would like to only include those that are reliably relevant. The risk of overfitting a model to its training data increases as more variables are added to the model. The composition of the hidden layer is also crucial in a neural network. A neural network with only a single hidden layer or very few hidden nodes is less capable of expressing nonlinear relationships between dependent and independent variables. A neural network with too many hidden layers and nodes will likely overfit to the training data and therefore perform worse on data outside of the training set. We will elaborate further on the performance and comparison of models in Section 3.4.1, after which we introduce and explain the algorithms we developed to select variables and parameters in Section 3.4.2.

3.4.1 Performance Measurement

When evaluating a model, we would like to see how a model performs in a situation that they have never seen before. To that end, we make use of K-fold cross-validation. This method initially splits the dataset in a training set and a testing set, where roughly 80% is for training and 20% for testing. The training set is then divided into K disjoint folds, each of the same size. One such fold is assigned to be the validation set, and the rest of the training set is used to train the model using the settings we want to evaluate. The model is validated using the validation set, yielding a measurement of performance. This is

repeated K times so that each of the K folds has been the validation set once. The average performance of these K folds is taken as the performance of the settings used to train the models. We repeat this K-fold cross-validation for each of the settings we want to evaluate. Using the best settings, we train a model on the complete training set, after which we test the model using the testing set giving us the final performance measurement of the model. When ultimately deploying the model, we use all available data for training, not just the training set.

Having mentioned performance measurement multiple times, some definitions are in order. In this thesis we use two performance measurements, the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). For a model i with parameters β_i, and a testing set consisting of m observations, these performance measurements are given by

$$RMSE_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\Big(\hat{y}(X_j,\beta_i) - y_j\Big)^2}, \tag{16}$$

$$MAE_i = \frac{1}{m}\sum_{j=1}^{m}\Big|\hat{y}(X_j,\beta_i) - y_j\Big|, \tag{17}$$

where $\hat{y}(X_j,\beta_i)$ is the estimated value of the dependent variable of observation j using model i, and y_j is its actual value. Using these performance measures, we can easily evaluate models using data that was not included in the training itself. The RMSE is often used in modelling as errors are punished quadratically, meaning that relatively small errors are more easily forgiven than very big estimation errors. This seems logical, as we would prefer a model that is always close over a model that is often exactly right and sometimes horribly wrong. However, due to the structure of our dataset with many relatively low dependent variable values combined with a handful of very large values, this performance measure greatly rewards the models that accurately estimate the very large values whilst performing poorly for the many low values. This data structure has been discussed in Section 2.4. During the variable and parameter-selection algorithms that are discussed later, this is especially problematic, as selecting variables that overfit to the training data limits the search space of the algorithm. The MAE, on the other hand, penalizes all mispredictions equally. While this is not ideal for evaluating the final performance of a model, it does seem better for the variable and parameter-selection algorithms, as the MAE is more likely to favor variables that are generally good rather than specifically good initially. Further on in the algorithm, variables that are specifically good can be included without limiting the search space as much.

Therefore, we will be using both RMSE and MAE as performance measurements in the variable and parameter algorithms. The RMSE is more likely to identify variables that explain observations with a large number of affected people, whereas the MAE is more likely to identify variables that explain small differences between observations. This balance should provide us with a mixed bag of models during the final model evaluation.

We should be aware that the MAE is also likely to favor models that are trained and tested using the logarithmic dataset over models that are trained and tested using the regular dataset, due to the structure of the dataset. As described in Section 2.4, the dataset consists of many observations with a small value for the dependent variable and a few with a very high value for the dependent variable. If we take this into the extreme, we could say we have one observation with a very high value for the

dependent variable, which we will call x, and $m-1$ observations with near-zero values for the dependent variable. Using this, we can prove that in the case of a zero variable model, the MAE for the regular dataset can be twice the MAE for the logarithmic dataset. The MAE for the regular dataset when using the assumptions mentioned above resolves as follows using $\hat{y}_r = \frac{x}{m}$:

$$MAE_r = \frac{1}{m}\sum_{j=1}^{m}\left|\hat{y}_r - y_j\right| = \frac{1}{m}\Big(\left|x - \frac{x}{m}\right| + (m-1)\left|\frac{x}{m}\right|\Big) \tag{18}$$

$$= \frac{1}{m}\Big(\frac{m-1}{m}x + \frac{m-1}{m}x\Big) = 2\frac{m-1}{m^2}x, \tag{19}$$

and the MAE for the logarithmic dataset resolves as follows using $\hat{y}_l = \exp\left(\frac{log(x)}{m}\right)$

$$MAE_l = \frac{1}{m}\sum_{j=1}^{m}\left|\hat{y}_l - y_j\right| = \frac{1}{m}\Big(\left|\exp\Big(\frac{log(x)}{m}\Big) - x\right| + (m-1)\exp\Big(\frac{log(x)}{m}\Big)\Big) \tag{20}$$

$$= \frac{1}{m}\Big(x - \sqrt[m]{x} + (m-1)\sqrt[m]{x}\Big) = \frac{1}{m}x + (m-2)\sqrt[m]{x}. \tag{21}$$

With a large enough x and m, this means that the MAE for the logarithmic dataset can get up to half the size of the MAE for the regular dataset as the $\sqrt[m]{x}$ term vanishes. As our dataset has characteristics like the assumptions made for the result, we should expect a much lower MAE for the models trained and tested using the logarthmic dataset than for the models trained and tested using the regular dataset.

The RMSE, on the other hand, is quite similar between the two datasets, with the RMSE for the logarithmic dataset being slightly higher than the RMSE for the regular dataset due to a few $\sqrt[m]{x}$ terms. This, in combination with the fact that RMSE rewards models that perform consistently, leads us to use the RMSE when selecting our final model in Section 4.4.

3.4.2 Algorithms

With the concept of K-fold cross-validation and the different performance measures explained, we can introduce the algorithms that we use to select the variables for linear models and neural networks, and to select the best hidden layer composition for the neural networks only. Both algorithms make use of K-fold cross-validation, and the variable selection also makes use of forward stepwise selection. Instead of evaluating the performance measure for every possible set of variables, we build up the variable set using the variable that performs best. This saves computation time as the number of possible variables increases, at the cost of precision. However, we expect to find the optimal set of variables this way. This cannot be guaranteed, however, and therefore this algorithm is best described as a heuristic, rather than as a sure method of finding the variables that correspond to the best fit to the training data. In order to guide the algorithm in the right direction, we can select variables at the beginning to be automatically included. This not only increases the speed of the algorithm, but should also decrease the likelihood of overfitting to training data. If the variable-selection algorithm works well, however, there should be no need for pre-selecting variables as they will be selected by the algorithm. Algorithm 1 concerns variable selection, whereas the parameter selection is performed in Algorithm 2.

Following the pseudocode given in Algorithm 1 on the next page, we will walk through the steps that the algorithm takes to find the best set of variables. We give the inputs in lines 2-6, the big M there could be a multiple of the performance when using a model without variables, for example. The

Algorithm 1: Variable selection

```
1  inputs:
2  a vector of preselected variables S,
3  a set of possible variables V,
4  a set of training data T,
5  the number of folds K,
6  a big M that gives an upper bound to the performance measure.
7
8  initialize vector P to store performances.
9
10 while set V is not empty do
11 |
12 |  set bestInd to 0.
13 |  set bestPerf to M.
14 |  generate a set of parameters A.
15 |
16 |  for indicator ind in V do
17 |  |
18 |  |  set tempPerf to 0.
19 |  |
20 |  |  for k in {1,...,K} do
21 |  |  |
22 |  |  |  set train[k] as observations in T not in fold k, variables in S ∪ ind.
23 |  |  |  set validate[k] as observations in T in fold k, variables in S ∪ ind.
24 |  |  |  train models using train[k] and parameters A.
25 |  |  |  validate models using validate[k].
26 |  |  |  set p to the best performance of validated models.
27 |  |  |  increase tempPerf by p.
28 |  |  |
29 |  |  end
30 |  |
31 |  |  if tempPerf < bestPerf then
32 |  |  |
33 |  |  |  set bestPerf to tempPerf.
34 |  |  |  set bestInd to ind.
35 |  |  |
36 |  |  end
37 |  |
38 |  end
39 |
40 |  add bestInd to S.
41 |  save bestPerf to P.
42 |  remove bestInd from V.
43 |
44 end
45
46 return optimal vector of variables from S, based on best performance in P.
```

set of preselected variables S could be empty as well, but should always be given as we use that vector to return the best set of variables. In line 8 we initialize the vector in which we save the performances

of each of the variable sets in S. Ultimately, the n-th value in P corresponds to the first n variables in S. In line 10 we choose to continue until the set V is empty and we cannot add any more variables, and in each iteration we reset the best variable found and corresponding best performance in lines 12-13. If applicable, we generate a set of parameter settings in line 14 to ensure that all variables are tested using the same parameters, whilst not having to test all possible parameters. We iterate over all possible variables in V, looking for the performance of the model should we add that variable to S in lines 16-38. In lines 20-29, the K-fold cross-validation takes place with $train[k]$ and $validate[k]$ being different for each value of k. We use the different parameters generated in A for training in line 24, giving us multiple models. Validating these models, as stated in line 25, simply means to try to forecast the dependent variable of the unseen data points in the validation set. This gives us a best performance p, which is measured by means of the MAE or RMSE in this thesis. This means that the performance in each of the K folds can be derived with different parameters, but that each prospective variable in V uses the same parameters. When increasing $tempPerf$ in line 27, we should divide the final performance measurement by K to get the average performance over K folds. In lines 31-36, we update $bestPerf$ and $bestInd$ if the current performance measurement is better than the previous best. After evaluating the performance of all variables, we add the best one to S and remove it from V, saving the performance to P in lines 40-42. Having emptied our set V, we return the best performances and the corresponding vector of variables in line 46. With this output, we can simply select which variables from S to include in a model based on their performance. We can also use the order of the variables as input for parameter selection, which we will be doing next.

When working with neural networks, we also need to decide on the hidden layer composition that is most likely to be optimal for the set of variables that the variable selection returned. We refer to the decision concerning the hidden layer composition as parameter selection, as these are the only parameters that we will be tweaking. What a hidden layer does exactly has been discussed in Section 3.3.1, but for now we see it as a set of parameters that we can work with. In general, having more hidden layers and more hidden nodes inside those layers gives a model more flexibility, allowing it to better capture non-linearity or other relations within a dataset. Simultaneously, it risks overfitting the model to its training data which makes the model worse when handling previously unseen data. After some testing, we found that models with three hidden layers that performed excellently during parameter selection could perform extremely poorly when confronted with new data. Therefore, we decide to allow up to two hidden layers during parameter selection. Additionally, we do not allow more nodes in a hidden layer than the number of variables used for the model for the same reason. When a layer is not used, we denote it in the algorithm by means of a 0.

We will once again go through the pseudocode given in Algorithm 2, starting with the inputs in lines 2-4. The algorithm requires the vector of variables S returned by the variable-selection algorithm, along with other inputs that were used previously such as the training data and the number of folds. Performances are stored in matrix P in line 6, and in line 7 we save the number of variables in S. We iterate over all variable combinations that were optimal in the variable-selection algorithm in lines 9 and 11, creating C with size m as the current set of variables. In lines 13-33 we iterate over all possible hidden layer compositions we would like to research for this set C. We iterate up to m as we do not want more hidden nodes in a layer than variables in C. j iterates from 0 rather than from 1, as a hidden layer of

Algorithm 2: Parameter selection

```
1  inputs:
2  a vector of optimal variables S,
3  a set of training data T,
4  the number of folds K,
5
6  initialize matrix P to store performances.
7  set i to be the number of variables in S.
8
9  for m in {1,...,i} do
10 |
11 |  set C to be the set consisting of the first m variables in S.
12 |
13 |  for j in {0,...,m} do
14 |  |
15 |  |  for l in {1,...,m} do
16 |  |  |
17 |  |  |  set perf to 0.
18 |  |  |
19 |  |  |  for k in {1,...,K} do
20 |  |  |  |
21 |  |  |  |  set train[k] as observations in T not in fold k, variables in C.
22 |  |  |  |  set validate[k] as observations in T in fold k, variables in C.
23 |  |  |  |  train model using train[k] and (j,l) as hidden layer composition.
24 |  |  |  |  validate model using validate[k], yielding performance p.
25 |  |  |  |  increase perf by p.
26 |  |  |  |
27 |  |  |  end
28 |  |  |
29 |  |  |  add (m,j,l,perf) to P.
30 |  |  |
31 |  |  end
32 |  |
33 |  end
34 |
35 end
36
37 return optimal hidden layer composition from P.
```

size 0 is unused and we want to allow models using only a single hidden layer. For each composition we perform K-fold cross-validation in lines 19-27, using the hidden layer composition to train the model with the variables in C. In line 29, the performance and model composition are saved to P, the best of which are finally returned in line 37, terminating the algorithm.

Now that we have selected the best set of variables and the best corresponding set of parameters, our model is complete. After training the model using the complete training set, we can test it on the rest of the data points in the test set. This performance ultimately decides which model performs the best among all the models we created.

4 Computational Results

In this section we present all relevant results obtained in this thesis. The code with which these results are produced is written in the programming language R. All neural networks were trained and evaluated using the `neuralnet` package in R.

First, we will look at the impact of accurate forecasting in Section 4.1 before beginning the modelling. The results of the clustering required to fill in missing data is discussed in Section 4.2. We present the results of the variable and parameter selection in Section 4.3. The models that are created there are put to the test in Section 4.4, where the final model is used to measure its impact in a case study.

4.1 Impact Measurement

Before making any forecasts, we would like to quantify the importance of making those forecasts. This aims to give insight in why forecasting is important, and why we should work to get them to be as accurate as possible. In Section 3.1, the application of forecasts was introduced by means of solving an optimization problem with an unknown uncertainty set.

There, a few parameters were not yet defined that we will flesh out further here using data derived from Jahre et al. (2011) and internal data at Argusi, the internship company where this thesis is written. The time horizon T is set to 12 weeks, which means that t is defined in increments of one week. We take b, the price of aid goods, to be 803 USD per metric ton. The penalty for emergency transport is set to $f = 120\%$. Each calculation is made using a single supply depot, but we perform multiple calculations. We calculate two different scenarios, one using a supply depot that is fairly close by, and using one that is further away. The first scenario uses lead times $\lambda_a = 1$, and $\lambda_o = 2$. This means that a shipment via air that is ordered immediately when a disaster occurs will be able to help provide in satisfying demand that arises in week 1, and a shipment via ocean for demand in week 2. The costs in this scenario are $c_a = 38,916$ USD per metric ton and $c_o = 721$ USD per metric ton. In the second scenario, we have $\lambda_a = 1$, $\lambda_o = 6$, $c_a = 59,923$ and $c_o = 2390$. It should be noted that the costs of air transport increase by roughly 50% between the first and second scenario, whereas the costs of ocean transport increase more than threefold. This can be explained by the fact that air transport is very inefficient over small distances. Even though this data is representational for a single humanitarian operation, we cannot state that they are similar for other disaster responses where variables such as distance from supply ports to disaster area, aid good composition and local infrastructure are different.

The most important input of all are the values of d_t and of $p(s,t)$. As there is no data available for demand fluctuation over time, we can decide to simplify the demand to be the same every period of our time horizon T. This would mean that $d_t = d = 100$, and that $p(s,t) = p(s)$. As we would like to include a learning factor in the demand forecasts, we make sure that the forecasts are updated each time period to better resemble the actual demand. With this, we incorporate the effect of some communication up the supply chain as a result of having a deficit or surplus of aid goods in each period. Starting from period $s = 3$, we assume that $p(s) = \frac{1}{10}\big((12-s)\cdot p(0) + (s-2)\cdot d\big)$. The forecasts at period $s = 3$ therefore consist of 90% of the original forecast and 10% of the actual periodical demand. In reality, this learning effect will only marginally affect the total costs as we expect new knowledge to be implemented slowly during the response to a humanitarian crisis. The sooner this learning starts, however, the less

cost effective a good forecast would be relative to a bad forecast.

We mentioned that the periodical demand as introduced above is likely not how demand fluctuates after a disaster has occurred. A more realistic representation would be if we were to have a time series of the demand, where demand is high in the first weeks after the disaster and gradually lower as time progresses. Even though time series data does not exist, we can roughly simulate what this periodic demand would look like. We will assume that, based on Jahre et al. (2011), demand declines linearly during the first twelve periods, with $d_1 = 118$ and $d_{12} = 82$. When incorporating the learning factor, we should then assume that it is known that demand declines linearly with the slope described above. If we take z_t to be the factor by which some base demand is multiplied in order to get the demand at period t, we then get $p(0,t) = p(0) \times z_t$. Similarly, we would get $p(s,t) = p(s) \times z_t$, where $p(s)$ is unchanged from its definition given above.

With both a static and a variable demand over time, we can test both scenarios twice. The forecasts we would like to put to the test are $p(0) = 50, 75, 90, 110, 125$, and 150, of which the costs relative to a perfect forecast are presented in the table below:

$p(0)$	50	75	90	100	110	125	150
Relative costs static scenario 1	402	251	160	100	109	123	145
Relative costs static scenario 2	167	133	113	100	109	123	147
Relative costs variable scenario 1	377	239	155	100	109	123	146
Relative costs variable scenario 2	158	129	112	100	109	123	147

Table 1: The impact of initial forecasts on the cost of transportation of aid goods after an earthquake

In order to put the different scenario's into perspective, the costs with perfect forecasts for a static demand are 5.6 and 32 million USD for the first and second scenario, respectively. The costs associated with perfect forecasts for a variable demand are 6.3 and 36 million USD. Having a supply port closer to the disaster is therefore logically better for reducing transportation costs. Other findings, based on Table 1, are that forecasting too much demand is generally better than forecasting too little. In fact, we see that the increase in costs is smaller than the increase in forecast for both scenarios and both demand types. This is to be expected as forecasting too much will cause the operation to simply scale, that is to say a forecasts that is 10% too high will cause all transports to increase by 10% initially, before normalizing to the actual demand as time progresses as a result of learning more about the actual periodical demand. This effect is seen in both scenarios and with both types of demand. Forecasting too little demand, however, is far more costly. Due to the high costs of air transport and especially of emergency air transport, having to fly in extra unexpected aid goods causes a large rise in total costs when forecasts are too low. This effect is larger for the first scenario, due to the relatively higher costs of air transport compared to ocean transport there. Having a variable demand somewhat softens this blow, where we see a lower relative cost than for the static demand for all forecasts that were too low. This is due to a relatively smaller increase in costs for forecasting too low than for the perfect forecast. Absolutely, the cost of forecasting too low grew more than the cost of a perfect forecast. For the second scenario, however, the costs of forecasting too little demand is almost the same as forecasting too much, although the difference remains significant.

These results are of course subject to the specific parameter values and the simplification of the

transportation problem used here. Impact can be measured more realistically by expanding the transportation problem. This can for instance be done by allowing more supply depots and by introducing different commodities. A more detailed time frame, better understanding of how forecasts are adapted to actual demand and of the development of demand over time can also improve the measurement of the impact of forecasting on cost associated with humanitarian logistics. With the current parameters, however, we can assume that it is better to forecast too much demand than too little, whilst forecasting the exact right amount is optimal.

4.2 Clustering

In Section 2.3, we discussed how to handle missing values in our dataset for country variables using clustering. We decide to cluster countries that are alike to one another, and use the values in the rest of the cluster to reliably fill in missing variables. We use two clusterings and compare their performance when creating neural networks to select the final clustering that we will be using. We base the first clustering on geographical regions employed by the World Bank. We make the second clustering by means of SCA, which is described in Section 3.2, and the specific procedure for which we will explain below. The outcome of these clusterings can be found in Appendix B.

In order to cluster the countries in such a way that the clusters can be used to fill in all missing values, whilst keeping the clusters small enough to be distinct from one another, we developed the following procedure for the spectral clustering. Initially, a small selection of indicators that are widely available and likely to be good for the clustering is made. These indicators can also be found in Appendix B. Of these indicators, we use the average values over the years between 2000 and 2019. With these indicators, we perform a spectral clustering algorithm as described in Section 3.2 with 4 eigenvectors. Starting with a large number of clusters, we continuously lower the number of clusters until we find a clustering in which all missing values can be filled in. The largest number of clusters for which this works, is 18. In comparison, the number of geographical clusters used by the World Bank is 7.

In order to evaluate the performance of each clustering, we make use of the parameter-selection algorithm described in Section 3.4.2, with a few alterations. The neural networks were trained on the full set of 55 variables, and with only a single hidden layer. We used a 5-fold cross-validation, which we will be using throughout the rest of this thesis. The complete training set consists of the first 525 observations in our dataset, leaving the final 131 observations for the test set. Therefore, each fold in the training set contains 105 observations. The folds have been created using increments of 5, meaning that the first fold has observations $1, 6, 11, \ldots, 521$ and the fifth fold has observations $5, 10, \ldots, 525$. Each clustering is evaluated by means of the RMSE and MAE of their best-performing neural network. The results are presented in the table below.

Clustering type	RMSE	MAE
Regional	126007	32358
Spectral	112778	27348

Table 2: Performance of the different clusterings for a 55-variable neural network

In Table 2 we can see that the regional clustering yields an RMSE of on average 10% more than the

spectral clustering, and an MAE that is 15% higher. From this, we can conclude that the application of machine learning to filling in missing data points by means of spectral clustering has a huge impact on the performance of our neural network. As a result, we decide to only use the data generated using spectral clustering for training and testing our models in the remainder of this thesis.

4.3 Variable and Parameter Selection

Having decided on the clustering we will be using, we can start creating models using the methodology described in Sections 3.3 and 3.4. We would like to create multiple models, each using different methods or data, in order to have a large diversity in models when testing them on new data. There are four binary choices to be made, which means we will have 16 models to send to testing.

The first choice is between the use of a linear model or a neural network. Linear models can be seen as a traditional style of forecasting, whereas the neural networks come from in the field of machine learning.

The second choice regards the performance measurement. Both linear models and neural networks are created by optimizing performance, so the choice between RMSE and MAE as the performance measurement works for all models. Different performance measurements are likely to influence whether a model is going to either overfit or generalize.

The third choice is between the regular dependent variable or the natural logarithm of the dependent variable. The difference between these two and their effects is described in detail in Section 2.4. The performance of models using the logarithmic dependent variable is judged after transforming forecasts back to their original scale. It is possible for the models using the regular dependent variable to forecast demand values lower than zero, even though this negative demand is impossible in reality. It is therefore possible to truncate these forecasts to be zero when negative, which should increase the performance of models. This comes at a cost however, as it discards valuable information about the quality of forecasts lower than zero which should allow us to select better variables and parameters. After careful consideration and testing, we decided to keep the negative forecasts when selecting variables and parameters, but truncate the forecasts when selecting the final model. With this choice we use all available information on performance during model creation, but mirror how real-life forecasts would work when applying the model to unseen observations.

The final choice is whether we want to pre-select certain variables. In order to help the variable-selection algorithm, we can select variables for which we have some intuition that they are likely to be good. The variables that we would like to pre-select are no. 51, 52, 53, 54 and 55, found in Table 16 in Appendix A. These variables concern the magnitude of the earthquake, the number of people living within a 25 km, 100 km and 250 km radius of the epicentre, and the focal depth of the earthquake. These variables have been chosen because, intuitively, they should always influence the number of people that are affected by an earthquake. In addition, we can be fairly certain that these variables represent the actual situation on the ground due to the reliability and precision of seismic data. Variables with information on the state of the country in which an earthquake occurs, however, are not necessarily relevant to the number of people affected by a disaster as they do not fully correspond to the situation in the struck area. These pre-selected variables should therefore be able to capture underlying structures in the data rather than overfitting to possibly unconnected variables. Whether these variables on country data are relevant should be concluded from the results in the following sections.

Regardless of the choices we make, we need to set a benchmark in order to properly compare the different models. The benchmark we decided upon is the performance of a very simple naive model, namely a model based on the regular dependent variable without any independent variables. A model without any variables will always suggest the average of the dependent variable over its training set as the most likely value of the dependent variable for a new data point. To be able to compare models to this benchmark, all models should train and validate themselves using the same observations. Therefore, we use the cross-validation set-up as described in Section 4.2 for the benchmark and all following models. With this setup of the 5-fold cross-validation, the benchmark - to which we will refer as model 0 - performs as follows:

Model ID	RMSE	MAE
0	93294	26779

Table 3: Performance of benchmark model

We can already see from comparing Tables 2 and 3, which were made using the same cross-validation, that a model without any variables is likely to outperform a model with all 55 variables. This should already warn us of the risk of overfitting when using this dataset. We will compare models that are made using the RMSE as performance measure with the RMSE of our benchmark, and models using the MAE with the MAE of our benchmark.

We will first branch on the model type we will be using, discussing the linear models in Subsection 4.3.1 and the neural networks in Subsection 4.3.2. We then branch on the dependent variable used, first covering the regular and then the logarithmic variant. Thirdly we branch on performance measurement, and lastly on whether we pre-select variables. For an overview of the choices for each model, we refer to Table 19 in Appendix C.

4.3.1 Linear Model

For all linear models, only a variable-selection procedure is performed, without any parameters needing to be selected. We employ the forward stepwise variable selection described in Section 3.4.2, using RMSE and MAE to rate the performance of each model. Tables 4-7 show these performances using different dependent variables and different performance measures. Within each table, the performances of the model that uses the variables found by the variable-selection algorithm using pre-selected variables and of the model that uses the variables selected by the variable-selection algorithm without any pre-selected variables are presented. In the tables below, only the number of variables selected is specified; Appendix D specifies which variables have been actually selected. Finally, models that perform worse than model 0 are given negative percentages for their improvement.

No. of variables	RMSE	% improvement over benchmark	Model ID
8	97623	-4.64	1
1*	93226	0.07	2

* = derived without pre-selected variables

Table 4: Effect of variable selection procedure using RMSE on the linear model with regular dependent variable

No. of variables	MAE	% improvement over benchmark	Model ID
6	29760	-11.13	3
4*	26575	0.76	4

* = derived without pre-selected variables

Table 5: Effect of variable selection procedure using MAE on the linear model with regular dependent variable

No. of variables	RMSE	% improvement over benchmark	Model ID
37	91854	1.54	5
45*	91837	1.56	6

* = derived without pre-selected variables

Table 6: Effect of variable selection procedure using RMSE on the linear model with logarithmic dependent variable

No. of variables	MAE	% improvement over benchmark	Model ID
31	15040	43.84	7
22*	15018	43.92	8

* = derived without pre-selected variables

Table 7: Effect of variable selection procedure using MAE on the linear model with logarithmic dependent variable

In Table 4 we see that model 1, that was created using the pre-selected variables, performs worse than the benchmark after picking up three more variables. Strikingly, model 2 seems optimal with only a single variable included, performing very slightly better than the benchmark without any variables.

In Table 5 we see similar things, with an even worse performance when using the pre-selected variables in model 3. Model 4, without pre-selected variables, settles for 4 variables and performs slightly better than the benchmark.

Now using the logarithmic dependent variable, we see in Table 6 that model 5 performs better than its counterpart in Table 4. It uses 37 variables and performs slightly better than the benchmark. Model 6 uses even more variables, including almost all variables at 45 in total. Its performance is similar to model 5, and it should be noted that they both outperform their counterparts using the regular dependent variable in Table 4.

As proven in Equations 18 - 21, the MAE is much lower for models using the logarithmic dependent variable than for the models using the regular variable. As a result, models 7 and 8 perform 43% better than the benchmark. Model 7 uses 31 variables for this, whilst model 8 picks up 22 variables.

We can conclude from these tables that the MAE indeed heavily favors the logarithmic dependent variable over the regular dependent variable. In addition, we see that the variable-selection algorithm does very little for models using the regular dataset, but selects a lot of variables for models that use the logarithmic dataset. It is, however, too early to draw conclusions on which modelling choices are better. That will be done when forecasting new observations in Section 4.4.

4.3.2 Neural Network

This section concerns the models we create based on neural networks. As these are more complex and offer more choices than the linear models, we must first elaborate on a few things before diving into the results. The methodology behind neural networks is explained in Section 3.3, so we will not go into detail about them here. We should, however, specify what other choices have been made in creating our neural networks. Choices such as the network structure and the activation function have been discussed, as well as the backpropagation procedure that is used to train the models. Amongst other parameters that we had not yet decided upon is the learning rate, which is crucial in updating weights. We set $\gamma = 0.1$. The maximum number of iterations is set to 100,000 for all neural networks. The gradient threshold, which decides whether we want to terminate the learning algorithm, is set to 0.01. The starting values of the weights are pseudorandomized each time by setting the same seed each time a neural network is trained. These choices are based on some initial testing, where our goal was to make sure learning could complete without massive overtraining, which is in line with the research questions of this thesis. In future research, these parameters could also be optimized using cross-validation and other machine learning techniques.

In order to select the variables and parameters that are most likely to be good, we make use of Algorithms 1 and 2 that are introduced and explained in Section 3.4.2, for which we will specify some design choices. We decided to try three different hidden layer compositions for A, where each composition is a single layer using a random integer lower than the number of variables currently in S. In addition, we set K to be 5 for both algorithms. The results from Algorithm 1 are used as the input for Algorithm 2, the output of which will be presented in the tables below. The compositions of the chosen variable sets are presented in Appendix D, as listing them all would be too cumbersome here.

No. of variables	Hidden layers composition	RMSE	% improvement over benchmark	Model ID
20	2 1	91460	1.97	9
17*	10 1	91477	1.95	10

* = derived without pre-selected variables

Table 8: Result of variable and parameter selection using RMSE for neural network with regular dependent variable

No. of variables	Hidden layers composition	MAE	% improvement over benchmark	Model ID
20	14 1	20587	23.12	11
16*	5 2	23098	13.75	12

* = derived without pre-selected variables

Table 9: Result of variable and parameter selection using MAE for neural network with regular dependent variable

No. of variables	Hidden layers composition	RMSE	% improvement over benchmark	Model ID
10	4 -	82142	11.95	13
4*	2 -	59618	36.10	14

* = derived without pre-selected variables

Table 10: Result of variable and parameter selection using RMSE for neural network with logarithmic dependent variable

No. of variables	Hidden layers composition	MAE	% improvement over benchmark	Model ID
5	2 2	15035	43.86	15
8*	4 3	13193	50.74	16

* = derived without pre-selected variables

Table 11: Result of variable and parameter selection using MAE for neural network with logarithmic dependent variable

Aside from finalizing the models we would like to use to forecast unseen data in the test set, we can also make some observations on the effect of several model design choices on the ability to forecast unseen observations within the training set from Tables 8 - 11.

All models seem to outperform the naive model 0, either by small or large margins. These models outperform the linear models presented in Tables 4 - 7. It is interesting to see that the models using the logarithmic dataset perform much better than their counterparts using the regular dataset. In addition, the logarithmic models select fewer variables than the regular models. We also see that the MAE for models using the logarithmic dataset is much lower than the MAE for the regular dataset, as expected according to Section 3.4.1. There is not yet any conclusive evidence that the pre-selection of variables is better for creating a neural network. One model, number 15, has elected to only use the pre-selected variables and nothing else, which suggests that these pre-selected variables are relevant even on their own. On the other hand, it could also mean that locking in the pre-selected variables leaves no room for other good combinations of variables. Model 14 performs unexpectedly well when compared to other logarithmic models using RMSE, in addition to having a very low number of variables and hidden nodes. This might be due to a stroke of luck when tested on the training data, and whether this model is overtrained or not shall become clear in Section 4.4.

When looking at the hidden layer compositions of the created models, we find a pattern of one big layer followed by a smaller layer for the models created using the regular dependent variable. The logarithmic models seem to prefer either a single small layer or two small layers.

From Appendix D we can conclude that the pre-selected variables are still selected quite often by models that did not pre-select them, being selected 7 times out of a possible 20, whereas the other

variables are selected 38 times out of the possible 200 times in those same models. There are multiple other variables that seem to do well, for instance the total population of a country when there are pre-selected variables, and the employment rate when there are no pre-selected variables present. There is, however, no combination of variables that is always selected.

4.4 Demand Forecast

Using the variable and parameter setups from Section 4.3, we have eight linear models and eight neural networks ready to try to forecast the demand of previously unseen data points below. First, we select the final model based on their performance in forecasting the observations in the test set after being trained on all observations in the training set. Then, the winning model is used to showcase the application of a forecasting model on a real-life disaster, comparing our forecast to the estimates that were made back then.

4.4.1 Final Model Selection

In order to select a final model, we require all models to be tested on data not seen before. All models are trained on the complete training set of 525 observations, using the variables and parameters selected in the previous section. The neural networks are trained using the same settings as in Section 4.3.2, except for the random initial weights as we noticed that these random weights could hugely alter the outcome of a model. Therefore, we decided to create ten neural networks with random starting weights for each model, and taking the median of those forecasts as the final forecast.

In this final test, we will find out whether models that performed well on the training set will also perform well on the testing set. The model that performs best here should be a good choice for forecasting humanitarian aid demand in emerging disasters. We test all finalized models on the remaining 131 data points of the dataset, recording the RMSE of their forecasts. In the table below we once again give the performance of the zero-variables model, which uses the historical mean for all forecasts, as a comparison.

Model ID	RMSE	% improvement over benchmark
0	23213	0.00
1	30545	-31.58
2	23212	0.00
3	29803	-28.39
4	25316	-9.01
5	23565	-1.52
6	22380	3.59
7	22647	2.44
8	22752	1.99
9	22029	5.10
10	22378	3.60
11	25082	-8.05
12	21973	5.34
13	22800	1.78
14	22804	1.76
15	22806	1.75
16	22766	1.93

Table 12: Performance of models in forecasting unseen demand measured by means of RMSE

The first thing that springs to attention, considering the results presented above, is that the performances of models in Table 12 are a lot worse than their performances in Tables 4 - 11 when compared to their respective benchmarks. Only models 6, 9 and 10 perform slightly better than during the variable and parameter selection phase, albeit only slightly. Models 1 and 3 already performed poorly before, but perform even worse on previously unseen observations. These correspond to linear models created using pre-selected variables and the regular dataset. From this, we can conclude that this combination generally performs poorly.

The overall drop in performance is partially due to the absence of the combination of MAE and logarithmic dependent variables, which caused excellent performances in the variable and parameter selection phase. A more generally applicable explanation can be found in the difference in variance between the training and the testing data. The testing data has a variance many times smaller than the variance of the training data, which means that the testing data can be forecast much better than the training data when using a historical average. As a result, the historical average works much better than before, and the created models perform relatively worse. This is a sign of overfitting to the training data by all models we created. A possible approach to this problem in further research would be to abandon the current chronological order between training and testing set, and rather find a distribution between testing and training datasets such that the variance is more even.

We would like to draw conclusions on the choices made when creating the models. The specifications of each model are given in Tables 4 - 10, and are also provided more clearly in Appendix C.

The first choice was regarding model type, where we see that the neural networks outperform the linear

models in both their relative performance and their overall consistency. The linear models perform worse than the naive model on average, whereas the neural networks on average outperform the naive model. This is a trend that we also saw forming in the variable and parameter selection earlier. Interestingly, the combination of regular dataset, MAE and pre-selected variables performs poorly regardless of model type in models 3 and 11.

When considering the different datasets used, we see that the logarithmic dataset performs more consistently than the regular dataset, especially when looking at the use of the regular dataset for linear models and the use of the logarithmic dataset for neural networks. This can be due to the tendency of the logarithmic dataset to forecast more average demands, whereas models using the regular dataset are more easily skewed by large observations. We also conclude that the logarithmic dataset yields, on average, the best models due to the inconsistency in the performance of the regular dataset. Strikingly, the three best performing models, models 9, 10 and 12, are all derived using the regular dataset and are based on neural networks. However, due to model 11 performing very poorly we cannot state that this combination of dataset and model type is the best solution.

Another modelling choice was regarding the performance measure. There seems to be little difference between models judged on their MAE and models judged on their RMSE. In the already promising combination of regular dataset and neural network we see that the RMSE outperforms the MAE. The same can be said in the combination of regular dataset and linear model, but the difference there is less meaningful as these performances are already poor. It does suggest that the RMSE outperforms the MAE, although this effect can be explained by the fact that all models are measured by their RMSE here. Therefore, we cannot state whether the use of either of the RMSE or MAE is better when selecting variables and parameters.

Finally, we would like to discuss the choice of whether we should pre-select variables. For ease of reading, all odd-numbered models are based on pre-selected variables and all even numbers are without pre-selected variables. We see that pre-selection of variables causes worse models on average, especially for the linear models using the regular dataset, and are responsible for 4 of the 5 negative performances relative to the benchmark. In the positive cases, however, we see that neither is clearly better than the other when all other choices remain the same. So even though there is some intuition that the pre-selection of several variables is promising, we do not see this clearly represented in the above results. The variables that were pre-selected are often selected by the other models that did not automatically include them. Therefore, our intuition is correct in the sense that the pre-selected variables are relevant to forecasting demand for humanitarian aid. Our intuition is incorrect, however, in thinking that selecting them automatically is beneficial for the model. The variable-selection algorithm selects the best variables regardless of whether they are one of the five pre-selected or not.

Overall, we should conclude based on the performances in Table 12 that model 12 is best suited to forecast previously unseen observations, using a neural network trained on the regular dataset by means of their MAE without any pre-selected variables. The combination of neural networks trained on the regular dataset by means of their RMSE is the most promising for future research into demand forecasting for humanitarian aid using machine learning based on their good performance and consistency.

4.4.2 Case Study

Now that we know which model is most likely to produce good results when applied to unseen observations, we can also show what the impact is of a good forecasting model. For this, we use the theory discussed in Section 3.1, and some of the parameters given in Section 4.1. Specifically, we would like to compare the performance of our model to a well-known earthquake from recent history. In January 2010, an earthquake with magnitude 7.0 ravaged the capital of Haiti, Port-au-Prince, displacing millions of people. International response was immense as donations soon exceeded the initial estimate of aid required in the first twelve months after the disaster, made by the United Nations Office for the Coordination of Humanitarian Affairs (UNOCHA), of 575 million USD on January 16th. Five weeks after the earthquake occurred, the estimate was nearly tripled to 1.4 billion USD after which this estimate remained unchanged (UNOCHA, 2010). Based on the findings in Section 4.1, we expect that - according to our simplified problem - a more accurate forecast would have meant that more of the donated funds could have been allocated to those in need, rather than being used for logistics.

Using the technique described in Section 2.4, we take the actual number of affected people by the 2010 earthquake in Haiti to be 1,691,208, or 1.7 million. This number is lower than the number of affected people as reported by the UNOCHA, which is around 3 million. It is, however, consistent with how the dependent variables for the other observations in the dataset have been generated, and therefore we take 1.7 million as the actual number of affected people. We assume this number to correspond to the revised appeal of 1.4 billion USD, which means that the initial estimate of 575 million USD would correspond to a $p(0)$ of 674,595 people affected. This corresponds to a forecast of 40 % of the actual demand.

To put our selected model - which we name Machine learning #12 - to the test, we also forecast the demand for the Haiti 2010 earthquake. This earthquake was intentionally left out of the initial dataset of 656 observations. Using the same method as in the previous section, where we trained ten neural networks and chose the median of those forecasts as the actual forecast, we estimate the number of affected people to be 2,418,218. We noticed that the method of taking the median of the set of forecasts as described in Subsection 4.4 was effective, as the other forecasts ranged from near-zero to seven million affected people. With an actual number of people affected of 1.7 million, our model forecasts 143% of the actual demand. Even though this forecast is much too large, it is relatively close to the 3 million as reported by the UNOCHA. We summarize the different forecasts in the table below.

Forecast name	Actual	UNOCHA	Machine learning #12
Forecast no. of affected people	1,691,208	674,595	2,418,218
% of actual no. of affected people	100	40	143

Table 13: Summary of forecasts for Haiti 2010 earthquake

For the comparison between the costs using the forecast from the appeal, a perfect forecast, and our forecast, we use the parameters for scenarios 1 and 2 as described in Section 4.1, considering both a static and a variable demand structure. These scenarios correspond to a single supply port where the lead time over ocean is either two or six weeks respectively, with all other parameters based on real-life data. For visualisation, these scenarios correspond to supplying Haiti from ports in Panama and Denmark, respectively. The forecast for the total number of affected people is translated into the $p(0)$ in terms

of weekly required aid goods in tonnes, where an average of 50 kg per week can help one person with shelter, medicine and food during the first twelve weeks after a disaster, according to Jahre et al. (2011). The costs according to these scenarios for the different forecasts is as follows:

Forecast name	Actual	UNOCHA	Machine learning #12
$p(0)$ in tonnes	253,681	101,189	362,733
Costs static scenario 1 in billion USD	4.78	22.1	6.63
Costs static scenario 2 in billion USD	27.6	49.7	38.6
Costs variable scenario 1 in billion USD	5.36	23.2	7.48
Costs variable scenario 2 in billion USD	30.3	51.5	42.6

Table 14: Estimated logistics costs for different forecasts and scenarios

From Table 14, we see that the perfect forecast performs best for both scenarios, and that the second scenario is about 6 times more expensive than the first scenario with the perfect forecast for both demand structures. Our forecast leads to a cheaper solution than the forecast by UNOCHA for all scenarios, taking 70 % and 22 % off the costs of the UNOCHA forecast in scenarios 1 and 2 using static demand, respectively. When using variable demand, this gap is slightly smaller as our forecast is now 68 % and 17 % cheaper for scenarios 1 and 2. We still provide a more expensive solution than a solution based on a perfect forecast, with an increase in costs of 39 % and 40 % for scenarios 1 and 2 using the static demand and an increase of 40 % for both scenarios with variable demand. These findings are consistent with those we found in Section 4.1, where hypothetical forecasts were assessed for both scenarios.

It should be noted that the expected costs, even when forecast perfectly, shipped from a supply port close to the disaster area and assuming a static demand structure, are much higher than the 1.4 billion USD that were actually required. Whether this is due to parameters that are too pricey, or to incompleteness in the aid provided in 2010, we cannot say. Even though the current parameters are able to be used to show the impact of a forecast, further exploration of parameter values is necessary before the costs from Table 14 can be safely used.

With the current parameters we can not only conclude that our forecast comes closer than the official forecast, but also that the costs were much lower. When we would compare these costs to the forecast made by the naive model, the cost reduction would be even larger even though the comparison between our final model and the naive model in Table 12 would suggest otherwise. During the final model selection, an NGO that supplies aid goods to disaster areas might prefer a model that minimizes costs rather than a mathematical performance measurement. With a better understanding of the cost parameters associated with the transport and acquisition of aid goods, we could rate forecast models on their expected costs rather than other performance measurements in future research.

5 Conclusion and Recommendations

Conclusion

In this thesis, we set out to examine the possibilities of applying machine learning techniques to the forecasting of demand for humanitarian aid goods as a result of a sudden-onset disaster. Being able to forecast this demand accurately is of vital importance, as anticipating too much or too little demand will result in unnecessarily high costs of acquiring aid goods, storing them and transporting them to people in need in a timely fashion. We refer to this practice of getting aid goods to people affected by a disaster as humanitarian logistics. We discussed the current state of the fields of humanitarian logistics and of demand forecasting using machine learning in commercial fields, as well as a few applications of machine learning in the humanitarian world, based on recent literature in those fields. We found that, despite recent developments in research, the application of machine learning in the field of humanitarian logistics lagged behind that in commercial fields. This is partially due to a lack of data proficiency in humanitarian logistics, as a result of a vicious circle between poor data collection and the difficulty in efficiently applying the available data to crucial operations in humanitarian logistics. We decided to break this circle by applying several machine learning techniques in modelling the demand for humanitarian aid goods after the occurrence of an earthquake and measuring their effectiveness, even when working with sparse data. Earthquakes were chosen specifically as they are easily quantifiable and because their data is accurate and widely available.

We used data from three different main sources, consisting of data related to the earthquake itself, the number of people estimated to be living around the epicentre of the earthquake, and indicators regarding the state of the country in which the earthquake occurred that were likely relevant to the demand for aid goods, as explained in Section 2. However, many countries were missing data points for the last set of indicators. Using spectral clustering analysis, one of the machine learning techniques we introduced in this thesis, we clustered countries together in such a way that countries were comparable to other countries within their cluster, and distinguishable from countries from other clusters. Countries with missing data points for an indicator were then assigned the average values of their respective clusters for that indicator. The dataset filled using spectral clustering was compared to a dataset filled using a common geographical clustering, which it outperformed when used to train models for forecasting aid good demand. Using spectral clustering analysis in this fashion will help fill in incomplete datasets reliably, which is shown to improve the performance of applications using those datasets.

With the filled datasets we set out to build a forecast model for the demand for aid goods after an earthquake, where we decided which variables to include and with which parameters to train our model, if applicable. There are, however, many different ways of creating a forecast model. As we wanted to investigate to what extent the demand could be forecast using machine learning techniques when compared to traditional methods, we narrowed down the space of possible models using a few choices. These choices relate to model structure, data transformations, and the selection of parameters and variables. We tested both linear regression models and neural networks, both a regular and a logarithmic dataset, two different performance measurements, and whether or not to pre-select certain variables when creating

the model. These choices left us with 16 different models, for which we initially selected variables and parameters in Section 4.3, where we compared their performances in forecasting folds of the training set. Here we found that neural networks outperform linear regression models, and that models created using the logarithmic dataset outperform those that used the regular dataset.

Ultimately, we wanted to see how each model would perform when forecasting previously unseen observations. These performances were presented in Section 4.4, where once again was confirmed that neural networks performed better than the linear regression models. In addition, we concluded that the regular dataset produced the better models, but that the logarithmic dataset performed more consistently when creating neural networks. We could not find a significant difference between models created by means of RMSE and the models created with the MAE as performance measure. On average, it was better to forego pre-selecting variables and let the variable-selection algorithm do its job. This shows that the variable-selection algorithm works well independently. The best model we found was a neural network trained on the regular dataset by means of their MAE without any pre-selected variables, which performed more than 5% better than a naive model that we used as a benchmark. This relatively low improvement over the naive model is explained by the variance of the unseen observations, which is many times lower than the variance in the training data. This causes the naive model to perform relatively well.

Overall, we conclude that demand is likely to be best forecast by means of a neural network. Using the logarithmic dataset produces more consistent results, but is on average very close to the regular dataset. Additionally, it seems better to forego pre-selecting variables, and letting our variable-selection algorithm run its course. Future research into the forecasting of demand for aid goods could start by taking these conclusions into consideration, allowing it to focus on other parameters within the neural networks themselves.

Having selected a best model type, we wanted to know how these forecasts can be practically applied to the field of humanitarian logistics. We created a simplified version of the choices humanitarian organizations face when shipping goods from a supply port to their destination. Using four scenarios with different demand structures and different lead times and costs based on an actual response to a humanitarian crisis, we concluded that forecasting too much by a percent caused slightly less than a one percent increase in costs, whereas forecasting a percent too little increased the costs by more than one percent. From this we see that even though forecasting too much is better than too little, a perfect forecast is always preferred, as is to be expected.

Using these scenarios, we put our final model to the test and compared it to the forecasts made by UNOCHA for a real-life large earthquake. Here, our model forecast 143 % of the actual demand and the UNOCHA forecast only 40 %. In both scenarios and for both demand structures our forecast resulted in lower costs, which shows that the methods presented in this thesis are likely to lead to cost reduction when applied to real-life scenarios. The impact of machine learning is already noticeable for this single disaster, and further development of these applications will only improve their effectiveness.

In conclusion, this thesis has examined the application of machine learning techniques in humanitarian aid forecasts, in an effort to bridge the gap between commercial and humanitarian logistics, and to break the vicious circle of data neglect in the field. This research applied machine learning in reliably

filling up empty datasets, and in creating the forecast models themselves and providing their variables and parameters. All the employed techniques outperformed their traditional counterparts, and the final model showcased its efficacy when applied to a real-life disaster and its logistic problems. Machine learning techniques are proven to be highly effective, and should help increase awareness of the importance of data science in the humanitarian world. With continued innovations in the field, aid will be delivered better, quicker and cheaper to those who need it most.

Recommendations

This research has introduced multiple applications of machine learning algorithms into the world of humanitarian logistics in an attempt to profit from discoveries made in the commercial sector. However, there is a wealth of subjects in these applications that could be topic of future research, as has been alluded to throughout this thesis. Further research into these subjects should help in achieving a larger foothold for machine learning in humanitarian logistics, for which we shall give direction here.

An important given in the world of humanitarian logistics is its lack of data proficiency which has been perpetuated by a lack of successful applications of that data. As a result, this thesis built a dataset from many different sources that did not always match, where sometimes the number of affected people had to be derived from the number of damaged buildings rather than from a direct statistic. This led us to remove several observations where different sources provided different information as we suspected the derived number of affected people to be incorrect. Information is likely to have differed between data sources as multiple sources were built up using estimates. This implies that other observations could also suffer from these inconsistencies, making the dataset overall less reliable. Should a single dataset on the earthquakes with verified values of the dependent variable become available, we then could confidently use all observations with greater accuracy than before, improving all results found in this thesis.

Another point related to data is that the models were created using countrywide data. This data will not be representational in the case of a disaster near a border, or in the case of a larger country with regional differences in the values of these indicators. Regional datasets are therefore more likely to produce observations that are closer to the truth than countrywide datasets. Making regional datasets available for so many different indicators, however, is a long and difficult task that might bring further inaccuracies into the dataset. A similar procedure as was used to fill in missing country data using spectral clustering could be employed to make this ambition more realistic.

The scope of the subject of machine learning is constantly growing and as a result far surpasses the methods used in this thesis. For an initial look into the application of machine learning techniques in humanitarian aid forecasts, the methods used here will suffice. In future research, however, many more machine learning forecast models could be researched. In particular, models that forecast demand over time would be of great interest when combined with the impact measurement as described in this thesis, rather than using a constant periodical demand as was employed here. As long as data on time-varying demand does not exist, however, these methods cannot be applied effectively. The availability of data with day-by-day demand for instance would unlock new research methods that can be applied to an even broader range of forecast models.

In the methods employed during this thesis, we encountered several parameters that in some way influence the training process for our models. Some of these we elaborated on and were tested by means of cross-validation, but for others we simply assumed their value. Further research could investigate this more thoroughly, optimizing not only the hidden layer composition but also other parameters such as neural network threshold and learning rate.

On the topic of cross-validation, we only tried a single division of the dataset into training set and testing set, and a single division of the training set into 5 folds. These divisions were based on the chronology of the dataset, which resulted in a testing set with a much lower variance than that of the training set. As a result, the naive model using the historical mean performed much better than would be expected for the rest of the dataset, and our models performed much worse. In addition, multiple divisions into folds during the variable and parameter selection could have produced different models. This is a modelling choice that could have been examined in this or future research.

An important feature in this thesis was showing how forecasts could impact real-life decision making and their associated costs. Two scenarios were developed based on data of a humanitarian response, using data on shipment size, costs and lead time for both ocean and air transport for multiple ports of origin. The number of people that could be aided with a certain shipment, along with the costs of acquiring those shipments, was based on previous research. These numbers should be revised using data on multiple humanitarian operations rather than the single operation used in this thesis before further research into impact measurement can be performed. Only then can we attach more credibility to this method of impact measurement.

In addition, our impact measurement is based on a simplified version of an actual humanitarian logistic operation. For instance, future research into impact measurement could introduce smaller time windows within which choices have to be made, multiple supply ports for humanitarian aid goods, and some method of carrying over surplus supply to future time periods. These adaptations to the logistic problem that is solved in order to measure impact should be made in accordance with humanitarian aid organizations if we want them to represent real-life as closely as possible.

With the directions for further research given here, application of machine learning in humanitarian aid forecasts and their effectiveness seems like fertile ground for future research. Hopefully, these recommendations will further help bridge the gap between commercial and humanitarian logistics, using years of technological advancements to better help those in need.

References

Behl, Abhishek and Pankaj Dutta (2019). "Humanitarian supply chain management: a thematic literature review and future directions of research". In: *Annals of Operations Research* 283.1, pp. 1001–1044.

Carbonneau, Real, Kevin Laframboise, and Rustam Vahidov (2008). "Application of machine learning techniques for supply chain demand forecasting". In: *European Journal of Operational Research* 184.3, pp. 1140–1154.

Debnath, Kumar Biswajit and Monjur Mourshed (2018). "Forecasting methods in energy planning models". In: *Renewable and Sustainable Energy Reviews* 88, pp. 297–325.

Jahre, Marianne et al. (2011). "Predicting the unpredictable–demand forecasting in international humanitarian response". In: *Proceedings of the 23rd Annual NOFOMA Conference. Harstad, Norway*, pp. 265–281.

Laan, Erwin van der et al. (2016). "Demand forecasting and order planning for humanitarian logistics: An empirical assessment". In: *Journal of Operations Management* 45, pp. 114–122.

Li, Yanting, Yan Su, and Lianjie Shu (2014). "An ARMAX model for forecasting the power output of a grid connected photovoltaic system". In: *Renewable Energy* 66, pp. 78–89.

Ofli, Ferda et al. (2016). "Combining human computing and machine learning to make sense of big (aerial) data for disaster response". In: *Big data* 4.1, pp. 47–59.

Shi, Heng, Minghao Xu, and Ran Li (2017). "Deep learning for household load forecasting—A novel pooling deep RNN". In: *IEEE Transactions on Smart Grid* 9.5, pp. 5271–5280.

Van Wassenhove, Luk N (2006). "Humanitarian aid logistics: supply chain management in high gear". In: *Journal of the Operational research Society* 57.5, pp. 475–489.

Von Luxburg, Ulrike (2007). "A tutorial on spectral clustering". In: *Statistics and computing* 17.4, pp. 395–416.

Wood, Harry O and Frank Neumann (1931). "Modified Mercalli intensity scale of 1931". In: *Bulletin of the Seismological Society of America* 21.4, pp. 277–283.

Yildiz, Baran, Jose I Bilbao, and Alistair B Sproul (2017). "A review and analysis of regression and machine learning models on commercial building electricity load forecasting". In: *Renewable and Sustainable Energy Reviews* 73, pp. 1104–1122.

Yu, Manzhu, Chaowei Yang, and Yun Li (2018). "Big data in natural disaster management: a review". In: *Geosciences* 8.5, p. 165.

Zhang, Fuqing and Yonghui Weng (2015). "Predicting hurricane intensity and associated hazards: A five-year real-time forecast experiment with assimilation of airborne Doppler radar observations". In: *Bulletin of the American Meteorological Society* 96.1, pp. 25–33.

Data References

National Geophysical Data Center / World Data Service (NGDC/WDS): NCEI/WDS Global Significant Earthquake Database. NOAA National Centers for Environmental Information. doi:10.7289/V5TD9V7K
Accessed 14-08-2020

Center for International Earth Science Information Network - CIESIN - Columbia University. 2018. Gridded Population of the World, Version 4 (GPWv4): Population Count, Revision 11. Palisades, NY: NASA Socioeconomic Data and Applications Center (SEDAC). https://doi.org/10.7927/H4JW8BX5.
Accessed 28-07-2020

The World Bank: World Development Indicators: Multiple sources.
https://databank.worldbank.org/source/world-development-indicators/#
Accessed 18-08-2020

United Nations, Department of Economic and Social Affairs, Population Division (2019). Database on Household Size and Composition 2019. Copyright @ 2019 by United Nations, made available under a Creative Commons license (CC BY 3.0 IGO) http://creativecommons.org/licenses/by/3.0/igo
https://www.un.org/development/desa/pd/data/household-size-and-composition
Accessed 09-09-2020

Global Data Lab - Area Database version 3.7.0
https://globaldatalab.org/areadata/hhsize/
Accessed 09-09-2020

United Nations Office for the Coordination of Humanitarian Affairs (UNOCHA) - Flash Appeal Haiti Earthquake 2010
Version 1.1 16 January 2010 - humanitarianresponse.info/en/programme-cycle/space/document/flash-appeal-haiti-2010
Revised 18 Februari 2010 - humanitarianresponse.info/en/programme-cycle/space/document/revision-flash-appeal-haiti-2010
Accessed 03-12-2020

Appendices

A Variables

There are 50 variables taken from the WDI, presented below:

No.	Variable description
1	Access to clean fuels and technologies for cooking (% of population)
2	Access to electricity (% of population)
3	Adults (ages 15+) and children (ages 0-14) newly infected with HIV
4	Birth rate, crude (per 1,000 people)
5	Cause of death, by communicable diseases and maternal, prenatal and nutrition conditions (% of total)
6	Current health expenditure (% of GDP)
7	Death rate, crude (per 1,000 people)
8	Diabetes prevalence (% of population ages 20 to 79)
9	Domestic private health expenditure (% of current health expenditure)
10	Employment to population ratio, 15+, total (%) (modeled ILO estimate)
11	External health expenditure (% of current health expenditure)
12	GDP per capita (current US$)
13	GDP per capita, PPP (current international $)
14	Gini index (World Bank estimate)
15	Hospital beds (per 1,000 people)
16	Immunization, DPT (% of children ages 12-23 months)
17	Immunization, HepB3 (% of one-year-old children)
18	Immunization, measles (% of children ages 12-23 months)
19	Incidence of HIV (per 1,000 uninfected population ages 15-49)
20	Incidence of malaria (per 1,000 population at risk)
21	Incidence of tuberculosis (per 100,000 people)
22	Intentional homicides (per 100,000 people)
23	Labor force participation rate, total (% of total population ages 15-64) (modeled ILO estimate)
24	Life expectancy at birth, total (years)
25	Maternal mortality ratio (modeled estimate, per 100,000 live births)
26	Mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 (%)
27	Mortality rate attributed to unsafe water, unsafe sanitation and lack of hygiene (per 100,000 population)
28	Mortality rate, infant (per 1,000 live births)
29	Newborns protected against tetanus (%)
30	Number of infant deaths
31	Nurses and midwives (per 1,000 people)
32	People practicing open defecation (% of population)
33	People using at least basic drinking water services (% of population)
34	People using at least basic sanitation services (% of population)
35	Physicians (per 1,000 people)
36	Population ages 0-14 (% of total population)
37	Population ages 15-64 (% of total population)
38	Population density (people per sq. km of land area)
39	Population in urban agglomerations of more than 1 million (% of total population)
40	Population, total
41	Poverty gap at $1.90 a day (2011 PPP) (%)
42	Prevalence of anemia among children (% of children under 5)
43	Prevalence of anemia among pregnant women (%)
44	Prevalence of HIV, total (% of population ages 15-49)
45	Prevalence of undernourishment (% of population)
46	Rural population (% of total population)
47	Share of youth not in education, employment or training, total (% of youth population)
48	Smoking prevalence, total (ages 15+)
49	Total reserves (includes gold, current US$)
50	Unemployment, total (% of total labor force) (modeled ILO estimate)

Table 15: Variables related to local indicator data

No.	Variable description
51	Magnitude of earthquake (on scale of Richter)
52	Estimated population within 25 km
53	Estimated population within 100 km
54	Estimated population within 250 km
55	Focal depth of earthquake (in kilometers)

Table 16: Variables related to disaster and population data

B Country Clusterings

For the spectral clustering, we used variables 2, 13, 24, 28 and 33 as introduced in Table 15. The results of the clustering are given below:

Cluster	Countries in cluster
1	Austria, Bahrain, Bermuda, Brunei Darussalam, Canada, Cayman Islands, Denmark, Germany, Hong Kong SAR (China), Iceland, Ireland, Kuwait, Luxembourg, Macao SAR (China), Netherlands, Norway, Qatar, San Marino, Saudi Arabia, Singapore, Sweden, Switzerland, United Arab Emirates, United States
2	Armenia, Georgia, Indonesia, Jamaica, Jordan, Mongolia, Namibia, Nauru, Sri Lanka, Ukraine
3	Angola, Belize, Bhutan, El Salvador, Eswatini, Guatemala, Guyana, Moldova, Morocco
4	Congo Rep., Cote d'Ivoire, Ghana, Honduras, India, Kyrgyz Republic, Lao PDR, Marshall Islands, Mauritania, Micronesia Fed. Sts., Myanmar, Nicaragua, Nigeria, Pakistan, Papua New Guinea, Sao Tome and Principe, Sudan, Tuvalu, Vietnam, West Bank and Gaza, Yemen Rep.
5	Burkina Faso, Burundi, Central African Republic, Chad, Congo Dem. Rep., Guinea-Bissau, Madagascar, Rwanda, Sierra Leone, Togo
6	Botswana, Brazil, Costa Rica, Lebanon, Maldives, Montenegro, Serbia, St. Lucia, Suriname, Thailand
7	Australia, Belgium, Finland, France, Guam, Italy, Japan, Oman, United Kingdom
8	Gambia, Lesotho, Nepal, Tanzania, Timor-Leste, Zimbabwe
9	Afghanistan, Eritrea, Haiti
10	Ethiopia, Liberia, Malawi, Mozambique, Niger
11	Antigua and Barbuda, Croatia, Estonia, Hungary, Latvia, Libya, Lithuania, Malaysia, Poland, Seychelles, Slovak Republic, St. Kitts and Nevis
12	Argentina, Bahamas, Chile, Curacao, Cyprus, Czech Republic, Equatorial Guinea, Greece, Israel, Kazakhstan, Korea Rep., Malta, New Zealand, Panama, Portugal, Puerto Rico, Russian Federation, Slovenia, Spain, Trinidad and Tobago, Turkey
13	Barbados, Belarus, Bulgaria, Gabon, Iran Islamic Rep., Mauritius, Mexico, Palau, Romania, Uruguay, Venezuela
14	Guinea, Kiribati, Mali, Solomon Islands, Uganda
15	Benin, Comoros, Cuba, Korea Dem. People's Rep., Somalia, South Sudan, Syrian Arab Republic, Tajikistan
16	Algeria, Azerbaijan, Colombia, Dominican Republic, Grenada, Iraq, North Macedonia, South Africa
17	Aruba, Bangladesh, Bolivia, Cabo Verde, Cambodia, Cameroon, Djibouti, Kenya, Philippines, Samoa, Senegal, Sint Maarten, Tonga, Uzbekistan, Vanuatu, Zambia
18	Albania, Bosnia and Herzegovina, China, Dominica, Ecuador, Egypt Arab Rep., Fiji, Paraguay, Peru, St. Vincent and the Grenadines, Tunisia, Turkmenistan

Table 17: Country clusters as a result of spectral clustering

The clustering based on geographical location of countries as employed by the World Bank is presented below:

Cluster name	Countries in cluster
East Asia & Pacific	Australia, Brunei Darussalam, Cambodia, China, Fiji, Guam, Hong Kong SAR (China), Indonesia, Japan, Kiribati, Korea Dem. People's Rep., Korea Rep., Lao PDR, Macao SAR (China), Malaysia, Marshall Islands, Micronesia Fed. Sts., Mongolia, Myanmar, Nauru, New Zealand, Palau, Papua New Guinea, Philippines, Samoa, Singapore, Solomon Islands, Thailand, Timor-Leste, Tonga, Tuvalu, Vanautu, Vietnam
Europe & Central Asia	Albania, Andorra, Armenia, Austria, Azerbaijan, Belarus, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Georgia, Germany, Greece, Hungary, Iceland, Ireland, Italy, Kazakhstan, Kyrgyz Republic, Latvia, Liechtenstein, Lithuania, Luxembourg, Moldova, Montenegro, Netherland, North Macedonia, Norway, Poland, Portugal, Romania, Russian Federation, San Marino, Serbia, Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Tajikistan, Turkey, Turkmenistan, Ukraine, United Kingdom, Uzbekistan
Latin America	Antigua and Barbuda, Argentina, Aruba, Bahamas, Barbados, Belize, Bolivia, Brazil, Cayman Islands, Chile, Colombia, Costa Rica, Cuba, Curacao, Dominica, Dominican Republic, Ecuador, El Salvador, Grenada, Guatemala, Guyana, Haiti, Honduras, Jamaica, Mexico, Nicaragua, Panama, Paraguay, Peru, Puerto Rico, Sint Maarten, St. Kitts and Nevis, St. Lucia, St. Vincent and the Grenadines, Suriname, Trinidad and Tobago, Uruguay, Venezuela
Middle East & North Africa	Algeria, Bahrain, Djibouti, Egypt Arab Rep., Iran Islamic Rep., Iraq, Israel, Jordan, Kuwait, Lebanon, Libya, Malta, Morocco, Oman, Qatar, Saudi Arabia, Syrian Arab Rep., Tunisia, United Arab Emirates, West Bank and Gaza, Yemen, Rep.
North America	Bermuda, Canada, United States
South Asia	Afghanistan, Bangladesh, Bhutan, India, Maldives, Nepal ,Pakistan, Sri Lanka
Sub-Saharan Africa	Angola, Benin, Botswana, Burkina Faso, Burundi, Cabo Verde, Cameroon, Central African Republic, Chad, Comoros, Congo Dem. Rep., Congo Rep. Cote d'Ivoire, Equatorial Guinea, Eritra, Eswatini, Ethiopia, Gabon, Gambia, Ghana, Guinea, Guinea-Bissau, Kenya, Lesotho, Liberia, Madagascar, Malawi, Mali, Mauritania, Mauritius, Mozambique, Namibia, Niger, Nigeria, Rwanda, Sao Tome and Principe, Senegal, Seychelles, Sierra Leone, Somalia, South Africa, South Sudan, Sudan, Tanzania, Togo, Uganda, Zambia, Zimbabwe

Table 18: Country clusters as a result of geographical clustering

C Model Specifications

We provide the specific choices made in each model in Table 19 below.

Model ID	Method	Dependent variable	Performance measure	Preselected variables
1	Linear	Regular	RMSE	Yes
2	Linear	Regular	RMSE	No
3	Linear	Regular	MAE	Yes
4	Linear	Regular	MAE	No
5	Linear	Logarithmic	RMSE	Yes
6	Linear	Logarithmic	RMSE	No
7	Linear	Logarithmic	MAE	Yes
8	Linear	Logarithmic	MAE	No
9	Neural	Regular	RMSE	Yes
10	Neural	Regular	RMSE	No
11	Neural	Regular	MAE	Yes
12	Neural	Regular	MAE	No
13	Neural	Logarithmic	RMSE	Yes
14	Neural	Logarithmic	RMSE	No
15	Neural	Logarithmic	MAE	Yes
16	Neural	Logarithmic	MAE	No

Table 19: Specifications of all models

D Variable Selection Results

Linear model, regular dependent variable

These tables give the composition of the variable sets shown in Tables 4 and 5

Model ID	No. of variables	Variables selected
1	8	51, 52, 53, 54, 55, 11, 50, 44
2	1	3

Table 20: Composition of the variable sets shown in Table 4

Model ID	No. of variables	Variables selected
3	6	51, 52, 53, 54, 55, 44
4	4	21, 7, 40, 22

Table 21: Composition of the variable sets shown in Table 5

Linear model, logarithmic dependent variable

These tables give the composition of the variable sets shown in Tables 6 and 7

Model ID	No. of variables	Variables selected
5	37	51, 52, 53, 54, 55, 22, 25, 26, 30, 8, 21, 33, 47, 27, 11, 48, 17, 36, 34, 42, 15, 24, 29, 44, 45, 41, 20, 6, 37, 7, 13, 12, 14, 28, 23, 10, 50
6	45	10, 31, 6, 23, 51, 38, 55, 11, 43, 21, 13, 39, 15, 50, 4, 37, 40, 32, 30, 42, 27, 25, 49, 48, 17, 24, 19, 33, 7, 47, 34, 22, 28, 35, 12, 8, 46, 5, 44, 36, 20, 45, 14, 41, 3

Table 22: Composition of the variable sets shown in Table 6

Model ID	No. of variables	Variables selected
7	31	51, 52, 53, 54, 55, 22, 25, 18, 35, 26, 47, 29, 33, 41, 11, 5, 20, 48, 9, 28, 4, 42, 8, 36, 13, 45, 15, 27, 14, 6, 7
8	22	10, 23, 52, 28, 20, 55, 43, 37, 34, 50, 8, 16, 30, 12, 40, 26, 27, 33, 35, 32, 44, 39

Table 23: Composition of the variable sets shown in Table 7

Neural network, regular dependent variable

These tables give the composition of the variable sets shown in Tables 8 and 9.

Model ID	No. of variables	Variables selected
9	20	51, 52, 53, 54, 55, 11, 18, 17, 40, 30, 44, 4, 1, 16, 31, 29, 37, 20, 46, 12
10	17	3, 14, 40, 20, 52, 28, 34, 17, 38, 11, 10, 2, 51, 32, 13, 31, 53

Table 24: Composition of the variable sets shown in Table 8

Model ID	No. of variables	Variables selected
11	20	51, 52, 53, 54, 55, 9, 15, 36, 7, 19, 40, 23, 17, 14, 29, 16, 21, 27, 34, 32
12	16	21, 37, 33, 30, 34, 52, 36, 51, 25, 39, 22, 42, 2, 17, 1, 24

Table 25: Composition of the variable sets shown in Table 9

Neural network, logarithmic dependent variable

These tables give the composition of the variable sets shown in Tables 10 and 11.

Model ID	No. of variables	Variables selected
13	10	51, 52, 53, 54, 55, 36, 30, 40, 28, 14
14	4	10, 54, 42, 32

Table 26: Composition of the variable sets shown in Table 10

Model ID	No. of variables	Variables selected
15	5	51, 52, 53, 54, 55
16	8	10, 54, 44, 9, 45, 20, 34, 29

Table 27: Composition of the variable sets shown in Table 11