Priority-Based Humanitarian Aid Modelling for Flood Impact in Malawi

Thomas Plaatsman

June 20, 2017

Abstract

Malawi suffers frequent floods after which Non Governmental Organizations, such as the Red Cross, provide humanitarian response. In order to distribute this help according to the needs, it is important that emergency response teams have the right information about the impact, the affected area and the affected population. We propose four different Machine Learning Techniques to predict the aid-neediness of a Traditional Authority in Malawi and identify the most important factors influencing this prediction. The four techniques used in this research are the CART Decision Tree, Conditional Inference Tree, Random Forest and Conditional Forest. These models are used to predict if a Traditional Authority is a high-, moderate- or Low aid-neediness area for the flood that took place in Malawi during the month of January 2015. A data set with 79 observations and 22 pre-selected explanatory variables is used, containing variables related to hazard and exposure, vulnerability and the lack of coping capacity. The models identify as most important factors, the percentage of a Traditional Authority that has been flooded and the percentage of drinking water that comes from natural sources. A repeated 5-fold cross validation, with 100 repeats is done to compare the Accuracy of the models and results into an Accuracy between 52% and 64% for the different models, with the Random Forest being the best performing one. We recommend increased use of Machine Learning techniques for disaster response in humanitarian aid. However, before being able to successfully implement these results in practice, further research is necessary.

Keywords: machine learning, decision trees, Random Forest, humanitarian aid modelling

Contents

Acronyms

CART Classification And Regression Tree

Ctree Conditional Inference Tree

DEM Digital Elevation Model

FACT Field Assessment Coordination Team

GIS Geographical Information System

HAND Height Above the Nearest Drainage

HDX Humanitarian Data Exchange

IASC Inter-Agency Standing Committee

IFRC Internation Federation of Red Cross and Red Crescent Societies

IHS Integrated Household Survey

INFORM Index for Risk Management

MASDAP Malawi Spatial Data Portal

NGO Non Govermental Organization

NSO National Statistics Office

OOB Out-Of-Bag Error

OSM Open Street Maps

TA Traditional Authority

TRMM Tropical Rainfall Measuring Mission

UN OCHA United Nations Office for the Coordination of Humanitarian Affairs

1 Introduction

Between 1994 and 2013, almost 7000 natural disasters took place all over the world, with floods and storms as the two most frequently occurring types of natural disasters. Earthquakes have been the type of disaster responsible for the largest part of deaths and overall, natural disasters are responsible for approximately 1.35 million deaths over the last two decades (UNISDR, 2015).

Thousands of humanitarian aid organizations worldwide commit themselves providing the best possible aid in the aftermath of a natural disaster. During this aftermath, it is of great importance to identify the affected areas quickly and correctly, to provide immediate and effective assistance to the many people in urgent need for help. In order to achieve this goal, international organizations send information specialists to the country to conduct "needs assessments" (Benini & Chataigner, 2014). This information is then combined with that of local institutions and Non-Governmental Organizations (NGOs) present in the area. In 1992 the Inter-Agency Standing Committee (IASC) was founded, with as overall goal to improve the delivery of humanitarian assistance to affected populations. As a part of the IASC, a Multi-Cluster/Sector Initial Rapid Assessment is performed. This assessment focuses on the immediate aftermath of a disaster and performs a Preliminary Scenario Definition within the first 72 hours after the disaster took place, providing information about the number of affected people. This assessment is followed by a more detailed report, two weeks after the disaster. This second report aims to help coordinate all the parties involved.

Although multiple frameworks to improve coordination in humanitarian aid have been established through the years, uncertainty about how to distribute help effectively remains a big concern for humanitarian aid teams delivering the help on the ground and is the main reason for the need for formal assessments. Furthermore, due to limited time and the need for fast decision making, information is not always shared with other parties and tends to be incomplete. Moreover, decisions are often made using rules of thumb and lack an underlying statistical framework. A Humanitarian Policy Report in 2003 already addressed these problems and mentioned three main concerns; financing is not equitable and does not reflect the levels of help needed, there is no system-wide framework and donors are sceptical about the objectiveness of assessments (Darcy & Hofmann, 2003). Furthermore, a study by Marc van den Homberg (2014) concludes that the political context is a very important factor and that this may lead to factors, other than the need on the ground, influencing decision making. This may result into certain areas not being taken into account and therefore not receiving the help they need.

This study aims for a better, faster and more reliable humanitarian relief using open source data and predictive analytics in order to overcome information gaps and give help solely based on the needs. These should be the first steps towards a more open and reliable way of data collection, sharing and interpreting. The aim of this study is to be able to create a forecast within the first

24-48 hours after a natural disaster took place or international help is requested. The goal is to give a quick and reliable overview of the affected areas, using available data and analysis by machine learning algorithms.

The scope of this paper is the Malawi flooding disaster in 2015. The reason to choose this scope is that the collection of data, especially considering the different types and sources of data, is time consuming. In January 2015, heavy rainfall caused floods in southern Africa, including Madagascar, Malawi, Mozambique and Zimbabwe. The Southern Region of Malawi received 400% more rainfall than usual, creating floods that affected 15 of Malawi's 28 districts and up to 638000 people (*Reliefweb, Southern Africa: Floods Jan 2015*, n.d.). This event is used to explore how data can be used more effectively to provide faster and more reliable disaster relief for floods in Malawi. This leads to the main research question, which is:

Can Machine Learning techniques be used to better allocate help in case of floods in Malawi?

In line with this research question, two sub questions are formulated, which are:

1. **Which Machine Learning Techniques can be used and what are the differences among them?**
2. **What are the most important factors, selected by the algorithms, that indicate the aid-neediness?**

It is important to note that the focus is not on predicting the risk of a new disaster, but the expected help needed in case of a disaster.

2 Literature Review

Natural disasters have been around for ages and a lot of research has been done in this field. Different risk assessment tools exist, such as the Index For Risk Management (INFORM). INFORM is a collaboration of the IASC Task Team for Preparedness and Resilience and the European Commission, with partners such as UN OCHA and Unicef. INFORM is a multiplicative index formulation, creating a risk factor for every country. This risk factor is based on three different components; Hazard & Exposure, Vulnerability and the Lack of Coping Capacity.

Although a risk index like INFORM exists, a number of criticisms have been raised about data use and collection in humanitarian aid. As mentioned in the introduction, Marc van den Homberg (2014) states that decision-making is often political. Darcy and Hofmann (2003) discuss, among other things, the lack of a system-wide framework for judging the relative severity of situations and for aligning

decisions about response accordingly, thus delivering aid accordingly to the aid-neediness. Ravallion (2010) expresses his concerns about the "mash-up" indices. These indices are primarily driven by the availability of the data and the interpretation and robustness are often unclear. He states furthermore that future progress in devising useful new composite indices of development will require that theory catches up with measurement practice. INFORM aims for transparency, robustness and reliability and uses high quality data. However, it is still a multiplicative index and although giving a clear indicator about a country's risk, its rendering of the risk might still be too simple.

It is important to note that natural disaster modelling is a multidisciplinary field and different definitions for concepts such as risk and vulnerability are used. Jonkman (2007) states that definitions in research often define risk in terms of hazard and vulnerability, where hazard refers to the source of danger or the cause and vulnerability relates to potential consequences in case of an event. The main difference between the hazard and risk factors are that risks often includes the probability of an undesired event or natural disaster.

Mwale, Adeloye, and Beevers (2015) mention that vulnerability has traditionally been defined as the susceptibility of communities, elements at risk and the coping capacity, but there has been a shift in frameworks to describe vulnerability also mentioning a distinction between different types of vulnerability, such as social and biophysical vulnerability. Cutter et al. (2008) mention the differences between vulnerability and resilience and state that whereas vulnerability represents the pre-event characteristics and is a function of the exposure and sensitivity, resilience is the ability to respond and cope with disasters, during and after the event of a disaster. In this research, the focus is on predicting the vulnerability in relation to a disaster and not on predicting the occurrence of the hazard itself.

Research published on vulnerability and resilience to disasters involves different types of research and methodology. Among these different types of research are research on policies and strategic decision making, geographical research such as geographical information system (GIS) modelling, hydrological modelling and statistical and machine learning methods. The scope of this research is on increased use of data and modelling in humanitarian aid. Therefore this literature review discusses research done on hydrological, GIS and statistical modelling. In the study of Gao, Nickum, and Pan (2007) on flood prevention in China, they state that a flood with the same intensity can have different effects in losses for different groups of people. Therefore it is important to take into account the potential extent of flooding, the areas at risk and the probable net loss. Furthermore, Gao et al. (2007) distinguish between three categories that generate flood disasters. These categories are disaster environment, such as land surface terrain and soil cover, disaster drivers, such as timing and depth of rainfall and disaster bearers, like the lives of humans and the economy. They state that their work is the first study in China using a vulnerability approach to assess flood prevention that incorporates a more comprehensive set of social and economic factors. A GIS analysis was

used to generate geographical information combined with an analytical hierarchy process approach to determine the weighting factors. The approach seems to be an effective approach of flood control that incorporates a greater complexity of economy, society and environment instead of project-focused approaches.

Tehrany, Pradhan, and Jebur (2013) propose a hydrological approach published in the Journal of Hydrology in combination with statistical methods. They combine a decision tree method and combination of frequency ratio and logistic regression to identify areas susceptible to floods in Malaysia. They then compare the prediction performances and conclude that the traditional hydrological methods should be complemented by Remote Sensing (sensors that collect data by detecting the energy that is reflected from Earth) and GIS.

Two other hydrological modelling approaches are presented by Rudari, Beckers, De Angeli, Rossi, and Trasforini (2016). New, more detailed elevation data leads to an opportunity for more detailed physical based flood hazard models in risk assessment. Rudari et al. (2016) present two different flood hazard models and compare the differences in the effects of modelling scale on hazard and impact losses. Their research is an example of how technological progress leads to new opportunities for the use of modelling to better cope with disasters.

Another approach of natural disaster modelling, with the use of statistical methods, is Moltchanova, Khabarov, Obersteiner, Ehrlich, and Moula (2011). In their research a stochastic modelling approach is applied to assess the value of information for quick earthquake response. They state that earth observations, such as satellite imagery, can lead to more efficient rapid response actions and they regard their work as the first step towards a more systemic understanding of earthquake response actions. Although their research does not involve flood modelling, it does show the opportunities for statistical- and machine learning techniques in the field of disaster modelling and the relatively small amount of research done in this field.

Another approach of disaster modelling that uses statistical methods is that of Kohara and Sugiyama (2013). They focus on a new typhoon warning system, looking at the relation between typhoon data and the damage these typhoons cause to humans and buildings. Kohara uses Self-organizing maps, multiple regression analysis and decision trees for their typhoon damage forecasting, where they consider two-class and three-class damage forecasting.

Wang et al. (2015) use both a Random Forest and support vector machine model with eleven risk indices in the field of regional flood hazard risk to compare the results and the selection of the most important indices. Some of the indices or derivatives of these indices are also used in this research, such as the Digital Elevation Model. The Random Forest techniques are an extension of the decision trees used by Kohara and Sugiyama (2013). The benefits of this technique as stated by Wang et al. (2015) are that it can run efficiently on large databases and provide estimates regarding the importance of specific variables in the classification.

It can be concluded that some research has been done in the fields of GIS, hydrological and statistical modelling. These papers are, as mentioned by Gao et al. (2007) and Moltchanova et al. (2011) often front runners in risk modelling. Thus, our research can be seen as part of these first steps in statistical modelling and the use of open data to improve the humanitarian aid sector and the manner aid is provided. We aim for a more general approach that combines statistical methods with GIS data.

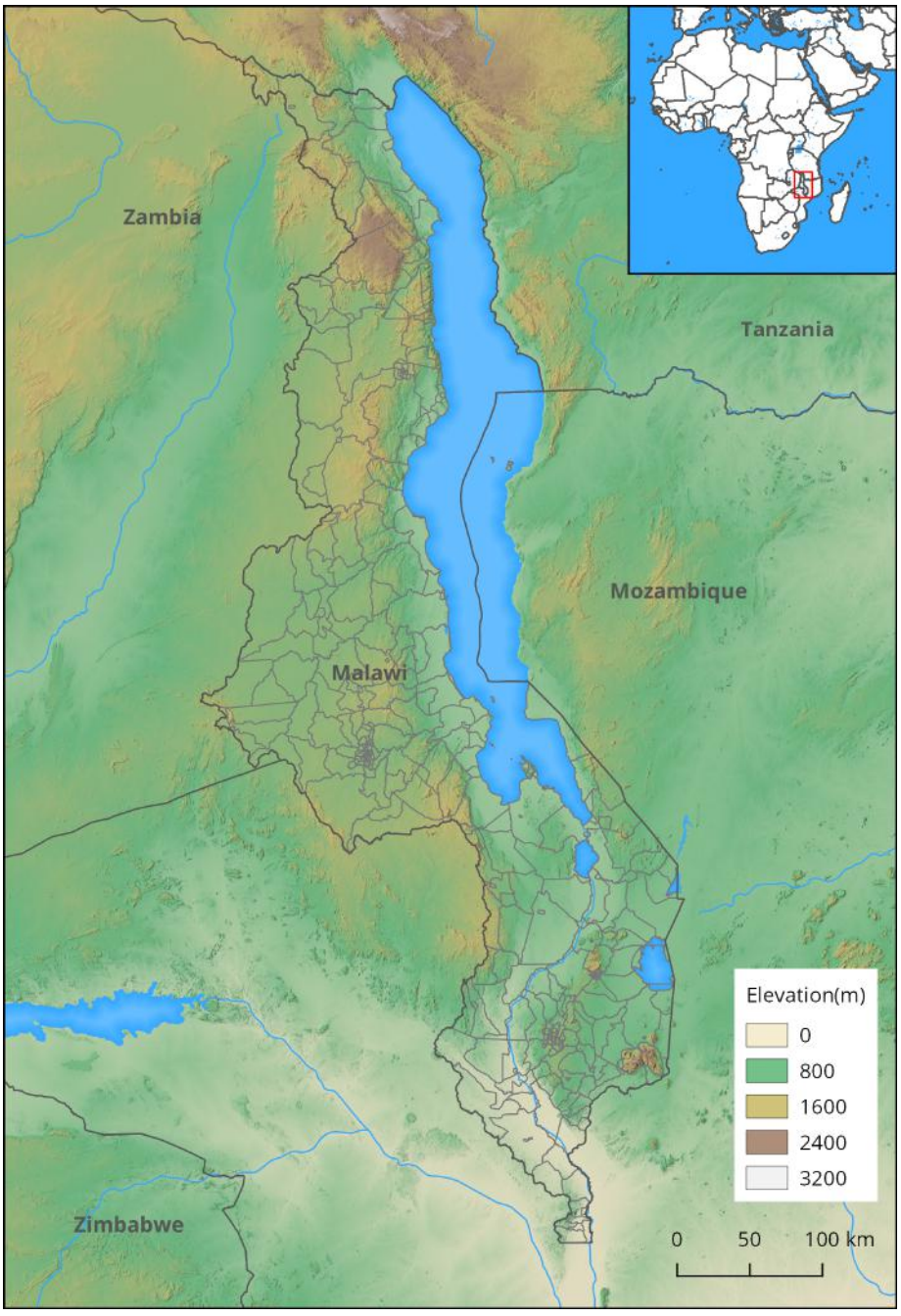

Figure 1: Malawi Overview

3 Data

This research focuses on the use of open data, making this research more transparent and easier to reproduce in other countries. A recent article by Janssen, Charalabidis, and Zuiderwijk (2012) defines open data as non-privacy-restricted and non-confidential data that are produced with public money and are made available without any restrictions on its usage or distribution. This research uses a slightly different definition. The definition for open data is non-privacy-restricted and non-confidential data, but the funding part of the definition is excluded, because it is irrelevant in this research. Furthermore, if data are openly available with some restrictions on use or privacy, they are still considered to be open data in this research. Important sources for the open data used in this research are government agencies, statistical offices, the Humanitarian Data Exchange (HDX) platform, research institutes and institutions such as NASA, providing satellite imagery. Although social media data can also be considered as open data, this is considered to be out of the scope in this research. The reason for the exclusion of certain types of data is that they do not fit into the existing framework used in this research, as described later on in this section, or because the raw data need to be processed with methods other than GIS and machine learning techniques.

The use of open data has a number of benefits, such as; more transparency, improvement of policy making processes, fair decision making by enabling comparison and easier access to data and discovery of data Janssen et al. (2012). For the Red Cross as an organisation, improvement of the policy making and fair decision making are important and in line with its fundamental principles, such as impartiality. Furthermore, transparency to the public and donors is of great importance and open data may contribute to this cause.

Some possible negative side effects of the use of open data do exist. Janssen et al. (2012) mention possible negative side effects: no information on the quality of the data, an unclear trade-off between transparency and privacy and a possible lack of meta standards, such as unclarity about the data source or the meaning of the variables. It is of great importance for the Red Cross to be able to perform its work correctly in the countries it is located in. Combining data and giving more insights into the overall well being of the population of a country may also expose differences and possible inequalities. This could be seen as a threat to governments, because the population might blame government policies for these inequalities. Although it is strongly believed that the benefits outweigh the downsides, these data and analyses should be treated with care and should in no way jeopardise the Red Cross' ability to perform their job.

A big part of this research aims to find the indicators or variables that play an important role in providing better and faster flood relief. Section 3.1 describes the best indicator or proxy indicator to use for the response variable. This response variable describes best the vulnerability to flooding and the amount of help needed after the 2015 flood took place. Section 3.2 describes the indicators that

may affect this vulnerability. A lack of relevant information in the field, especially during the first days after a disaster took place is a recurring problem. The aim is therefore to be able to execute these models within the first few days after a disaster took place and ideally within the first 24-48 hours and thus complement the Preliminary Scenario Definition available within the first 72 hours. The data were collected up to January 13^{th} 2015, with one exception for the flood extent that will be explained in more detail in Section 3.2.

In order to understand this research, it is important to have some understanding of Malawi's administrative regional subdivisions. Malawi consists of 3 regions: South, North and Central. These regions consist of 28 districts and can be subdivided into 350 smaller areas, called Traditional Authorities (TAs). In Figure 1, Malawi's is shown as well as its subdivisions into TAs. The TA level, which is Malawi's third administrative level (regions being 1 and districts 2) is the measurement level of this research. The administrative boundaries used in this research are collected from the Humanitarian Data Exchange Platform and provided by United Nations Coordinator for Humanitarian Aid and Development Activities (UN OCHA). For this research, 79 out of the 350 TAs are considered. These 79 TAs are located within 15 different districts which were all affected by the flood. National parks and nature reserves were excluded in this research, since these are not priority areas for the Red Cross, because of their low number of permanent residents.

3.1 Response Variable

As mentioned in the Introduction, the purpose of this research is forecasting the vulnerability of TAs in Malawi to the flood that took place in 2015. The International Federation of Red Cross and Red Crescent Societies (IFRC) aims to reduce the number of deaths, injuries and impact from disasters and the emphasis is therefore on the number of affected people and not the material damage. Although data on damage is an indicator for the number of people affected, it is likely that big cities have more damage than poorer rural areas, even though this does not reflect the help needed. Therefore, the choice was made to consider different data sources: (1) Help affected people got in the form of receiving tents. (2) The number of people displaced in shelter camps. These data are available from two different data sets on TA level. The first set is the Displacement Tracking Matrix data from the UN International Organization for Immigration, describing the number of individuals that had to be displaced to shelter camps. This gives an indication of the number of houses destroyed or damaged to an extent that they were uninhabitable. The second data set contains information about the UN shelter cluster. The shelter cluster is part of the UN cluster system (as depicted in Figure 2) with as main goal, the improvement of coordination between government and NGOs during emergencies. Every cluster has a designated global lead agency. For the shelter cluster, this lead agency role is co-chaired by the United Nations High Commissioner for Refugees and the International Federation of Red Cross and Red Crescent Societies (IFRC). These data contain information about the number

of people that received help in the form of tents and other materials used to temporarily replace housing. This information functions as a proxy for the amount of houses damaged and the amount of help needed. It is important to note however, that although this indicates the help needed, the results have a bias towards the shelter cluster. For example, some areas with heavily damaged houses, could have a working sanitation system. This would lead to different priorities for the Water, Sanitation and Health cluster. Naturally these are not independent factors and they do influence each other. However, it is necessary to keep these influences in mind, when interpreting the results.

Figure 2: UN OCHA Clusters

Although these data both give an indication for the help needed and could in theory be used as the response variable, some limitations exist. For the shelter data, some of the help might not end up where its needed most, depending the quality of the information NGOs receive, the accessibility of an area and the number of warehouses and NGOs already settled in the area. Therefore, it could occur that certain areas did not receive the exact amount of help they needed. Regarding the displacement data, shelter camps in high poverty areas may have more people than lower poverty areas, because people living in areas with food scarcity are more likely make use of available help, even if their house is not destroyed. These shelter camps could therefore reach their full capacity around dinner time, but only half of it during the day. Due to these issues, the continuous dependent values can not be

interpreted and have to be categorized.

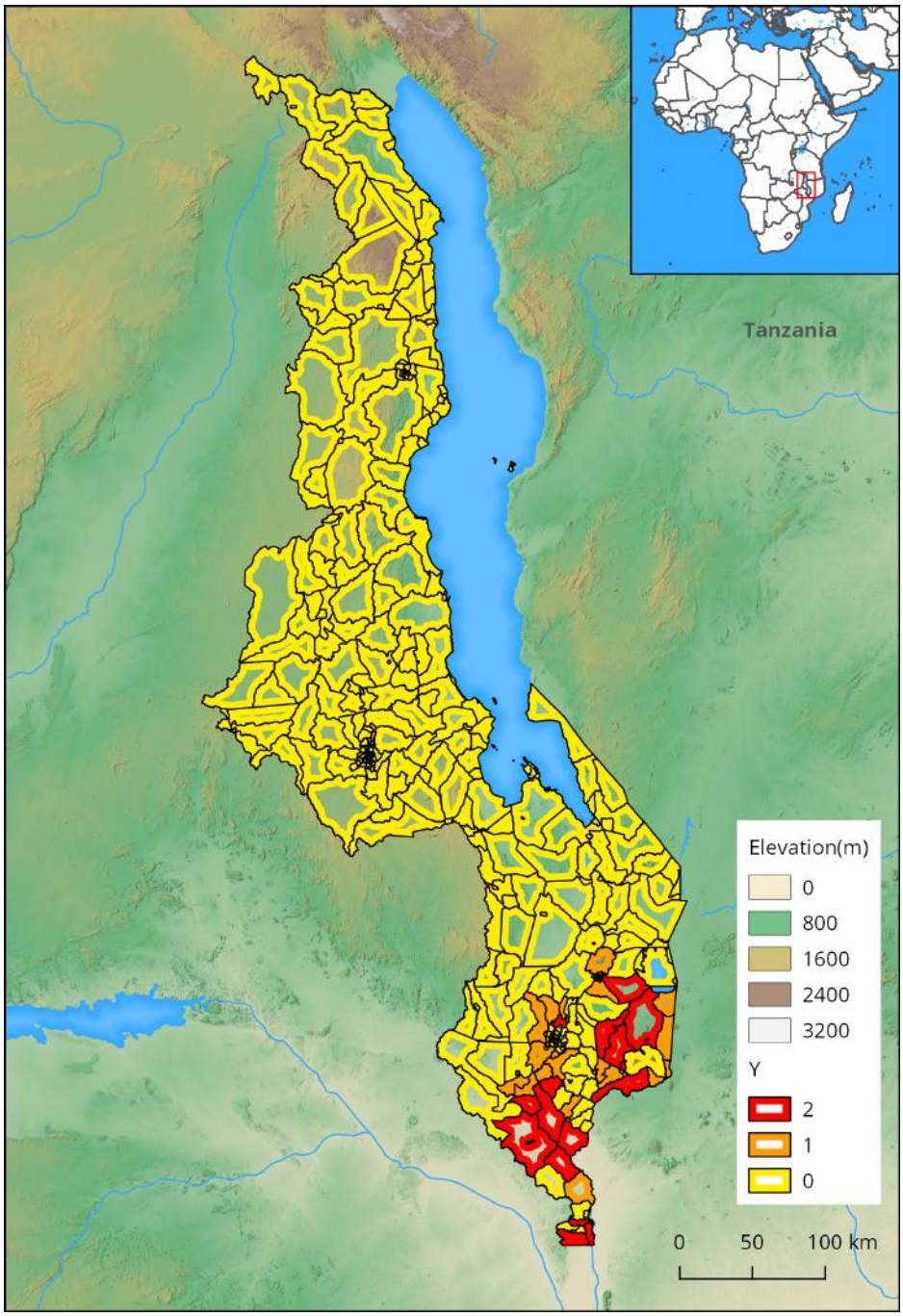

Figure 3: Traditional Authorities aid-neediness Categories

Therefore, the information of the two data sets is combined on TA level and with the help of an expert who worked for the Red Cross Field Assessment Coordination Team during the flood in Malawi, these 79 TAs could be categorized into three categories: Low aid-neediness (0), Moderate

aid-neediness (1) and High aid-neediness (2). The areas are depicted in Figure 3, with 45 TAs in the Low aid-neediness, 18 in the Moderate aid-neediness and 16 in the High aid-neediness category.

The areas as depicted in Figure 3 have been checked and validated by, among others, Malawian Red Cross managers and The Ministry of Water Resources in Malawi during a field visit that took place from the 4^{th} until the 18^{th} of February 2017. One of the Red Cross Managers noted that many people from STA Mbiza moved to TA Mkhumba, which could give a wrong representation of the help needed in general, so this needs to be taken into account when analysing the results. The most important remark, however, was that the districts Chikwawa and Nsanje are always heavily affected by the floods. Therefore, their expectation was that all the lower TAs on the map in Nsanje district, would be marked red. When discussing these areas more thoroughly, they noted that some of these areas, including TA Nsanje Boma have a high elevation and therefore may have been less affected. Figure 4 shows that indeed some of these TAs are on higher ground, explaining why these areas were categorized as lower or Moderate aid-neediness areas. However, not all of these areas are on higher ground and TA Nsanje Boma, which is the smallest yellow TA on the right side in Figure 4, is not one of them. A possible explanation for these areas being lower in terms of aid-neediness could be that its inhabitants fled to relatives living on higher ground in TAs nearby. This example does show us, however, that some of the ideas embedded in people's minds about vulnerable areas are not necessarily correct. Furthermore, showing that the use of data plays a critical role in correctly identifying vulnerable areas.

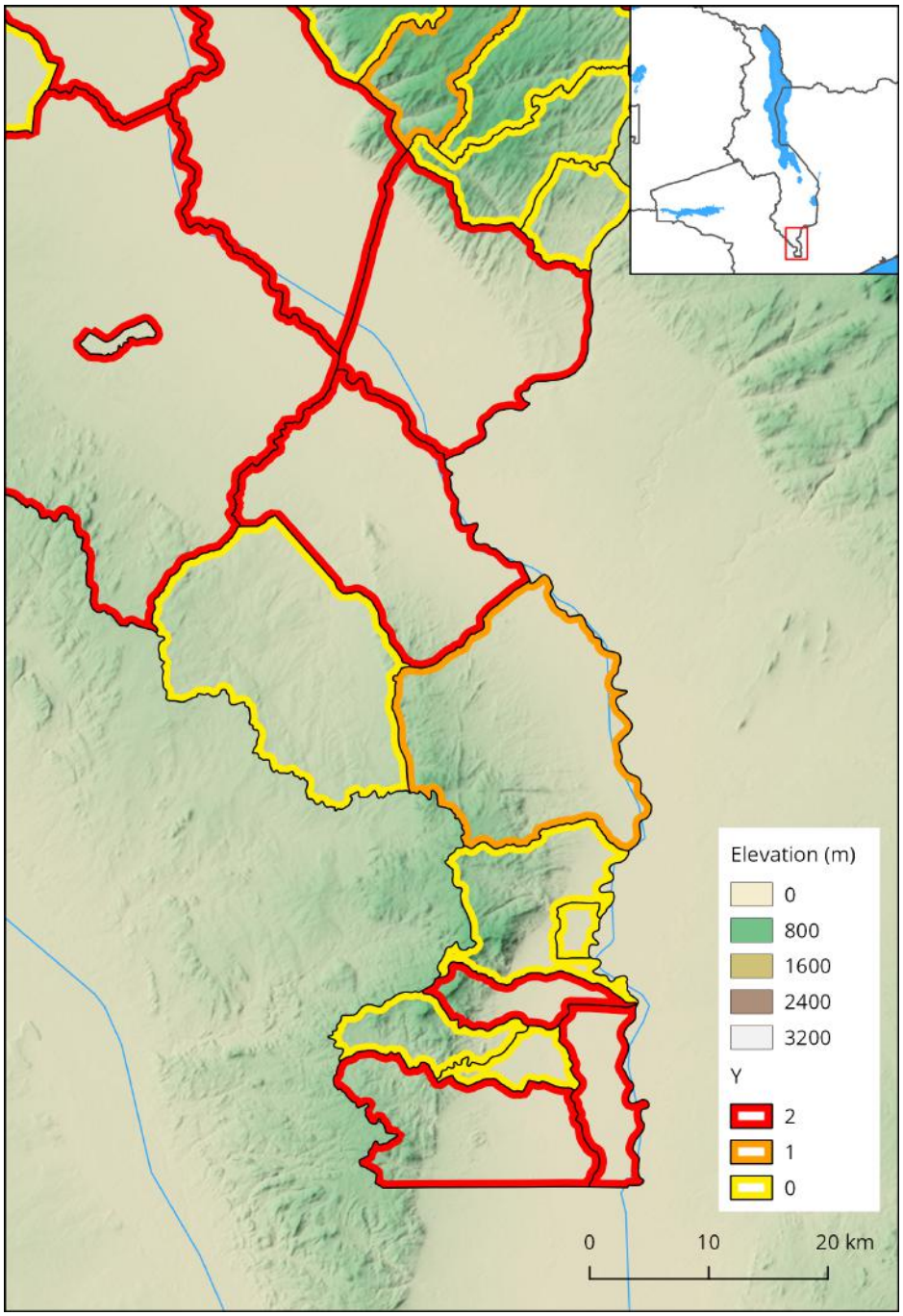

Figure 4: Level of aid-neediness and Elevation of Southern Traditional Authorities

3.2 Explanatory Variables

This section discusses the explanatory variables or indicators that could have an influence on the severity of the impact of a flood. Although this research uses machine learning techniques that are able to process raw data, a rough pre-selection is required. One of the main reasons to pre-select

indicators is that the number of observations is low and the number of possible variables very high. Due to these few observations and a large amount of options, the algorithms will always be able to find variables to split on, but not necessarily the right ones. A first selection of possibly relevant variables is thus made and then the models determine the most important variables out of the pre-selected set of variables. In order to create an appropriate set of variables, this research builds upon the INFORM framework initiated by the European Union and the IASC. The INFORM framework creates a nationwide risk factor for crises and disasters by subdividing the risk factor into different components. In this research, the INFORM index is used as a basis. However, this research only concerns floods and not the other natural disasters that are part of the INFORM index. Furthermore, the focus is on a specific flood event, whereas the INFORM index focuses on a general risk factor. Thus, due to these reasons, we adapted the INFORM index for the purposes of this research.

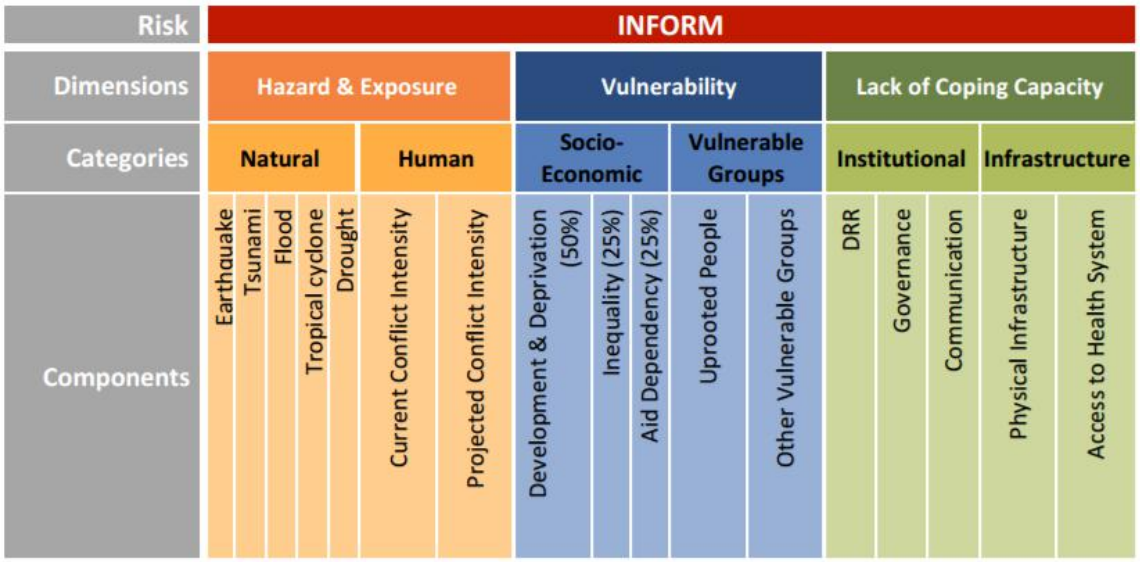

Figure 5: Level of aid-neediness and Elevation of Southern Traditional Authorities

The INFORM index, as shown in Figure 5, is divided into 3 dimensions: Hazard & Exposure, Vulnerability and Lack of Coping Capacity. These main dimensions are used to structure the explanatory variables. However, there are some changes.

In the category Hazard & Exposure, floods and droughts are the two most occurring disasters in Malawi, with a third place for Earthquakes. Since this research focuses on floods, the other hazards are not taken into account. Furthermore, Malawi is not a high conflict area and scores zero on current conflict and only 2.2 out of 10 on conflict risk in the INFORM Index. Because of Malawi scoring very low on conflict, this factor is not taken into account in this research.

When it comes to the Vulnerability category, the data available largely cover socio-economic

vulnerability such as poverty and life expectancy, but also indicate vulnerable groups by considering factors such as child and infant mortality rates.

For the Lack of Coping Capacity, infrastructural variables are considered such as total road length and road density, amount of hospitals, but also mobile phone use, and access and the type of water sanitation. Next these dimensions are described in more detail.

Hazard & Exposure The dimension Hazard & Exposure is divided into two parts in this research: the general hazard and exposure related variables and the event specific ones. In general, flood related geographical variables and variables specific for floods, such as slope and elevation data are mentioned as important indicators in literature, for example by Wang et al. (2015). This research therefore incorporates multiple flood related variables. These variables describe hydrological characteristics of a drainage basin and it is therefore required to have a good understanding of the hydrology of the drainage basin for the areas of interest. A drainage basin is an area of land where all surface water from rain, melting snow, or ice converges to a single point at a lower elevation where the water joins another water body, such as a river, lakes etc.

The general hazard and exposure variables are: the Contributing Area, Stream Order, Height Above the Nearest Drainage (HAND) and Slope. In order to construct these variables, elevation data are needed, which are available online in different resolutions. For this research, 30m resolution data are used, provided by NASAs Shuttle Radar Topography Mission (STRM). Then, to create these variables, the elevation data was analyzed in QGIS (an open source geographic information system). Slope and HAND are calculated at grid level and then averaged over the TAs. Contributing Area and Stream Order were first calculated for each sub basin and stream network, using the zonal statistics tool in QGIS. Then values are assigned for each TA. The Slope is a measurement of steepness and does not need much extra explanation.

The Stream Order is a measure of the relative size of streams. The smallest streams located upstream are referred to as first-order streams. The stream order increases going downstream, hence TAs located further down the river have a higher stream order and thus a bigger river size compared to the TAs located upstream. Figure 6 is a visual representation of a stream order calculation. Figure 7 shows the streams joining at a lower elevation point. This area is the drainage basin.

The Contributing Area is calculated by combining the hydrological flow direction and flow accumulation derived from a Digital Elevation Model (DEM) (Wallis, Watson, Tarboton, & Wallace, 2009) and is defined as the basin area that will supply water runoff to the stream/river during a flood. The main idea behind this concept is that rain that falls upstream affects not only that TA, but also the TAs downstream. These downstream TAs will endure the rainfall in their area and the rainfall from the areas upstream, carried by the river. For this reason, the contributing area in downstream TAs is the sum of the Contributing Areas upstream plus the basin area of the TA itself.

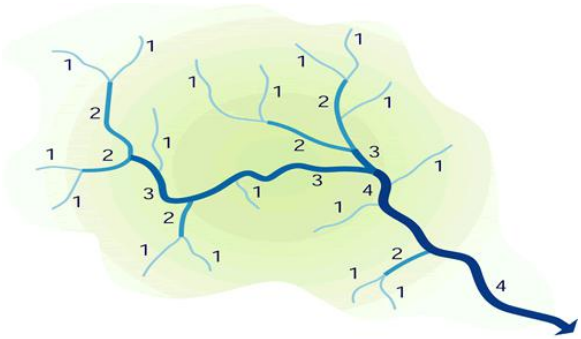

Figure 6: Stream Order

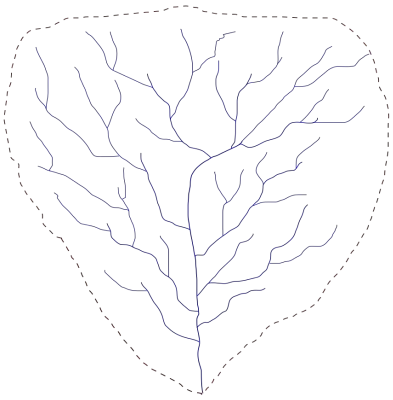

Figure 7: Drainage Basin

The last flood related variable used in this research is the HAND. The HAND is the relative elevation of a point in reference to the closest stream/river and is defined by Rennó et al. (2008). For all the points on the map, the difference in height between that point and the nearest drainage area (or river) is calculated. Then these are averaged per TA. This variable gives a clear overview of the average elevation compared to the height of the rivers.

Event Specific Two variables that are highly event specific are the rainfall and the flood extent, meaning the area that has flooded. These data are only available during or after a flood took place and can not be collected beforehand, like the other variables used in this research.

Rainfall data is available from NASA's Tropical Rainfall Measuring Mission (TRMM). The rainfall indicator for this research is the accumulated rainfall per TA during the 5 days before the call for international help on January 13^{th} and is shown in Figure 8.

The flood maps currently used for this research have been made available by the Netherlands Red Cross and can be found on *nlrc.carto.com*. Ideally, only the use of data up to January 13^{th} is wanted, but with flood extent satellite images this has proven to be a bit more difficult. Satellite images (depending on the type) often have clouds on them, making it impossible to identify all areas that were flooded. Furthermore, because of its orbit around the earth, images are not available every day and may not have been available on January 13^{th} and, depending on the orbit, the satellite imagery may not cover the whole flood area on that date. Therefore, for this research, a combination of four flood extents available on *nlrc.carto.com* is used and retrieved from satellites such as 'Radarsat', 'Radarsat2' and 'Terrasar-X'. These satellite images partly overlap and partly cover different areas. In order to have the full flood extent, the choice is made to select the satellite with the maximum flood extent per TA. The combination of these 4 satellite images and thus the total flood extent is depicted in Figure 9.

Although these images have been retrieved on different dates during the month of January 2015,

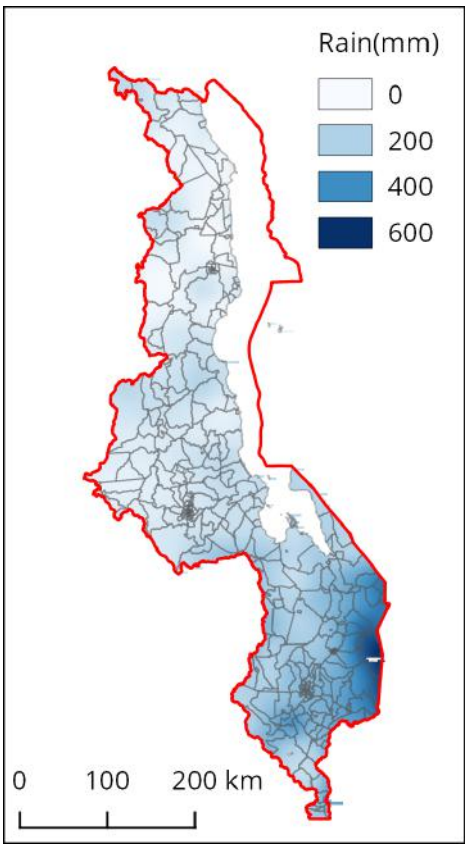

Figure 8: 5Day Cumulative Rainfall Before

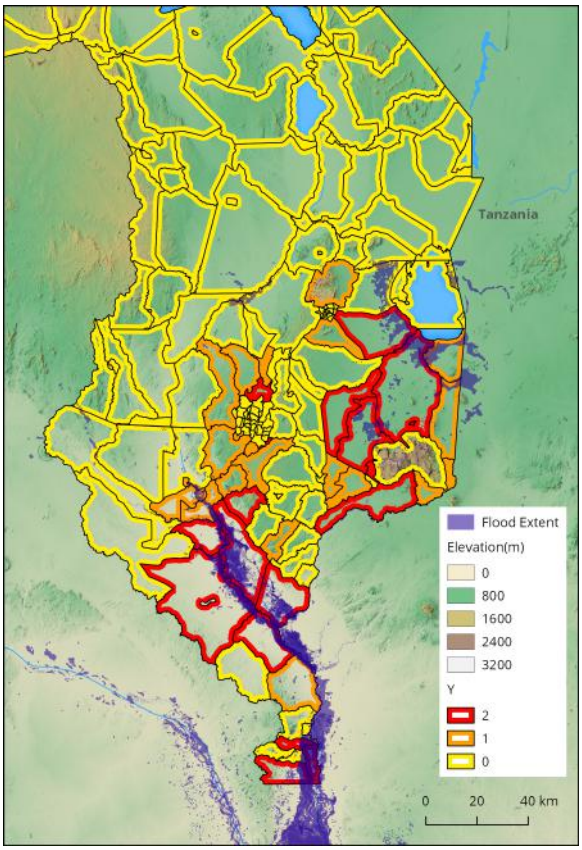

Figure 9: Extended Floodmap January 2015

they give an accurate overview of the flood extent on January 13^{th}. That is because the flood extent reached its maximum around this date and only declined afterwards. However, in order to run the model again for new floods, these data should be readily available within a few days or even hours after a flood took place. This could be done using experts who can draw a flood extent from the satellite images and information they have or by cooperating with research institutes or universities. Another option is downloading raw satellite images (including images of clouds) and using them for the model. For the future usability of this type of modelling, it is important to study the most effective way of retrieving and interpreting satellite imagery on flood extents.

Vulnerability When it comes to vulnerability and socio-economic data especially, one of the main sources providing data is the Malawi Data Portal. This portal is sponsored by the African Development Bank and contains data from the National Statistics Office (NSO) in Malawi. Unfortunately most of these data are not available on TA level, but merely on District level. The variables used from this source are 'Life expectancy at birth', 'Proportion with access to improved sanitation', 'Source of drinking water', 'Child Mortality Rate' and 'Infant mortality rate'. These variables come from different NSO surveys and the dates of these surveys range from the year 2010 up to 2012. It is important to note that some surveys are held only once every 5 or 10 years and these are therefore the most recent, relevant data sources.

More detailed data on TA level are available from a household survey executed by the NSO and sponsored by the Worldbank. Two of these large and detailed household surveys have been published. The first one was done in 2010/2011 and had a sample size of 12,271. The second one in 2013 had a sample size of 4000. Although the 2011 data set is less recent, it is of better quality in terms of completeness and therefore it is decided to use this data set. This survey covers data such as 'Wall Type', describing the type of material used for the walls of a house. It is important to note that the survey design is designed to provide District-level representativeness and a reasonable level of Precision for key socio-economic and agricultural indicators. However, the choice has been made to use the survey data on TA level. The reason is that, although a few TAs are missing, the average amount of observations per TA (excluding the missing values) is almost 57 and therefore sufficient to involve these data into this research. The enumeration areas are shown in Figure 10 with a dot per area. Because of privacy reasons, every dot represents a center of a number of households interviewed and not the actual household.

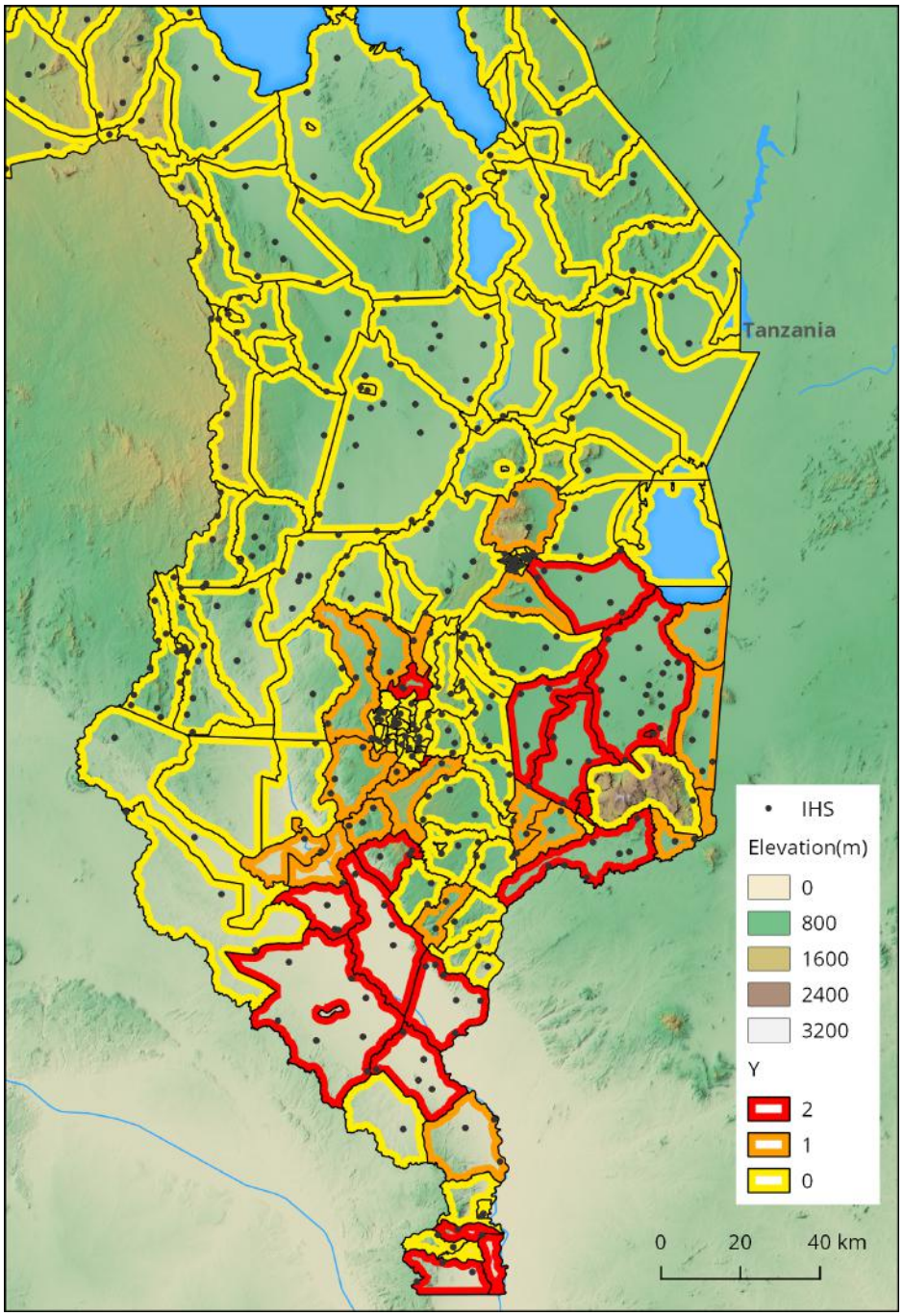

Figure 10: Integrated Household Survey 2010/11 Survey Coordinates

Another important variable in this category is the poverty index. This poverty index is available in the data sets above, but a more detailed index is published on the Worldpop website. Worldpop is a mapping initiative, providing open demographic spatial data. Worldpop produced a high resolution map with a poverty index, based on a Bayesian analysis. Although these data are not 'raw', we decided to incorporate this information into our research for two reasons. First of all, the map is

very detailed and therefore perfect for use on TA level. Secondly, Worldpop produces these maps for different countries around the world. This has the benefit that the indicators can be implemented in further research and expansions of this work, involving other countries.

Lack of Coping Capacity For the last category, infrastructural indicators are included and data about access to health system and communication, such as road length, road density, the number of health facilities and the number of mobile phones are used. The road maps are downloaded from Open Street Maps (OSM). An analysis in QGIS then calculates the total length of the roads and the road density per TA. A similar approach could be used for the number of health institutions, by downloading OSM building data and analysing these data. However, some areas have bad coverage, especially considering buildings. This leads to a biased image of the actual amount of buildings or health facilities in an area. Therefore a different source for the number of health facilities in a TA is used. This source is Malawi Spatial Data Platform (MASDAP). MASDAP provides spatial data about Malawi and is managed by the Malawi Government's Department of Surveys. All organizations active in Malawi can contribute to this platform by sharing their data with the Department of Surveys. The number of mobile phones are available from the Integrated Household Survey (IHS).

4 Methodology

This research's methodology section describes a number of Machine Learning Techniques used to forecast the vulnerability of a TA to floods in Malawi and to point out the most important variables for flood modelling. Four different models are described in this section, namely the CART Decision Tree, Conditional Inference Tree (Ctree) and two types of Random Forests based on these two decision trees. The choices for these models and the differences between them are discussed and next, cross validation, which is used in Section 5 to compare the predictive performance of the models, is described.

4.1 Decision Trees

Decision trees can be used both as a predictive model, as well as a tool to select the most important variables. Decision trees are top-down recursive algorithms, meaning the algorithm starts with one root node containing all observations and keeps partitioning the data until all records in a node belong to the same class y_i or if they have identical attribute values. The root node contains all independent attributes $x_1, x_2, ..., x_m$ and the response variable y. The data will then be partitioned into child nodes of the previous node, based on an attribute test condition, which decides the attribute and node to split on for every new split. This is the first important step in a decision tree algorithm, followed by the second, which is the pruning of the tree. Growing a tree with a stopping criteria that keeps splitting until all records in a node belong to the same class or until they have identical

attribute values, leads to large trees and therefore possibly overfitting. To prevent overfitting, a decision tree algorithm can use an early stopping criteria or pruning. The latter grows a full tree and then removes parts of the tree that have little classification power.

In this research, two different decision trees are discussed. The first one is a well known and oftenly used model, namely the CART model introduced by Breiman, Friedman, Stone, and Olshen (1984). The other is a Conditional Inference Tree (Ctree), which is a more recently introduced algorithm by Hothorn, Hornik, and Zeileis (2006). These two trees differ both on the way of splitting and the way of pruning the tree and have in common that they are able to deal with a categorical response variable that contains more than two categories. For the splitting of the tree, CART uses the Gini Criterion for the selection of the attribute and node to split on, whereas Ctree uses permutation tests to select the attribute to split on and a test statistic to decide the best split point. For the pruning of the tree, CART uses Cost Complexity pruning, whereas Ctree uses an early stopping condition.

Our motivation for the use of CART is that the algorithm is a well known and widely used decision tree algorithm. However, two main problems that arise when using decision trees are overfitting and selection bias. To decrease the effect of overfitting, an early stopping condition or pruning is used, but the selection bias remains. Ctree tries to solve this problem with the use of permutation tests, which is discussed in more detail in Section 4.1.2, after explaining the CART Decision Tree algorithm. Our focus in this research is not only on predictive Accuracy, but also on identifying the most important variables for predicting vulnerability to floods. Ctree may therefore play an important role in the interpretation of the variables that make an area prone to damage by flooding. Both models are discussed in more detail in Sections 4.1.1 and 4.1.2.

4.1.1 CART

The CART algorithm has been introduced in 1984 by Breiman and has been widely used ever since. As mentioned above, the CART algorithm starts building a tree from a single node. For the splitting of the tree, CART uses an attribute test, called the Gini Criterion. This criterion measures the node's purity or homogeneity. The Gini criterion, as mentioned by Breiman et al. (1984), is given by the following formula:

$$i(t) = \sum_{k=1}^{J} \sum_{k \neq l}^{J} p(w_k|t)p(w_l|t), \tag{1}$$

with w_j the j^{th} category of y, $p(w_j|t)$ the proportion of units in node t belonging to the j^{th} category of y and with $j = 1, ..., J$ and $l \neq k$. This formula can also be written as:

$$i(t) = 1 - \sum_{k=1}^{J} p(w_k|t)^2, \tag{2}$$

which is perhaps a more commonly used notation. It can be seen that these formulas are the same in a few steps:

$$
\begin{aligned}
i(t) &= \sum_{k=1}^{J} \sum_{l=1}^{J} p(w_k|t)p(w_l|t) \\
&= \sum_{k=1}^{J} p(w_k|t)(1 - p(w_k|t)) \\
&= \sum_{k=1}^{J} (p(w_k|t) - p(w_k|t)^2) \\
&= \sum_{k=1}^{J} p(w_k|t) - \sum_{k=1}^{J} p(w_k|t)^2 \\
&= 1 - \sum_{k=1}^{J} p(w_k|t)^2,
\end{aligned}
$$

where is used that $\sum_{k=1}^{J} p(w_k|t) = 1$.

The Gini Criterion is calculated for each possible split point and variable. For each of these possibilities, two child nodes are created, t_l and t_r. Then to calculate the decrease in impurity or increase in purity by the split s in node t is given by:

$$\Delta Imp(t, s) = i(t) - p(t_R)i(t_R) - p(t_L)i(t_L), \tag{3}$$

where $p(t_R)$, $p(t_L)$ and $p(t)$ are the proportion of observations that belong to nodes t_R, t_L and t and with $p(t_R) = \frac{n_{t_R}}{n_t}$ and similarly, $p(t_L) = \frac{n_{t_L}}{n_t}$. The highest value in Equation 3 gives the new split.

After the splits in the model have been performed and the tree has been built completely, the tree needs to be pruned. One of the most popular pruning techniques is proposed by Breiman et al. (1984) and uses the following cost-complexity function:

$$R_\alpha(T) = R(T) + \alpha \cdot card(T), \tag{4}$$

where $R(T)$ gives the misclassification rate, $card(T)$ the complexity of the tree, which indicates the number of leaves and α is a tuning parameter showing the trade-off between predictive performance and tree complexity. For every subtree the α can be calculated with the following formula:

$$\alpha = \frac{R(t) - R(S)}{L(S) - 1}, \tag{5}$$

where $R(S)$ is the misclassification rate for the subtree and $R(t)$ for the subtree's best leaf (the leaf that remains when the subtree is pruned) and $L(S)$ the number of leaf nodes in the subtree. The α is calculated for all subtrees and the subtrees with the minimum value of α are pruned. Then for the new tree, the $\alpha's$ are calculated in the same manner and this cost-complexity pruning thus creates a sequence of trees with a different pruning level. Out of this sequence of trees, a final pruned tree needs to be chosen. This is done, using either cross validation or by using a separate validation set.

Then, the tree with the minimum error rate (T_{min}) or the smallest tree within one standard error of T_{min} is selected.

The CART algorithm described above can be applied for numerical and ordinal response variables, but treats them the same and does not use the additional information that ordinal variables have (Archer, 2010). In order to take this additional information into account, the generalized Gini impurity function from Breiman et al. (1984) is now introduced for node t, instead of the Gini criterion used in Formula 2. This formula is given by:

$$i_{GG}(t) = \sum_{k=1}^{J} \sum_{l=1}^{J} C(w_k|w_l) p(w_k|t) p(w_l|t), \tag{6}$$

where $C(w_k|w_l)$ represents the misclassifcation cost of assigning category w_k to a sample unit belonging to category w_l. As can been seen, if $C(w_k|w_l) = 1 \forall k \neq l$, this formula equals Formula 2. For an ordinal response variable, with a set of increasing scores $s_1 < s_2 < ... < s_J$ assigned to the categories of the response variable y, the misclassification costs can be chosen in two different ways, as mentioned in Galimberti, Soffritti, and Di Maso (2012). This is given by the following formulas:

$$C(w_k|w_l) = |s_k - s_l|, \tag{7}$$

or as

$$C(w_k|w_l) = (s_k - s_l)^2, \tag{8}$$

where Formula 7 gives the absolute difference between scores s_k and s_l and Formula 8 the squared difference between them.

The choice for an algorithm that takes into account the ordinal nature of the response variable, also leads to a different choice for the pruning technique. As stated by Galimberti et al. (2012), $R(T)$ as in Formula 4 is computed as the within-node deviance, which implicitly uses the average score for y. Two more suitable techniques are therefore pruning based on the total misclassification rate, given by Formula 9 or on the total misclassification cost given by Formula 10. The formulas are as follows:

$$R_{mr}(T) = \sum_{i=1}^{n} [1 - I_{s_i}(\hat{s}_{i,T})], \tag{9}$$

and

$$R_{mc}(T) = \sum_{i=1}^{n} |s_i - (\hat{s}_{i,T})|, \tag{10}$$

where s_i is the observed score for unit i and $\hat{s}_{i,T}$ is the predicted score for unit i. $I_{s_i}(\hat{s}_{i,T}) = 1$ if $s_i = \hat{s}_{i,T}$ and 0 otherwise.

The implementation of this method is done with the use of the R package *rpartScore*, which enables the users to set the misclassification costs and pruning parameters.

4.1.2 Conditional Inference Trees

Algorithms such as CART generally deal with two main problems: Overfitting and selection bias towards explanatory variables with many possible splits (Kuhn & Johnson, 2013). A possible solution for the selection bias is the use of Conditional Inference Trees. Algorithms such as CART tend to select variables with many possible splits or many missing values, because those are favoured by the Gini Index. For example, the CART algorithm ignores missing values, meaning that a split on a variable will lead to a higher proportion of records ($p(w_k|t)$) that belong to category w_k in case of missing values, which then leads to a lower Gini Criterion $i(t)$, as can be seen in Formula 2. Conditional Inference Trees, however, use a significance test procedure to select variables and therefore solve this problem. Apart from the Conditional Inference Tree having a variable selection procedure based on statistical theory, the tree algorithms are quite similar. Whereas the CART model checks all the possible variables and splits, the Conditional Inference Tree divides this into two steps: (1) the variable selection and (2) the search for the split point. The algorithm, introduced by Hothorn et al. (2006) tests dependency between the response variable and the explanatory variables for every split with hypothesis tests and selects the variable X_j^* with the strongest association (strongest dependency between $\mathbf{Y}$ and X_j). Then, after a variable is selected, the best split point is chosen. This is an iterative process and repeats itself until the null hypothesis of independence cannot be rejected anymore.

Variable Selection For every possible split, the algorithm defines a global null hypothesis H_0 of independence between $\mathbf{Y}$ and all $\mathbf{X}$ with:

$$H_0 = \cap_{j=1}^{m} H_j^0 \text{ and } H_0^j : D(\mathbf{Y}|X_j) = D(\mathbf{Y}), \tag{11}$$

where $D(\mathbf{Y}|X_j)$ is the distribution of $\mathbf{Y}|\mathbf{X}$, which is equal to $D(\mathbf{Y})$ under the null hypothesis. This global hypothesis is divided into m partial hypotheses. Each partial hypothesis, separately tests one variable X_j for association with the response variable $\mathbf{Y}$. If the global null hypothesis is rejected, the variable with the strongest association is chosen. This splitting procedure then continues until the global null hypothesis, and thus all the partial hypotheses, can no longer be rejected. In order to test this hypothesis, a parametric test is necessary. However, the distribution as given below in Formula 12, is unknown under almost all practical circumstances (Hothorn et al., 2006).

$$D(\mathbf{Y}|\mathbf{X}) = D((\mathbf{Y}|X_1, ..., X_m) = D(\mathbf{Y}|f(X_1), ..., f(X_m)) \tag{12}$$

Hothorn et al. (2006) propose the use of permutation testing as a solution, since permutation testing does not require a distribution assumption. The process of permutation testing for all of the m partial hypotheses can be explained in five steps:

1. Calculate a test statistic under the null hypothesis; $\mathbf{T}_0$
2. Calculate a test statistic $\mathbf{T}$ for all permutations of pairs X_j, $\mathbf{Y}$
3. Count the number of $\mathbf{T}$ which are more extreme than $\mathbf{T}_0$, we call this $n_{extreme}$
4. Calculate the p-value, $p = \frac{n_{extreme}}{n_{permutations}}$
5. Reject H_0 if $p < \alpha$, with α being the significance level

This is an iterative process that continues until the global null hypothesis can no longer be rejected. Furthermore, this process is done separately for every partial hypothesis and thus for every explanatory variable X_j. Note that only the permutations of the response variable are calculated.

The test statistic, mentioned in the five steps is given by Formula 13. When using permutation testing, you are free to choose the test statistic and Hothorn et al. (2006) chose a test statistic that is derived from (Strasser & Weber, 1999), as shown below:

$$\mathbf{T}_j(\lambda_n, \mathbf{w}) = vec\Big(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(\mathbf{Y}_i, (\mathbf{Y}_1, ..., \mathbf{Y}_n))^T \Big) \tag{13}$$

where vec stands for the vectorization of the matrix, λ_n is the learning sample, w_i are case weights that are assumed either 0 or 1, depending if a node contains the observation, g_j a transformation of the explanatory variable X_j and h the influence function, which depends on the responses.

Formula 13 gives a very general formulation, covering all possible cases with the use of the transformation- and influence function. However, in our research we only focus on three types of variables: numeric, nominal and ordinal. We first describe the most simplistic case, where both X_j and $\mathbf{Y}$ are numeric. Then we continue with a test statistic for numeric explanatory variables and a nominal response variable and finally we extent this to the transformations necessary for numeric explanatory variables and an ordinal response variable, which is the case in our research.

First of all, when dealing with numeric explanatory variables and a numeric response variable, the influence function h and the transformation function g_j are given by the identity function. Thus:

$$h = \mathbf{Y}_i \text{ and } g_j = X_j, \text{ for } j = 1, ..., m. \tag{14}$$

The test statistic $\mathbf{T}$ can now be written as:

$$\mathbf{T}_j(\lambda_n, \mathbf{w}) = \sum_{i=1}^{n} w_i X_{ji} \mathbf{Y}_i. \tag{15}$$

This test statistic can be standardized as described in the Split Point paragraph below and is proportional to Pearsons correlation coefficient for numerical X_j and $\mathbf{Y}$ (Molnar, 2013). Hothorn et al. (2006) state that due to the flexibility of the transformation and influence functions, special test procedures such as the Spearman test and permutation tests based on ANOVA statistics or correlation coefficients are covered by the framework proposed in their paper. In case of numeric covariates and response variable, this means that the null hypothesis of independence between X_j and $\mathbf{Y}$ can be formulated as: "the correlation between X_j and $\mathbf{Y}$ is equal to zero." To illustrate the five steps explained in this section and the calculation of the test statistic, an example is given here.

Example: In this example, we consider a response variable 'Weekly Income' and one explanatory variable, which is 'Age'. The data available in node t consists of three observations and is given in the table below:

Table 1: Explanatory variable and response variable

Age	**Weekly Income**
20	500
30	1000
40	900

Step 1 The first out of the five steps is the calculation of the test statistic $\mathbf{T}_0$ as given by Formula 14:

$$\mathbf{T}_0 = \sum_{i=1}^{n} w_i X_{ji} \mathbf{Y}_i = 20 * 500 + 30 * 1000 + 40 * 900 = 76000,$$

with w_i being equal to 1 for all these three observations, since they are part of the current node t and $j = Age$. The permutations of the response variable are given in Table 2. Node t contains 3 observations, which results into $3! = 6$ permutations and thus 6 permutated pairs of X_j, $\mathbf{Y}$.

Table 2: Permutations

Permutations					
P_1	P_2	P_3	P_4	P_5	P_6
500	500	1000	1000	900	900
1000	900	500	900	500	1000
900	1000	900	500	1000	500

Step 2 In Step 2, we calculate the test statistic $\mathbf{T}$ for all the permutation pairs. This gives the following values $\mathbf{T}_1 = 76000$, $\mathbf{T}_2 = 77000$, $\mathbf{T}_3 = 71000$, $\mathbf{T}_4 = 67000$, $\mathbf{T}_5 = 73000$, $\mathbf{T}_6 = 68000$. Note that one of the permutations is the original one. This does not influence the results much in

general, but does here, because we are using a very small example. After having calculated all the test statistics, we can calculate the $n_{extreme}$ and the p-value.

Step 3 & 4 The number of extreme cases ($n_{extreme}$), that are larger than 76000 is 1, only for $\mathbf{T}_2 = 77000$. The number of permutations ($n_{permutations}$) is 6. Thus, p-value is given by: $p = \frac{1}{6} = 0.17$.

Step 5 Let's say that for this example, the pre-specified $\alpha = 0.2$. Thus, $p < \alpha$ for the variable X_{Age}, because $0.17 < 0.2$. The null hypothesis is therefore rejected and X_{Age} will be the next variable to split on if it is the only partial null hypothesis that has been rejected. In case of multiple rejected partial null hypotheses, the variable with the lowest p-value will be the next variable to split on.

The five steps described in the example for numerical cases are the same for nominal cases. However, the transformation and influence function are somewhat different and make use of the unit vector. For nominal explanatory variables, $g_j = e_K(X_{ji})$, where K corresponds to the levels of the explanatory variable. For a nominal response variable, the influence function is given by $h = e_J(\mathbf{Y}_i)$, which is a vector with the same dimensionality as the number of categories. In our research the response variables has three categories: '0' Low level of aid-neediness, '1' Moderate level of aid-neediness and '2' High level of aid-neediness. We now consider the response variable to be nominal and the explanatory variables remain numeric. The influence function is now as follows:

$$h = e_J(\mathbf{Y}_i) = \begin{cases} (1,0,0)^T & \text{Category 0, Low level of aid-neediness} \\ (0,1,0)^T & \text{Category 1, Moderate level of aid-neediness} \\ (0,0,1)^T & \text{Category 2, High level of aid-neediness} \end{cases} \tag{16}$$

The transformation for the numeric explanatory variables X_{ji} is again given by the identity function $g_j = X_{ji}$. This results into the following formula for the test statistic $\mathbf{T}$:

$$\mathbf{T}_j(\lambda_n, \mathbf{w}) = vec\Big(\sum_{i=1}^{n} w_i X_{ji} e_J(\mathbf{Y}_i)\Big) = \begin{pmatrix} \sum_{i=1}^{n} X_{ji,low} \\ \sum_{i=1}^{n} X_{ji,moderate} \\ \sum_{i=1}^{n} X_{ji,high} \end{pmatrix}, \tag{17}$$

which is a column vector for variable j with the sum over the observations per category (low, moderate, high) of $\mathbf{Y}$.

In both examples, the test statistic $\mathbf{T}$ can be calculated under the null hypothesis and for all the permutation pairs. Due to the flexibility of the influence and transformation functions, in case of numerical covariates and a nominal response variable, permutation testing for the difference in means can be used when the response variables contains 2 categories and ANOVA when it contains 3 or

more categories. In these cases, the difference in means and the ANOVA F-Statistic can be calculated under the null hypothesis and for all possible permutations. The p-value can then be calculated by dividing the number of extreme cases by the number of permutations and if this p-value is lower than the pre-specified α, the null hypothesis is rejected.

In case of an ordinal response variable and numeric covariates, the test statistic in Formula 13 becomes a linear combination of the test statistic $\mathbf{T}_j$ and is given by the formula:

$$\mathbf{MT}_j(\lambda_n, \mathbf{w}) = vec\Big(\sum_{i=1}^{n} w_i X_{ji}(\xi^T e_J(\mathbf{Y}_i)\Big), \tag{18}$$

where ξ are associated with the score vectors, measured at J levels, which reflect the distances between the levels, with $\xi = (1, 2, 3)$ in this case. $\mathbf{M}$ is a block matrix and (in case of an ordinal response variable only) is given by the formula:

$$\mathbf{M} = \left[\begin{array}{ccc|c|ccc} \xi_1 & & 0 & & \xi_q & & 0 \\ & \ddots & & \dots & & \ddots & \\ 0 & & \xi_1 & & 0 & & \xi_q \end{array}\right] \tag{19}$$

In our case, the number of levels is $J = 3$ and q stands for the dimension of the influence function $e_J(\mathbf{Y}_i)$, which is equal to J. The dimension of the transformation function $g_j = e_K(X_{ji})$ is equal to the number of categories K, but since we are dealing with numeric covariates, this is equal to 1. Thus, $\mathbf{M}$ is now given by the vector $\mathbf{M} = (1, 2, 3)$ and:

$$\mathbf{MT}_j(\lambda_n, \mathbf{w}) = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} \sum_{i=1}^{n} X_{ji,low} \\ \sum_{i=1}^{n} X_{ji,moderate} \\ \sum_{i=1}^{n} X_{ji,high} \end{pmatrix} = \sum_{i=1}^{n} X_{ji,low} + 2 * \sum_{i=1}^{n} X_{ji,moderate} + 3 * \sum_{i=1}^{n} X_{ji,high} \tag{20}$$

Split Point After choosing the variable to split on X_{j^*}, the algorithm searches for the best split point. This can be done with any splitting criterion, such as described by Breiman et al. (1984). However, Hothorn et al. (2006) state that most splitting criteria are not applicable to response variables measured at arbitrary scales and therefore it is preferred to use the permutation test framework as described in this section.

In order to find the right split point, a special case of the test statistic $\mathbf{T}$ can be used, with the following formula:

$$\mathbf{T}^A_{j^*}(\lambda_n, \mathbf{w}) = vec(\sum_{i=1}^{n} w_i I(X_{j^*i} \in A) h(\mathbf{Y}_i, (\mathbf{Y}_1, ..., \mathbf{Y}_n))^T) \tag{21}$$

with:

$$I(X_{j^*i} \in A) = \begin{cases} 1, & X_{j^*i} \in A \\ 0, & X_{j^*i} \notin A, \end{cases} \tag{22}$$

where A is the possible partition of the current observations and the j^* in X_{j^*i} is used to indicate the j^{th} variable that has been selected for the split.

In order to find the best split point, the test statistic $\mathbf{T}_j$ needs to be standardized. Under the null hypothesis, the conditional expectation μ_j and variance Σ_j are known and can be calculated. The conditional expectation and variance of the test statistic $\mathbf{T}_j(\lambda_n, \mathbf{w})$ in Formula 13 are given by:

$$\mu_j = E(\mathbf{T}_j(\lambda_n, \mathbf{w})|\mathbf{S}_j(\lambda_n, \mathbf{w})) \text{ and,} \tag{23}$$

$$\Sigma_j = V(\mathbf{T}_j(\lambda_n, \mathbf{w})|\mathbf{S}_j(\lambda_n, \mathbf{w})), \tag{24}$$

with $\mathbf{T}_j(\lambda_n, \mathbf{w})$ the test statistic and $\mathbf{S}_j(\lambda_n, \mathbf{w})$ the symmetric group of all permutations. Given these two formulas, we are able to standardize the test statistic. The choice for this standardized linear statistic is the maximum of the absolute values of the standardized linear statistic and given by the following formula:

$$c_{max}(\mathbf{t}, \mu, \Sigma) = max \left| \frac{(\mathbf{t} - \mu)_k}{\sqrt{(\Sigma)_{kk}}} \right| \tag{25}$$

The standardized test statistic can be utilized for the two-sample linear statistic given by Formula 21, using $\mu^A_{j^*}$ and $\Sigma^A_{j^*}$. In this case, the standardized test statistic is given by $c(t^A_{j^*}, \mu^A_{j^*}, \Sigma^A_{j^*})$. For all possible subsets A, the split A^* with the maximal c, is given by the following formula and is the point where the next split takes place:

$$A^* = \underset{A}{argmax} c(t^A_{j^*}, \mu^A_{j^*}, \Sigma^A_{j^*}). \tag{26}$$

To have a better understanding of the second step of the algorithm, let us again consider the numeric example with the variables 'Weekly Income' and 'Age'.

Example Continued In the first part of the example, we have established 'Age' to be the variable to split on. The second part of the algorithm is to decide on the split point. Node t has 3 observation and thus 2 possible split points: the split that divides the data into ≤ 20 or > 20 and the split that divides the data into $\{\leq 30$ or $> 30\}$. If we go for the first option (≤ 20 or > 20), A contains $\{20\}$ and A^c contains $\{30, 40\}$. The test statistic $\mathbf{T}$ in that case, according to Formula 21, is given by:

$$\mathbf{T}_{j^*}^A(\lambda_n, \mathbf{w}) = vec(\sum_{i=1}^{n} w_i I(X_{j^*i} \in A) \cdot \mathbf{Y}_i) = \sum_{1:X_{j^*i} \in A} \mathbf{Y}_i = n_A \overline{Y}_A = 1 \cdot 500,$$

knowing that $n_A = 1$. Now, to standardize this test statistic, we calculate $\mu_{j^*}^A$ and $\Sigma_{j^*}^A$. A paper by Molnar (2013) describes the formulas for $\mu_{j^*}^A$ and $\Sigma_{j^*}^A$ for the numeric example. The formula for $\mu_{j^*}^A$ is given by:

$$\mu_{j^*}^A = n_A \overline{Y}_{node} = 1 * \frac{500 + 1000 + 900}{3} = 800,$$

and the $\Sigma_{j^*}^A$ can be given by:

$$\Sigma_{j^*}^A = Var(\mathbf{Y}_{node}) \cdot Var(Z),$$

with $Z \sim B(n_{node}, \frac{n_A}{n_{node}})$, a binomial distribution with $n = n_{node}$ and $p = \frac{n_A}{n_{node}}$. The variances of $\mathbf{Y}$ and Z can now be calculated, with $Var(\mathbf{Y}_{node}) = 70000$ and $Var(Z) = np(1-p) = 3 * \frac{1}{3} * \frac{2}{3} = \frac{2}{3}$ and thus:

$$\Sigma_{j^*}^A = 70000 * \frac{2}{3} \approx 46667.$$

Then the standardized test statistic is given by:

$$c_{max}(\mathbf{t}, \mu, \Sigma) = max \left| \frac{(\mathbf{t} - \mu)_k}{\sqrt{(\Sigma)_{kk}}} \right| = n_A \left| \frac{\overline{Y}_A - \overline{Y}_{node}}{\sqrt{Var(\mathbf{Y}_{node}) \cdot Var(Z)}} \right| = 1 * \left| \frac{500 - 800}{46667} \right| \approx 0.006. \quad (27)$$

For the other split, with $A = \{20, 30\}$, the standardized test statistic can be calculated in the same way, which leads to $c_{max} \approx 0.004$. The maximum c_{max} decides what the split point is. Thus, $c_{max} \approx 0.006$ splits the data into two groups: $X_{age} \leq 20$ and $X_{age} > 20$, with $A^* = \{20\}$. When looking at Formula 27, it can be seen that maximizing the c_{max} means finding the partition where the difference between the mean of $\mathbf{Y}$ in the node and the mean of $\mathbf{Y}$ in the partition is the largest and where the number of observations in A is large.

It is important to note that the Conditional Inference Tree algorithm uses a simple Bonferroni correction to correct for multiple testing, thus for the number of hypotheses m. The Bonferroni correction tests every individual hypothesis at a significance level of α/m.

4.1.3 Random Forest

In this section two Random Forest Algorithms are described, based on the decision tree algorithms described in Subsections 4.1.1 and 4.1.2. These Random Forests can be modelled by the R packages *randomforest* and *cforest*. Random Forests use an ensemble of decision trees instead of a single tree model, to increase Accuracy and were introduced by Breiman (2001). Random Forests are famous for

their Accuracy and ability to handle large data sets. The ability to handle big data sets is especially useful for future purposes, when adding new disasters and areas to improve the Accuracy of the model.

In short, a Random Forest grows as follows: A number of trees are created by applying CART or Ctree to bootstrap samples from the original data. In each tree, for every split a number of $m < M$ input variables is selected at random. All of these trees are grown without pruning and form the Random Forest.

For each tree in the Random Forest, a different bootstrap sample is used. About one-third of the cases are left out and not used for the construction of the k^{th} tree. This is called the out-of-bag (OOB) data, which is used to calculate the OOB error rate. Since the trees in the Random Forest are different, each observation gets assigned to a class by each tree and is ultimately assigned to a class by using the majority vote from all trees. In each iteration, so for each individual tree that is generated, the OOB error rate is calculated. This error rate gives an estimate of the out-of-sample error rate. In each iteration, the test cases are labeled according to the majority vote. The proportion of times that it is not equal to the actual class averaged over all cases is the OOB error estimate.

Random Forests have two ways of dealing with missing values in the training set. The simplest solution, is to use a "rough" approach with a mean/median imputation for non-categorical variables and for categorical the most frequent non-missing value, performed by the R function *na.roughfix*. A more sophisticated approach is given by the function *rfImpute*, which makes use of proximities. The first step of this method is again a rough implementation of missing values. Then, a proximity matrix is constructed and updated for every tree in the Random Forest. The updating of the proximity matrix is done as follows; all observations are classified by the Random Forest, every time a new tree is added, and for all observations that end up in the same terminal node, their proximity is increased by one. After the construction of the Random Forest, the proximity matrix is used to replace the missing values by the imputed values that are most similar, according to the proximity matrix. For continuous missing values, the imputed value is the weighted average of the non-missing observations, where the weights are the proximities. For the categorical missing values, the imputed value is the category with the largest average proximity. This process is repeated a number of (by default 5) times, where every new iteration starts with the imputed values from the previous iteration instead of the rough implementation of the missing values.

The two functions used to program the CART Random Forest and Conditional Random Forest are respectively, *randomforest* and *cforest*. The main difference between these two algorithms is the decision trees they are based upon. The *cforest* algorithm is described as an implementation of the Random Forest and bagging ensemble algorithms utilizing Conditional Inference Trees from Hothorn, Hornik, Strobl, and Zeileis (2013) as base learners, whereas the CART Random Forest uses the CART Decision Trees as base learners. For these two Random Forests, the categorical response is handled

as a nominal variable and not as an ordinal one, as for the decision trees.

4.1.4 Repeated Cross Validation

For the comparison of the predictive ability of the models cross validation is used. The data set used for this research is small and therefore makes dividing it into a training and test set not possible. Therefore, to give an approximation of the model performance on unseen data, cross validation is used.

Cross validation splits the data into k random data sets, $D_1, D_2, ..., D_k$. Then $k-1$ data sets are used for the training of the model and 1 set is used as the test set. This process is repeated k times until every set has been used exactly once for testing and $k-1$ times as part of the training set. The reason to use cross validation is to get a more accurate idea of the predictive performance on unseen data. To make the estimate even more reliable, repeated cross validation can be used, in which the k-fold cross validation is repeated a n number of times. For this research a 5-fold cross validation with 100 repeats is used.

5 Results

In this section, the outcomes of the methodology that are used in this research are discussed. We discuss the outcomes of two CART models (the default and pruned CART model), two Conditional Inference Tree models (the default- and the tuned Ctree) and two tuned Random Forest models, the CART Random Forest and the Conditional Forest, that are based on the CART Decision Tree and Conditional Inference Tree, respectively. Firstly, the models and their features are interpreted and discussed, then the repeated cross validated results are analysed and compared. As mentioned in Section 4, note that for the pruning of CART, cross validation is already used and for the growing of the Random Forests bagging is used. Repeated cross validation might seem like an odd choice, when Random Forests can be evaluated using the OOB error rate. However, this is a good method to evaluate all models in the same way. Furthermore, the data set is too small to make use of a hold-out sample and gathering data about another flood is not possible. Therefore, gathering more data about more floods should be a direct follow-up of this research.

Another important note is the way these models handle missing values. The CART model uses 'na.omit', the CART Random Forest, 'na.fail' and the Ctree and Conditional Forest, 'na.pass'. 'Na.omit' omits the rows for which (at least) one of the column has missing values. 'Na.fail' returns an error message and won't run the model unless the user deals with these missing values himself and 'na.pass' returns these observations unchanged by the model. Throwing out observations is costly, especially when working with a small data set. Furthermore, in decision trees, the most important variables for splits might not be the ones with missing values. Thus, throwing out an

observation that has a missing value on another variable (one that might not be used by the model) is unfortunate. Due to these reasons and to be consistent, this research uses median imputation for all algorithms, which can be implemented by the 'na.roughfix' function in R.

5.1 CART

As mentioned in the methodology, to code the CART model in R, the functions *rpart* and *rpartScore* can be utilised. To take into account the ordinal nature of the response variable, *rpartScore* is used for this research. When using this function with the default settings, it is important to understand the stopping conditions, which are: 1. If a node contains less than 20 observations, the splitting of the tree stops (*minsplit* parameter), and 2. The α or complexity parameter (*cp*) is set to 0.01. The latter means that if a next split does not lead to a decrease of the insample (relative error) of 0.01 or more, a next split won't take place. With merely 79 observations, we expect the default tree to be rather small, which is true and can be seen in Figure 11.

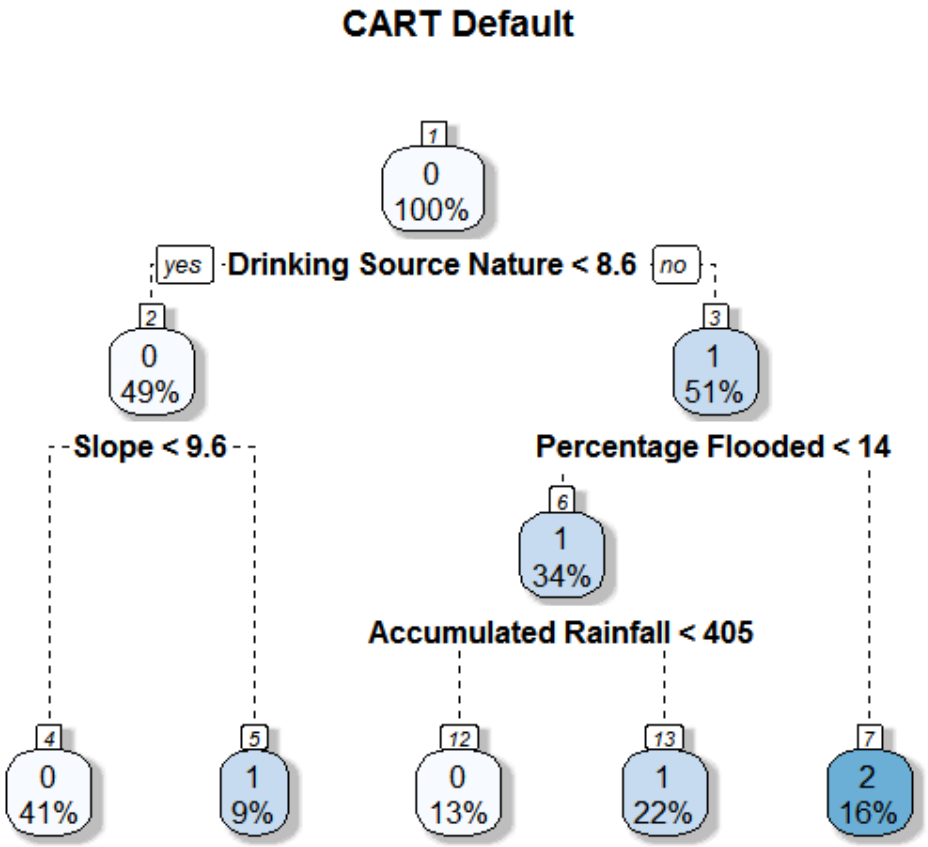

Figure 11: Default (Pre-Pruned) CART Decision Tree

For the pruning of the tree, a bigger default tree would be ideal, because pruning the (already small) default tree most likely leads to an even smaller tree. For the final choice of the pruned tree, the tree with the minimum error rate can be chosen. This is called the SE-0 rule. However, the minimum error tree may be sensitive to changes in the parameter values and therefore another choice can be the smallest tree within one standard error of the tree with the minimum error rate. This is called the SE-1 rule. Nonetheless, both rules result in the same tree, consisting of only one split. Therefore, a bigger tree is created, by changing the default setting.

One way of changing the default settings, is changing the size of the *cp*. However, in this case, the tree stops splitting because of the *minsplit* criterion and not because of the size of α. Therefore, the CART model is run again with $minsplit = 10$, so half the size of the default, which is 20. It would be possible to tune this parameter, using cross validation, but (repeated) cross validation is already used to compare the models. Tuning the *minsplit* parameter, using the in-sample Accuracy, would lead to the smallest number of *minsplit* and is thus not a correct method either. Simultaneously changing α from the default 0.01 to 0 does not change the tree. Changing the default settings in this way, leads to a tree with a number of 16 splits that obviously overfits, with an insample performance of almost 99%. However when pruning this tree with the SE-1 rule (the SE-0 rule would pick the full tree), leads to the tree, shown in Figure 12. When comparing the default and pruned (non-default) tree, it is easy to see that the pruned tree is not a shorter version of the default one, but a different tree. The split on the variable 'Slope' only appears in the default tree, while the 'Number of Health Clinics' only appears in the pruned tree. This can be explained by the different choice for the *minsplit* parameter. With a higher *minsplit*, the split on 'Number of Health Clinics' is not possible, because that node only consists of 13 observations. By lowering the *minsplit* to 10, this split is possible and the preferred one, using the Gini Index.

After having constructed the trees, we look at the results and interpret them. The first splits made by the trees are the most important ones. Among the most important variables, 'Drinking Source Nature', 'Flood Percentage' and 'Rainfall' appear in both trees. The most important split (in terms of decrease in Gini Index) is the variable, 'Drinking Source Nature'. This variable is gathered on District level, meaning that if a TA is part of a district where 8.6% or more of its drinking water comes from natural drinking sources, the TAs will need more help in case of a flood and are predicted to be in the (category 1) Moderate aid-neediness group. If this percentage is lower than this 8.6% on District level, the TAs are predicted to be in the (category 0) Low aid-neediness group. This variable could be seen as an indicator of the quality of the water infrastructure. Areas with a high percentage of people getting their water from natural sources, might indicate more vulnerable groups. TAs that have a higher percentage of drinking water from nature sources and where the percentage of the TA that was flooded is more than 14%, end up in the High aid-neediness group. If the 'Flood Percentage' is lower than this 14%, the TAs stay in the Moderate aid-neediness group. For this latter group,

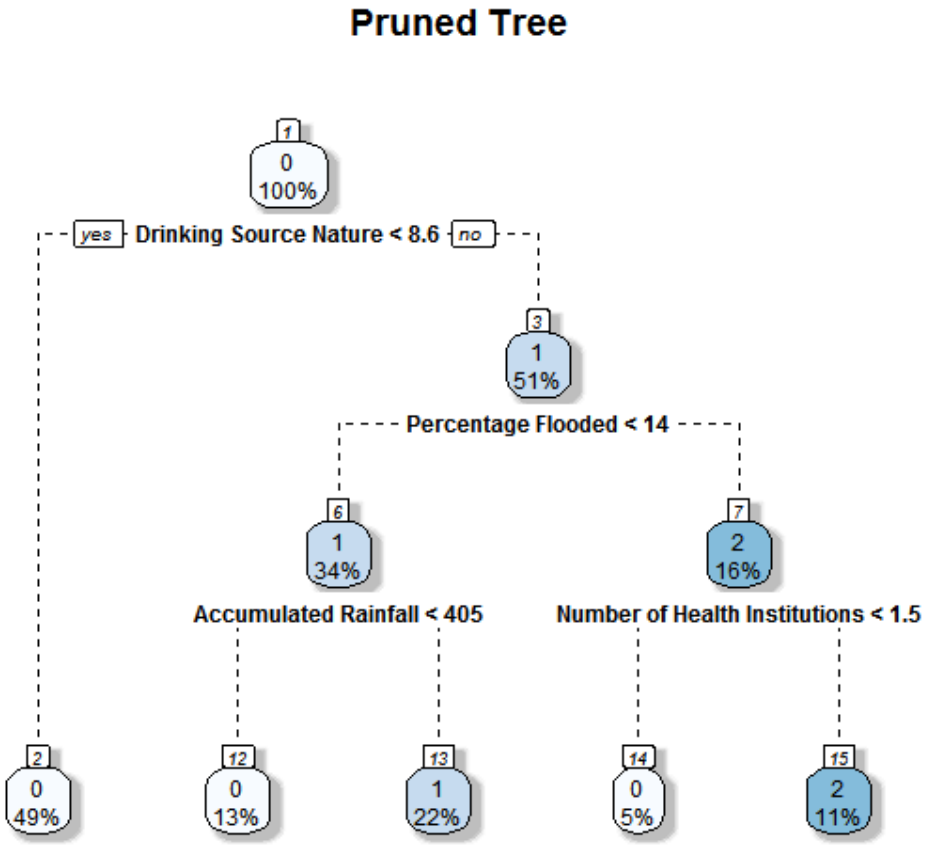

Figure 12: CART Pruned Decision Tree with (minsplit 10)

the accumulated rainfall during the 5 days before the flood decides whether these TAs will stay in the Moderate aid-neediness group or won't need any help. If the accumulated rainfall during these 5 days is lower than 405mm, they are categorized as 0.

Then, for the two variables that are different in the two trees; 'Slope' and 'Number of health clinics', the result can be interpreted as follows. For the default CART tree, the group where the percentage of drinking source that comes from nature is lower than 8.6% on District level, or in different words, where the drinking source in that District comes for roughly 91% from other sources predicts these TAs as the Low aid-neediness (0) category. This group can then be split into TAs with a high slope (more than 9.6 degrees), which can be categorized as Moderate aid-neediness (1) during a flood and with a low slope, which can be categorized as category 0. The slope can be an indicator of how fast water can move through the area. With higher slopes, water from the flood and rainfall could move faster and leads to more destruction. For the pruned tree, the TAs that have a high percentage of 'Drinking Source Nature' and a 'Flood Percentage' higher than 14%, are predicted to

be in the High aid-neediness (2) category. However, if the 'Number of Health Institutions' for that TA is higher than 2, more help is needed and if it is lower, they end up in category 1. This seems counter intuitive, but might be due to the fact that areas with more health institutions generally have a higher population density. This decision underlines the fact that these models should be used as a guidance and not be fully relied on to make decisions. A model can not make ethical decisions and the allocation of help and the amount of people that can be helped depend on much more, such as the location, reachability of an area, resources etc. Therefore, human decision making remains an essential part of these operations.

5.2 CTREE

One of the first things that we notice when observing the Condiftional Inference Tree output in Figure 13 is that the tree only consists of one split. Ctree's null hypothesis for every split, stating independence between the explanatory variables and the response variable, is thus only rejected for the first split. It is possible to get slightly larger trees, by decreasing the amount of explanatory variables. The difference in tree size between the data set with all pre-selected variables and the one with only a few explanatory variables is explained by the Bonferroni correction. The Bonferroni correction corrects for multiple testing (the number of hypotheses tested). For the Conditional Inference Tree, the dependence between the explanatory variable and the response variable is tested separately for every explanatory variable, meaning that the number of partial hypotheses is equal to the amount of explanatory variables, which is corrected for using the Bonferroni correction.

The default α for Ctree is 0.05 and with the low number of observations in our data set, this results into a small tree with only one split. A strategy to assure that any type of dependence is detected could be, to increase the significance level α (Hothorn et al., 2006). Hothorn et al. (2006) furthermore mention that the α controls the probability of falsely rejecting the null hypothesis in each node and thus the typical trade-off between the Type I and Type II error. The Type I error (α), or False Positive, is the error of rejecting the null hypothesis when true. The Type II error, or False Negative, is the error of not rejecting a false null hypothesis. Most importantly, this research aims to find the most important variables that influence the level of aid-neediness. In order to build a bigger tree and find more important variables, the Ctree model is run again with a higher α of 0.30. This increases the Type I error and thus decreases the Type II error, because of the inverse relationship between the two. We are thus willing to have a larger Type I error, to decrease the Type II error and ensure that any type of dependence is detected, which leads to a bigger tree.

This second tree is depicted in Figure 14. Although another split is added to the model, the tree is still small, with only two splits. Whereas the CART model picks 'Drinking Source Nature' as most important split, according to the Gini Index, the Ctree with permutation testing chooses 'Flood Percentage' as most important variable. These two splits are also the two most important

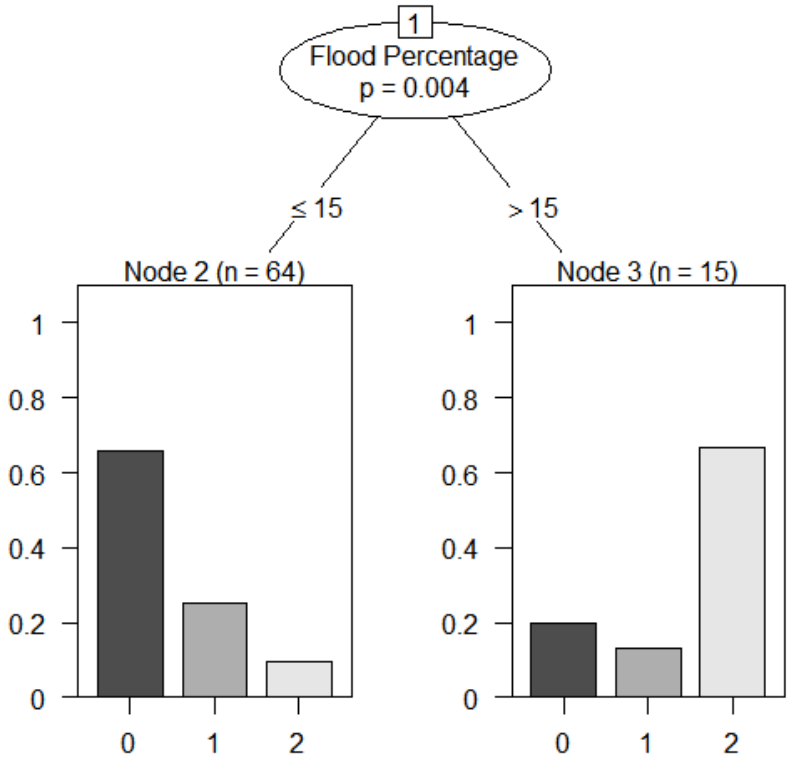

Figure 13: Default Ctree

ones in CART and therefore underline the importance of these variables. The interpretation is similar to the interpretation of the splits in the CART model, although the numbers are slightly different. The Ctree creates a split on 15% 'Flood Percentage' instead of on the 14% as in the CART model. For 'Drinking Source Nature', the split is created on 9.9% instead of the 8.6% by the CART model. These are the results of the difference in split criterion.

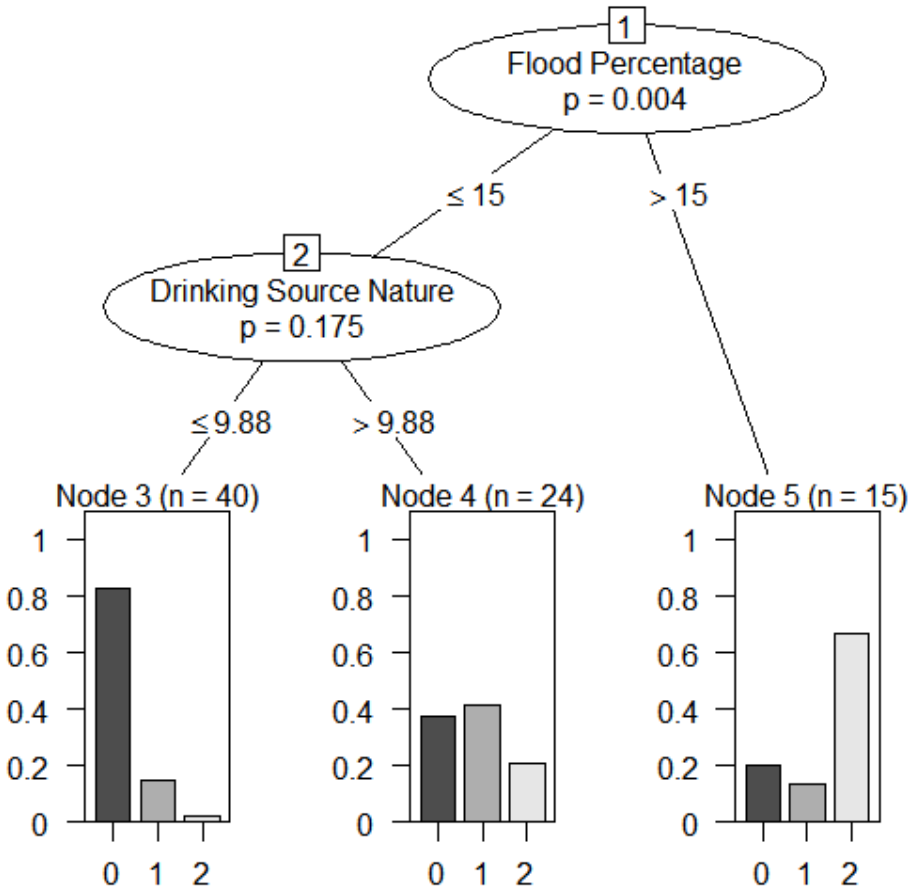

Figure 14: Ctree with $\alpha = 30\%$

5.3 Random Forest CART

As mentioned in the Methodology section, Random Forests grow a certain number of trees (*ntree* parameter in R) and for every split in a tree a number of m variables are chosen as split candidates (*mtry* parameter in R). For both the CART Random Forest and Conditional Forest the default number of trees grown is 500. For this research, dealing with a rather small data set, the computation time is low. Therefore, this parameter does not need to be set lower. However, the number of variables chosen per split is an important factor. Especially for the Conditional Forest, where fewer variables per split, generally lead to bigger trees. For this research, the CART Random Forest ran all the different possible values of *mtry* with the default settings and plotted them against the OOB error rate. The results are shown in Figure 15. The OOB error rate was expected to decrease until a certain minimum has been reached and then to increase again, similar to the shape of a convex parabola. However, when looking at the graph, it is easy to see that this is not the case. The value for *mtry* with the lowest OOB error rate (0.3038) that generates the smallest tree is 9. At this point the graph

seems highly volatile and the difference in OOB error rate between the next value of *mtry* is 0.05. In order to have a more stable *mtry*, that is not influenced by external factors such as the choice of the random seed, the value of *mtry* was chosen to be 21. An even better approach to get more stable results would be to use (repeated) cross validation to tune the *mtry* parameter. However, the repeated cross validation is used to compare the Accuracy of the models and can therefore not be used here.

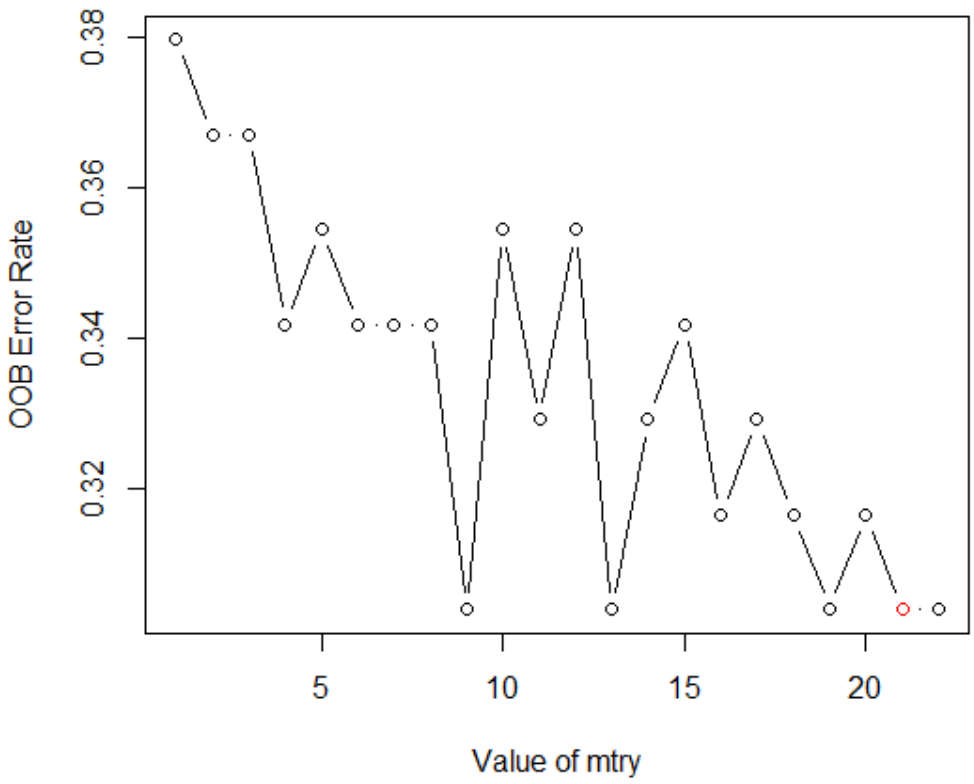

Figure 15: Random Forest Mtry

The predictive abilities of the model are discussed in the Repeated Cross Validation section. The interpretation can be done with two variable importance plots in Figures 16 and 17. The graph on the left shows the mean decrease in GINI index, which is the total decrease in node impurities from splitting on the variable, averaged over all trees (Liaw, Wiener, Breiman, & Cutler, 2009). Thus, a higher mean decrease in GINI Index, leads to a split with purer nodes. The graph on the right depicts the decrease in OOB Accuracy and is therefore more relevant. For every tree in the forest the prediction error on the OOB data is calculated as well as the prediction error when permuting every

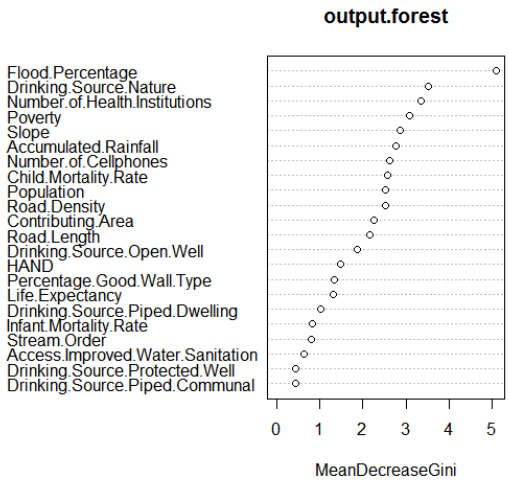

Figure 16: Mean Decrease Gini Index

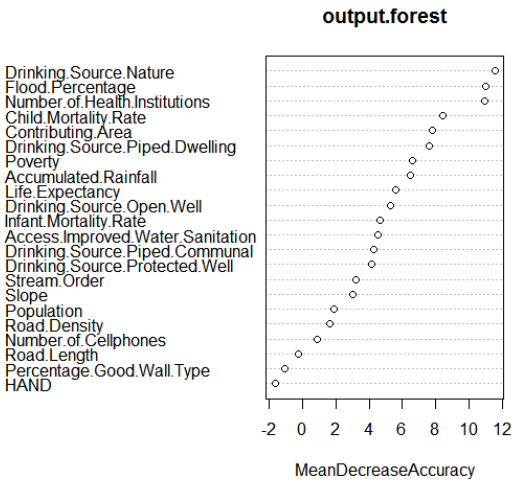

Figure 17: Mean Decrease Accuracy

predictor variable. Then the difference between these two is averaged over all trees, and normalized by the standard deviation of the differences (Liaw et al., 2009). The variables that rank high on Mean Decrease in Accuracy are the most important ones.

In both graphs, although in a different order, the three most important variables are the same, namely: 'Flood Percentage', 'Drinking Source Nature' and 'Number of Health Institutions'. For the other variables, the differences are bigger. Although 'Poverty' and 'Rainfall' are not that far apart in both graphs, the 'Slope', for example, has a high decrease in Gini Index but not in Accuracy. On the other hand, the variable 'Drinking Source from Piped Dwelling' has a relatively high decrease in Accuracy, but not in Gini Index. This means that, although splitting on this variable does not lead to very pure nodes, the split is important for an increase in Accuracy. The results are in line with the Ctree and CART Decision Tree, that both selected 'Flood Percentage' and 'Drinking Source Nature' as most important variables as well.

5.4 Conditional Forest

For the Conditional Forest, the default number of trees grown is also 500 and the *mtry* is tuned to return the tree with the maximum OOB Accuracy. The results of tuning this *mtry* parameter are depicted in Figure 18. We can see that the value of *mtry* with the smallest tree and lowest OOB error rate is 6. This Figure, compared to the *mtry* graph from the CART Random Forest, does show a decrease in error rate and then a stabilisation of the results, therefore making it possible to choose the smallest tree with the lowest OOB error rate. Another difference with the CART Random Forest is that in this graph, the points with an *mtry* close to each other, often have the exact same

OOB error rate. A possible reason is that these levels of *mtry* lead to the same trees (for the same random seed). Conditional Inference Trees tend to be smaller than the CART Trees, as can be seen in Figures 13 and 14 and especially for a larger *mtry*, the Bonferroni correction will be more strict and therefore lead to smaller trees that are more likely to be the same. To be more certain that 6 is the correct value of *mtry*, we would like to use (repeated) cross validation. However, just like in Section 5.3, repeated cross validation is used for the comparison of the models and can therefore not be used twice.

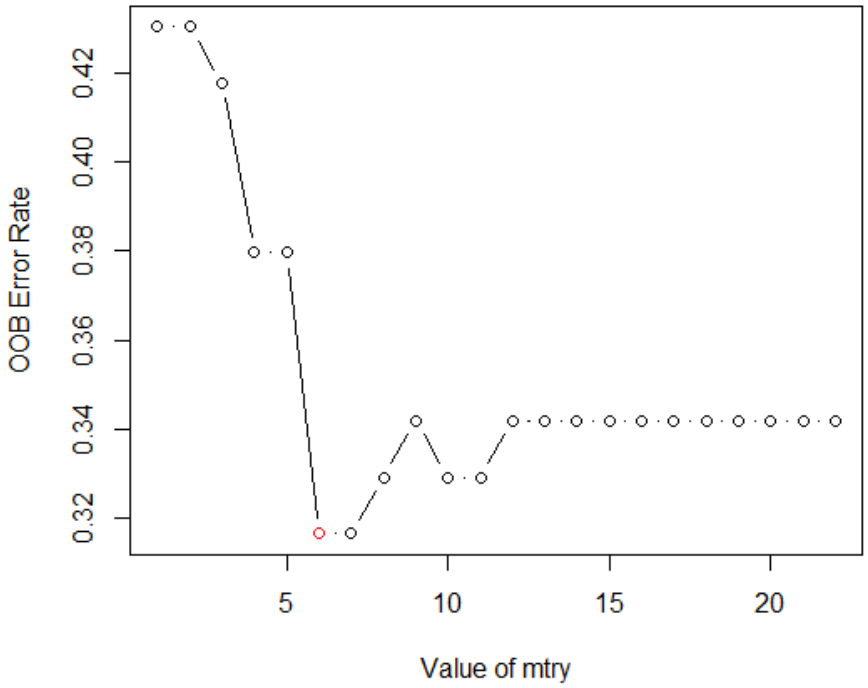

Figure 18: Conditional Forest Mtry

In Figure 19 the variable importance in the Conditional Forest Model is depicted. 'Flood Percentage' and 'Drinking Source Nature' are again the most important variables and have been among the most important ones in all models. Then 'Child Mortality Rate' and 'Drinking Source Open Well' play an important role, but already less than the previous two and it can be seen in the graph that the other variables are similar when it comes to variable importance.

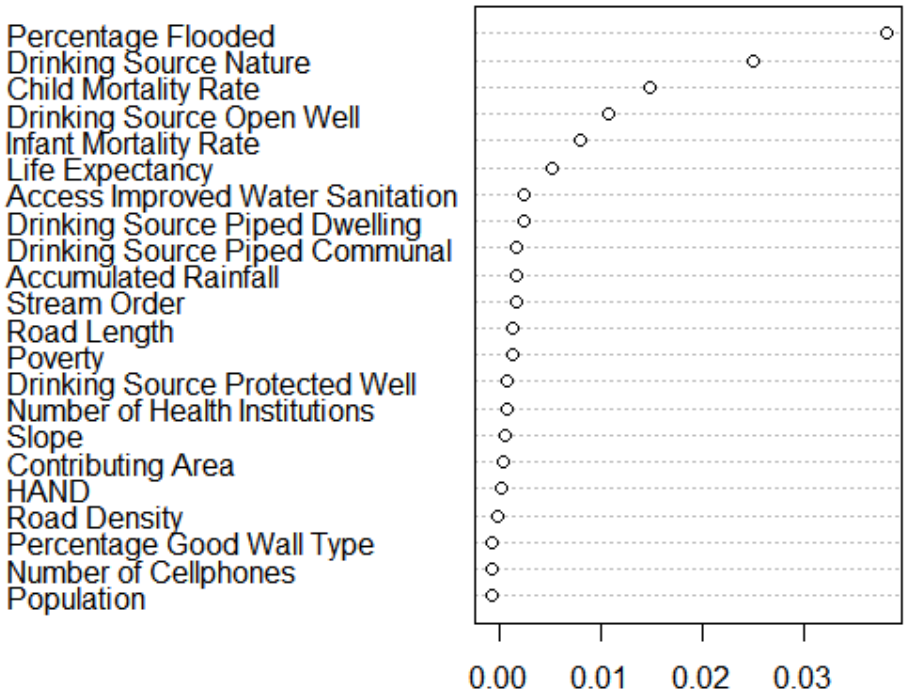

Figure 19: Conditional Forest Variable Importance

5.5 Cross Validated Results

This section discusses the cross validated results of the CART Decision Tree, Ctree, CART Random Forest and Conditional Forest. In order to not over-complicate this section, we only incorporate the default CART and Ctree for comparison and not the 'tuned' versions. A repeated 5-fold cross validation with 100 repeats is used.

First of all, the repeated cross validated 5-fold Accuracy is compared for all models. The Accuracy for CART is 52%, for Ctree 55%, for the Random Forest 64% and for the Conditional Forest 58%. Thus, the OOB estimates were somewhat optimistic, compared to the repeated cross validated results. The model with the highest Accuracy is the Random Forest, which is a popular model because of its predictive ability and therefore not surprisingly the best model here. The Conditional Forest is somewhat closer, with 58% and the Decision Trees have quite a low Accuracy. Especially the CART Decision Tree is low, even though its results are easier to interpret than the Random Forests. However, it is good to have a second look at the models with different measurements. These models

could for example predict the 0 group very well, but the other groups poorly and might give a good Accuracy but are not suitable to use in practice. Therefore, a closer look at the results is necessary.

One option would be to interpret the confusion matrix. However, due to the 100 repeats, this research ends up with 100 confusion matrices. To interpret these results, two types of transformations can be considered; Micro-averaging (summing all the confusion matrices) and Macro-averaging. These transformations result in a Precision and Recall per category, with '0' Low level of aid-neediness, '1' Moderate level of aid-neediness and '2' High level of aid-neediness. Sokolova and Lapalme (2009) state that Macro-averaging treats all classes equally while Micro-averaging favors bigger classes. In this case, Micro-averaging would favor the 0 class. For this research, predicting the 1 and 2 categories is more important, because the focus is on correctly predicting the areas that are in most need of aid. Therefore, Macro-averaging is in favor, but nonetheless the results of both methods are presented in Table 3.

Table 3: Micro & Macro Precision & Recall

	CART			CTREE		
	Micro Precision	Macro Precision	Recall	Micro Precision	Macro Precision	Recall
0	0.682	0.681	0.664	0.620	0.618	0.830
1	0.274	0.270	0.302	0.068	0.150	0.032
2	0.408	0.403	0.376	0.379	0.368	0.340

	Random Forest			Conditional Forest		
	Micro Precision	Macro Precision	Recall	Micro Precision	Macro Precision	Recall
0	0.733	0.732	0.840	0.619	0.618	0.933
1	0.430	0.433	0.346	0.117	0.152	0.024
2	0.503	0.503	0.411	0.417	0.421	0.216

In this table, the Micro and Macro Precision and Recall can be found. Precision is defined by $Precision = \frac{TP}{TP+FP}$, so the number of True Positives divided by the sum of the True Positives and False Positives. For a category x this means, how many out of all the observations you've predicted to be x are correct? The Recall is defined as $Recall = \frac{TP}{TP+FN}$, with FN the False Negatives. In other words, what proportion of an actual category x are predicted as x? The Micro-Averaged Precision is calculated as follows:

$$\text{Micro-Average Precision} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)}, \tag{28}$$

with TP_i the number of True Positives for repeat i, with a total of n repeats. The Micro-Averaged

Recall can be given by the same formula, by replacing FP_i for FN_i. The Macro-Average Precision is given by:

$$\text{Macro-Average Precision} = \frac{1}{n}\sum_{i=1}^{n} P_i, \tag{29}$$

with P_i the Precision for repeat i and n the number of repeats. The Macro-Average Recall can be calculated in the same manner, by replacing the Precision (P_i) for Recall (R_i).

The Accuracy and Recall are the same for both methods and Table 3 therefore only contains three columns instead of four. Micro- and Macro-Averaging are often used to compare the Precision and Recall for multiple data sets. In that case, the total number of observations and the number of observations per category would differ. In our case, however, the data set remains the same. For every cross validation, the number of predicted observations per category differs, but the actual number of observations per category does not change. The Recall, given by $\frac{TP}{TP+FN}$, has as denominator $TP + FN$. This denominator is equal to the actual number of observations per category and thus remains the same for every cross validation. Due to the unchanging denominator, the Micro-Average Recall is equal to the Macro-Average Recall.

When looking at the results in Table 3, it is noticeable that the differences between the Micro-Average and Macro-Average Precision are very small. Therefore, we only look at the Macro-Average Precision. For almost all categories, the Precision and Recall are highest for the Random Forest, as well as the Accuracy. This model seems to perform best on almost all measurements, except for the interpretation which is clearer for the Decision Trees. For the Ctree and Conditional Forest, The Precision and Recall for the category 1 class (Moderate aid-neediness) are low. Ctree has a Recall of 0.032, meaning out of the actual number of 1's, only roughly 3% is predicted correctly. The Precision for class 0 and 2 are similar for Ctree, CART and the Conditional Forest, although some differences in Recall exist. The CART model does worse on Recall for the 0 category, which contributes to the low Accuracy rate. The Conditional Forest does a very good job predicting the 0 category correctly, with a Recall of 0.933. However, the Recall for the 2's is the lowest out of all models. Overall, the models do relatively well predicting the 0 category, but struggle to correctly predict the 1 and 2 categories. This research deals with a bias in the distribution of the observations over the categories, with 45 observations belonging to category 0 and respectively 18 and 16 to categories 1 and 2. This could be one of the reasons, besides the lack of data.

In order to check the robustness of the results, the variance of the Accuracy is calculated for the 100 repeats. These variances are respectively 0.0027 for CART, 0.0014 for Ctree, 0.0013 for the Random Forest and 0.0005 for the Conditional Forest. The variances are all small and there is no reason to favor one model over the other regarding the consistency over the 100 repeated k-folds.

6 Conclusion

The research question, as defined in Section 1, is: **Can Machine Learning techniques be used to better allocate help in case of floods in Malawi?** To answer this question, two subquestions were formulated regarding: 1. The type of Machine Learning techniques that can be used and the differences among them, and 2. The most important factors (variables), selected by these methods.

Various research, such as Kohara and Sugiyama (2013), have shown possibilities in the field of (statistical) modelling to improve humanitarian aid and we believe that this research definitely shows possibilities in this field as well, although multiple limitations exist. This research uses four different types of models; the CART Decision Tree, Conditional Inference Tree, CART Random Forest and the Conditional Random Forest. The methodological differences between the different models are discussed, then the most important variables from every model are analyzed and finally the models are compared in terms of cross validated Accuracy and their Precision and Recall.

The models show similar results considering variable importance and predict with a repeated cross validated Accuracy of between 52% and 64%, depending on the model. This Accuracy, which is not outstandingly high, could be hopefully further improved by increasing the number of observations, for example through the addition of more flood events. Besides increasing the Accuracy, this would also make the results more generalizable. We now discuss the two subquestions in more detail, followed by a extensive discussion about the limitations and possibilities for further research in Section 7.

6.1 Machine Learning Techniques

For the four different types of models, the main differences are discussed elaborately in the Methodology Section and come down to the differences in interpretation and variable selection. When looking at the model results, a few things come to mind. First of all, the two Random Forest algorithms perform best when it comes to Accuracy. Especially the CART Random Forest outperforms the other algorithms. The Ctree and Conditional Forest have problems predicting the Moderate aid-neediness (1) group correctly. Furthermore, the Ctree is rather small, especially compared to the CART Decision Tree, with only one split for the default $\alpha = 0.05$. All models struggle with predicting the Moderate (1) and High (2) aid-neediness categories correctly and perform best (both in Precision and Recall) on predicting the Low aid-neediness category. The reason for this is that the data are skewed, with an amount of observations in this category that is twice as large as either the Moderate or High aid-neediness category. Overall it seems that the Random Forests in general, but specifically the CART Random Forest performs best on predictive Accuracy.

The Ctree is very small, because of the low significance, but when enhanced with more data it could give better predictions. Therefore Decision Trees could be used in the future too, to get more insights into the data. Especially with more data, these splits become more accurate and provide

more information than the Random Forests, because the exact split point is shown. Random Forests also perform well on a larger sample size and reduce training and testing error rate as is mentioned by Wang et al. (2015) and would therefore be a good choice when the focus is on predictive performance. The Decision Trees and especially the CART Decision Tree give a clear overview of the split and the most important data. However, they perform worse according to predictive performance and thus, to be more certain about the exact point of split, more data would be needed. The Ctree and Conditional Forest are good methods to solve the problem of selection bias. However, more and better quality of data is needed to ensure the tree grows bigger and creates a higher amount of splits. With the low amount of observations and high amount of pre-selected variables, the Bonferroni correction prevents the tree from growing larger. More data is therefore the solution to better models and a better Accuracy.

6.2 Most Important Variables

The two most important variables are the 'Flood Percentage' and the 'Drinking Source Nature'. These two variables appear as the most important variables in all of the models. The 'Flood Percentage' is intuitively an important variable. The percentage of a TA that is flooded, should give a good indicator of how badly the TA is affected and should therefore always be retrieved during a disaster to give an estimate for the level of aid needed. The variable 'Drinking Source Nature' might not be as intuitive as the 'Flood Percentage', but seems to be a proxy for the overall vulnerability of an area. This is definitely an interesting variable to further investigate. Other variables that are marked as important by a number of the models, but less frequent are: 'Rainfall' and 'Number of Health Clinics', 'Child Mortality Rate', 'Poverty Index' and 'Slope'. The 'Rainfall' and 'Slope' give an indication of the amount of water that fell in the region and how fast it will reach other areas. The 'Child Mortality Rate' and 'Poverty Index' give an indication of the vulnerability of an area and the 'Number of Health Clinics' seems to be an indicator of the population density, since the CART Decision Tree labeled TAs with on average more than 1.5 health institutions as High aid-neediness areas.

The models in this research clearly point out two variables that are considered as the most important ones. For the other variables, the models are less consistent and although some indication is given, more data needs to be evaluated to better generalize the results. Even for the two most important variables this is recommended. Especially the variable 'Drinking Source Nature' may be measured differently per country and might take some extra research in case of enhancing the models with more data.

7 Limitations & Further Research

The biggest limitation of this research concerns the data. Here, the different limitations and possibilities for further research are discussed, regarding five different categories: Quantity, Quality, Availability, Completeness and Recency. Starting with the biggest limitation and therefore the largest possible improvement, which is the Quantity.

Quantity The scope of this research is the 2015 Flood in Malawi and covers 79 different areas, known as Traditional Authorities. This is a relatively small number of observations and only contains data about one specific flood, making it more difficult to generalize the results. Furthermore, the indication of the most important variables and the comparison of the models has to be carefully interpreted and should not be used blindly for the prediction of vulnerable areas to floods in case of a new disaster. It is important to be aware of this limitation and a first step for further research would therefore be to enhance the model with more (flood) events. This could be done only for Malawi, or possibly for other countries. Collecting more data about floods in Malawi may lead to a better model for Malawi. On the other hand, including floods from different countries would create a model of which its results and most important variables are more generalizable.

Quality Apart from enhancing the quantity of the data, the quality of the data can be enhanced too. For this research, the variables that were retrieved from the National Census provided by the NSO are on District level, while all other information is on TA level. A way to improve these data would be, to issue a request for these data to the NSO in Malawi. Enhancing the data this way, should lead to more accurate results. Furthermore, sources such as ReliefWeb considered the flood in 2015 to be a 1 in 500 years event and of huge proportions. This flood may therefore have different characteristics than smaller floods. Furthermore, this flood covered 79 TAs, so a smaller flood would mean an even smaller flood region and thus a smaller new data set to be added. The quality and level of detail of these data would therefore be of great importance, in order to be able to enhance the current data set.

Availability Another important factor is the availability of the data. The variable 'Drinking Source Nature', for example, that was indicated as one of the most important variables in this research, might not be available in every country. These data were collected as part of the 2010 National Census. However, other National Statistic Offices may use different ways to measure the quality of drinking sources or not collect these type of data at all. It is therefore of high importance to investigate the availability of the data on the variables used in this research for floods that took place at other times or in other countries. If some of these variables wouldn't exist, good proxy variables should be created, as well as more generalizable ways to collect data.

The aim of this research is to be able to give a good prediction of the most affected areas as fast as possible, preferably within the first 24-48 hours. This means that all the necessary data should be available within these first 24-28 hours, as well as the 'Flood Percentage' per TA, that was indicated to be one of the most important variables. As mentioned in Subsection 3.2, the satellite data give a good overview of the flood extent on April 13^{th} 2015, but not all data were available on the day itself. Often satellite imagery is published freely after a natural disaster took place from different companies. Although this type of data is only available during the disaster, the processes of analysing these data should already be put in place before a disaster takes place. The processing of these data could be done by including raw data available from a number of different sources. These images do often contain clouds however. Another option would be to work together with an expert analysing the flood extent or working together with an institute or university that would be able to do a quick analysis and deliver the correct data on the flood extent.

Completeness The availability of data is an important factor, but the absence of data might be almost as important as the availability. To pre-select useful data, this research uses the INFORM Index as a framework. These data are then analysed by the models and point out the most important variables. However, variables that are not part of the pre-selection are never considered and certain possibly important variables could therefore not have been incorporated. It is thus important to always have a complete framework, such as INFORM, combining this with different expert opinions to ensure the framework does cover all the important variables. Moreover, if some of these variables are not available, it is important to incorporate good proxies to make sure the data are complete.

Recency The last category considers the recency of the data. Certain data sets, such as the National Census, are only gathered every 5 or 10 years. Although this is not always a problem, it is advisable to look for proxy data that are updated more frequently. In this research, relative differences between TAs are considered, thus, as long as the relationship between the TAs does not change, this is not a problem. However, in reality, certain areas improve more on certain aspects (such as poverty ratio) than others, so considering different alternatives and proxies would be wise. Furthermore, it is important to carefully check the time stamps and make sure the data used is not the most recent data, but the most recent data before a flood took place.

Other Remarks At last, it is important to note that the aid described in this research was mainly focused on the Shelter Cluster. The response variable has been constructed using the Displacement Tracking Matrix and data from the Shelter Cluster, in order to be able to correctly categorize every TA according to its aid-neediness. A limitation of this scope used is that the results might be less relevant for other clusters and areas of help. For example, if the flood did not cause much damage to buildings, but it did to crops, not the Shelter Cluster would have been activated, but instead the

Food & Security Cluster. Although the clusters often shown an overlap (if houses collapse, there are probably also problems with water and sanitation), it is important to be aware of this scope and bias towards the Shelter Cluster.

Finally, possibly one of the most important factor for further research is its actual implementation. Different types of research have been and are being done in this field, but until they are used in the field, they are of little use. During the Field Trip to Malawi and different interviews among FACT trained people within the Netherlands Red Cross, one of the most important remarks was that the way of implementing and the type of study are very cluster specific, as mentioned above. The choice of the response variable is therefore of great importance. Another issue that was mentioned, is that often, there is the lack of basic knowledge and information on the ground. It would therefore be best to combine the forecast with information about factors such as poverty, the amount of children and women living in area etc. Furthermore, decisions made in the field rely on the resources on the ground and the stakeholders already in place. Therefore the information from this research should also be enhanced with the 3Ws (Where, What, Who). A possible example would be to give a first indication of the severity of the damage in an area and then to provide more detailed information, zoomed in per area. To ensure these data are efficiently used, they could become part of the FACT trainings provided by the Red Cross. In this case, first responders would be familiar with the types of information and would know how to use it correctly. The information should be sufficient, but should not provide an overload of information, which would lead to confusion and abolishment of the use of it. Furthermore, a disaster (and its aftermath) is a constantly changing situation, with organizations coming and going and the disaster, as well as possibly new ones, still developing. Therefore, it is of great importance to use all information, like the data gathered in this research, but simultaneously to be well informed about situational changes on the ground.

8 References

Archer, K. J. (2010). rpartordinal: an r package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, *34*, 7.

Benini, A., & Chataigner, P. (2014). Composite measures of local disaster impact-lessons from typhoon yolanda, philippines.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Cutter, S. L., Barnes, L., Berry, M., Burton, C., Evans, E., Tate, E., & Webb, J. (2008). A place-based model for understanding community resilience to natural disasters. *Global environmental change*, *18*(4), 598–606.

Darcy, J., & Hofmann, C. (2003). According to need?: needs assessment and decision-making in the humanitarian sector.

Galimberti, G., Soffritti, G., & Di Maso, M. (2012). Classification trees for ordinal responses in r: the rpartscore package. *Journal of Statistical Software*, *47*(1), 1–25.

Gao, J., Nickum, J. E., & Pan, Y. (2007). An assessment of flood hazard vulnerability in the dongting lake region of china. *Lakes & Reservoirs: Research & Management*, *12*(1), 27–34.

Hothorn, T., Hornik, K., Strobl, C., & Zeileis, A. (2013). Package party. [Computer software manual]. Retrieved from `http://cran.r-project.org/package=party`

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674.

Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. *Information systems management*, *29*(4), 258–268.

Jonkman, S. N. (2007). *Loss of life estimation in flood risk assessment; theory and applications* (Unpublished doctoral dissertation). TU Delft, Delft University of Technology.

Kohara, K., & Sugiyama, I. (2013). Typhoon damage scale forecasting with self-organizing maps trained by selective presentation learning. In *International workshop on machine learning and data mining in pattern recognition* (pp. 16–26).

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer.

Liaw, A., Wiener, M., Breiman, L., & Cutler, A. (2009). Package randomforest. *Retrieved December*, *12*, 2009.

Marc van den Homberg, M. S., Robert Monn. (2014). Bridging the information gap: Mapping data sets on information needs in the preparedness and response phase.

Molnar, C. (2013). *Recursive partitioning by conditional inference.* Munich.

Moltchanova, E., Khabarov, N., Obersteiner, M., Ehrlich, D., & Moula, M. (2011). The value of

rapid damage assessment for efficient earthquake response. *Safety science*, *49*(8), 1164–1171.

Mwale, F., Adeloye, A., & Beevers, L. (2015). Quantifying vulnerability of rural communities to flooding in ssa: a contemporary disaster management perspective applied to the lower shire valley, malawi. *International journal of disaster risk reduction*, *12*, 172–187.

Ravallion, M. (2010). *Mashup indices of development. world bank policy research working paper no. 5432.* Washington, DC: The World Bank.

Reliefweb, southern africa: Floods jan 2015. (n.d.). `http://reliefweb.int/disaster/fl-2015-000006-mwi`.

Rennó, C. D., Nobre, A. D., Cuartas, L. A., Soares, J. V., Hodnett, M. G., Tomasella, J., & Waterloo, M. J. (2008). Hand, a new terrain descriptor using srtm-dem: Mapping terra-firme rainforest environments in amazonia. *Remote Sensing of Environment*, *112*(9), 3469–3481.

Rudari, R., Beckers, J., De Angeli, S., Rossi, L., & Trasforini, E. (2016). Impact of modelling scale on probabilistic flood risk assessment: the malawi case. In *E3s web of conferences* (Vol. 7, p. 04015).

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Strasser, H., & Weber, C. (1999). On the asymptotic theory of permutation statistics.

Tehrany, M. S., Pradhan, B., & Jebur, M. N. (2013). Spatial prediction of flood susceptible areas using rule based decision tree (dt) and a novel ensemble bivariate and multivariate statistical models in gis. *Journal of Hydrology*, *504*, 69–79.

UNISDR, C. (2015). The human cost of natural disasters: A global perspective.

Wallis, C., Watson, D., Tarboton, D., & Wallace, R. (2009). Parallel flow-direction and contributing area calculation for hydrology analysis in digital elevation models. In *Proceedings of the international conference on parallel and distributed processing techniques and applications.* Las Vegas, Nevada, USA.

Wang, Z., Lai, C., Chen, X., Yang, B., Zhao, S., & Bai, X. (2015). Flood hazard risk assessment model based on random forest. *Journal of Hydrology*, *527*, 1130–1141.

Appendix A - R Code

Thomas Plaatsman

```r
#READ ADMINISTRATIVE BOUNDARIES
#THESE BOUNDARIES ARE FROM:
#https://data.humdata.org/dataset/malawi-admin-level-3-boundaries
admin <- read.csv('adminbound3.csv', sep = ';')
admin$OBJECTID <- NULL
colnames(admin)[colnames(admin) == 'P_CODE'] <- 'P_CODE_TA'
admin$P_CODE_DISTRICT <- substring(admin$P_CODE_TA, first = 0, last = 9)
admin$P_CODE_REGION <- substring(admin$P_CODE_TA, first = 0, last = 7)
admin <- admin[, c(1:5,8,9)]
```

```r
#--------------------------------------READ X-DATA--------------------------------------
#Data from http://malawi.opendataforafrica.org/ and Knoema
communication <- read.csv("xvariables/malawidataportal/pcoded_infocommun.csv", sep=",")
demographic <- read.csv("xvariables/malawidataportal/pcoded_demographic.csv", sep=",")
economic <- read.csv("xvariables/malawidataportal/pcoded_economic.csv", sep=",")
environment <- read.csv("xvariables/malawidataportal/pcoded_environment.csv", sep=",")
health <- read.csv("xvariables/malawidataportal/pcoded_health.csv", sep=",")
houses <- read.csv("xvariables/malawidataportal/pcoded_houses.csv", sep=",")

#SELECT IMPORTANT INDICATORS
communication_var <- c('Proportion of households with access to Landline telephone',
                       'Proportion of households with access to mobile phone')

demographic_var <- c('Average household size', 'Crude birth rate', 'Dependency ratio',
                     'Households', 'Life expectancy at birth', 'Population density')

economic_var <- c('Mean per capita income', 'Poverty headcount ratio', 'Poverty status')

environment_var <- c('Amount of rainfall', 'Land and water area',
                     'Ownership of toilet facility',
                     'Population without access to safe water',
                     'Proportion of household population taking 15 minutes to
                     less than 30 minutes to source of drinking water',
                     'Proportion of household population taking less than 15 minutes
                     to source of drinking water',
                     'Proportion of households with source of drinking water',
                     'Proportion of population with sustainable access to an
                     improved water source',
                     'Proportion of population without sustainable access to an
                     improved water source',
                     'Proportion with access to improved sanitation',
                     'Source of drinking water', 'Water point density',
                     'Water points')

health_var <- c('Child mortality rate', 'Doctor-population ratio',
                'Hospital bed utilisation rate (average length of stay)',
                'Hospital bed utilisation rate (turnover)', 'Infant mortality rate',
                'Neonatal mortality rate', 'Nurse-population ratio')
```

```
houses_var <- c('Population distribution by type of dwelling structure',
                'Type of construction materials', 'Type Of Housing Tenure')
```

```
#SELECT LAST YEAR PER INDICATOR
communication_year <- NULL
for (i in 1:length(communication_var)) {
  communication_year[[i]] <- max(
    communication$Date[communication$indicator == communication_var[[i]]])
}

demographic_year <- NULL
for (i in 1:length(demographic_var)) {
  demographic_year[[i]] <- max(
    demographic$Date[demographic$indicator == demographic_var[[i]]])
}

economic_year <- NULL
for (i in 1:length(economic_var)) {
  economic_year[[i]] <- max(
    economic$Date[economic$indicator == economic_var[[i]]])
}

environment_year <- NULL
for (i in 1:length(environment_var)) {
  environment_year[[i]] <- max(
    environment$Date[environment$indicator == environment_var[[i]]])
}

health_year <- NULL
for (i in 1:length(health_var)) {
  health_year[[i]] <- max(health$Date[health$indicator == health_var[[i]]])
}

houses_year <- NULL
for (i in 1:length(houses_var)) {
  houses_year[[i]] <- max(
    houses$Date[houses$indicator == houses_var[[i]]])
}
```

```
#SUBSET PER YEAR AND INDICATOR SELECTION
communication <- rbind(subset(communication,
                              communication$indicator == communication_var[[1]] &
                                communication$Date == communication_year[[1]]),
                       subset(communication,
                              communication$indicator == communication_var[[2]] &
                                communication$Date == communication_year[[2]]))

health <- rbind(subset(health, health$indicator == health_var[[1]] &
                         health$Date == health_year[[1]]),
                subset(health, health$indicator == health_var[[2]] &
                         health$Date == health_year[[2]]),
                subset(health, health$indicator == health_var[[3]] &
                         health$Date == health_year[[3]]),
                subset(health, health$indicator == health_var[[4]] &
```

```
                        health$Date == health_year[[4]]),
                subset(health, health$indicator == health_var[[5]] &
                        health$Date == health_year[[5]]),
                subset(health, health$indicator == health_var[[6]] &
                        health$Date == health_year[[6]]),
                subset(health, health$indicator == health_var[[7]] &
                        health$Date == health_year[[7]]))

demographic <- rbind(subset(demographic, demographic$indicator == demographic_var[[1]] &
                            demographic$Date == demographic_year[[1]]),
                     subset(demographic, demographic$indicator == demographic_var[[2]] &
                            demographic$Date == demographic_year[[2]]),
                     subset(demographic, demographic$indicator == demographic_var[[3]] &
                            demographic$Date == demographic_year[[3]]),
                     subset(demographic, demographic$indicator == demographic_var[[4]] &
                            demographic$Date == demographic_year[[4]]),
                     subset(demographic, demographic$indicator == demographic_var[[5]] &
                            demographic$Date == demographic_year[[5]]),
                     subset(demographic, demographic$indicator == demographic_var[[6]] &
                            demographic$Date == demographic_year[[6]]))

economic <- rbind(subset(economic, economic$indicator == economic_var[[1]] &
                           economic$Date == economic_year[[1]]),
                  subset(economic, economic$indicator == economic_var[[2]] &
                           economic$Date == economic_year[[2]]),
                  subset(economic, economic$indicator == economic_var[[3]] &
                           economic$Date == economic_year[[3]]))

environment <- rbind(subset(environment, environment$indicator == environment_var[[1]] &
                              environment$Date == environment_year[[1]]),
                     subset(environment, environment$indicator == environment_var[[2]] &
                              environment$Date == environment_year[[2]]),
                     subset(environment, environment$indicator == environment_var[[3]] &
                              environment$Date == environment_year[[3]]),
                     subset(environment, environment$indicator == environment_var[[4]] &
                              environment$Date == environment_year[[4]]),
                     subset(environment, environment$indicator == environment_var[[5]] &
                              environment$Date == environment_year[[5]]),
                     subset(environment, environment$indicator == environment_var[[6]] &
                              environment$Date == environment_year[[6]]),
                     subset(environment, environment$indicator == environment_var[[7]] &
                              environment$Date == environment_year[[7]]),
                     subset(environment, environment$indicator == environment_var[[8]] &
                              environment$Date == environment_year[[8]]),
                     subset(environment, environment$indicator == environment_var[[9]] &
                              environment$Date == environment_year[[9]]),
                     subset(environment, environment$indicator == environment_var[[10]] &
                              environment$Date == environment_year[[10]]),
                     subset(environment, environment$indicator == environment_var[[11]] &
                              environment$Date == environment_year[[11]]),
                     subset(environment, environment$indicator == environment_var[[12]] &
                              environment$Date == environment_year[[12]]),
                     subset(environment, environment$indicator == environment_var[[13]] &
```

```
                              environment$Date == environment_year[[13]]))

health <- rbind(subset(health, health$indicator == health_var[[1]] &
                         health$Date == health_year[[1]]),
                subset(health, health$indicator == health_var[[2]] &
                         health$Date == health_year[[2]]),
                subset(health, health$indicator == health_var[[3]] &
                         health$Date == health_year[[3]]),
                subset(health, health$indicator == health_var[[4]] &
                         health$Date == health_year[[4]]),
                subset(health, health$indicator == health_var[[5]] &
                         health$Date == health_year[[5]]),
                subset(health, health$indicator == health_var[[6]] &
                         health$Date == health_year[[6]]),
                subset(health, health$indicator == health_var[[7]] &
                         health$Date == health_year[[7]]))

houses <- rbind(subset(houses, houses$indicator == houses_var[[1]] &
                         houses$Date == houses_year[[1]]),
                subset(houses, houses$indicator == houses_var[[2]] &
                         houses$Date == houses_year[[2]]),
                subset(houses, houses$indicator == houses_var[[3]] &
                         houses$Date == houses_year[[3]]))
```

```
#SELECT ONLY DISTRICT LEVEL
communication <- communication[nchar(as.character(communication$code)) == 9 ,]
demographic <- demographic[nchar(as.character(demographic$code)) == 9 ,]
economic <- economic[nchar(as.character(economic$code)) == 9 ,]
environment <- environment[nchar(as.character(environment$code)) == 9 ,]
health <- health[nchar(as.character(health$code)) == 9 ,]
houses <- houses[nchar(as.character(houses$code)) == 9 ,]

#CREATE RIGHT FORMAT
reshape_data <- function(dataset) {
  dataset$indicator <- apply( dataset[ , c('indicator', 'subgroup') ] , 1 ,
                              paste , collapse = "_" ) #combine subgroup & indicator
  output <- reshape(dataset[, c(3,9,11)],
                    timevar = "indicator",

                    idvar = "code",
                    direction = "wide")
}

communication_form <- reshape_data(communication)
demographic_form <- reshape_data(demographic)
economic_form <- reshape_data(economic)
health_form <- reshape_data(health)
houses_form <- reshape_data(houses)

#environment has one variable, both in numbers and percentages, so extra combination
environment$indicator <- apply( environment[ , c('indicator', 'subgroup', 'measure') ],
                                1, paste , collapse = "_"
) #combine subgroup & indicator & measure (number, %)
```

```
environment_form <- reshape(environment[, c(3,9,11)],
                            timevar = "indicator",
                            idvar = "code",
                            direction = "wide")
```

```
#MERGE DATA
datax <- merge(admin, communication_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
datax <- merge(datax, demographic_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
datax <- merge(datax, economic_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
datax <- merge(datax, environment_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
datax <- merge(datax, health_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
datax <- merge(datax, houses_form,
               by.x = 'P_CODE_DISTRICT', by.y = 'code', all.x = T)
```

```
#FINAL SELECTION
datax <- datax[, c(1:7,14,15,35:40,43,47)]
colnames(datax)[8:17] <- c('life expect at birth female DISTR',
                           'life expect at birth male DISTR',
                           'proportion with access to improved sanitation DISTR',
                           'drink_source_nature_distr',
                           'drink_source_piped_dwelling_distr',
                           'drink_source_piped_communal_distr',
                           'drink_source_protected_well_distr',
                           'drink_source_open_well_distr',
                           'child mortality rate 1-4yo DISTR',
                           'infant mortality rate <1yo DISTR')

#CREATE RIGHT COLUMN NAMES FOR MLW DATA PORTAL, DELETE BACKTICK
datax$life_expect_F <- datax$`life expect at birth female DISTR`
datax$`life expect at birth female DISTR` <- NULL

datax$life_expect_M <- datax$`life expect at birth male DISTR`
datax$`life expect at birth male DISTR` <- NULL

datax$prop_access_improv_sanitation <-
  datax$`proportion with access to improved sanitation DISTR`
datax$`proportion with access to improved sanitation DISTR` <- NULL

datax$child_mortality_rate <- datax$`child mortality rate 1-4yo DISTR`
datax$`child mortality rate 1-4yo DISTR` <- NULL

datax$infant_mortality_rate <- datax$`infant mortality rate <1yo DISTR`
datax$`infant mortality rate <1yo DISTR` <- NULL

#Combine Life Expectancy Male/Female
datax$life_expectancy <- (datax$life_expect_F + datax$life_expect_M) / 2
datax$life_expect_F <- NULL
datax$life_expect_M <- NULL
```

```r
#Source dwelling
datax$drink_source_nature <- datax$drink_source_nature_distr
datax$drink_source_piped_dwelling <- datax$drink_source_piped_dwelling_distr
datax$drink_source_piped_communal <- datax$drink_source_piped_communal_distr
datax$drink_source_protected_well <- datax$drink_source_protected_well_distr
datax$drink_source_open_well <- datax$drink_source_open_well_distr

datax$drink_source_nature_distr <- NULL
datax$drink_source_piped_dwelling_distr<- NULL
datax$drink_source_piped_communal_distr<- NULL
datax$drink_source_protected_well_distr<- NULL
datax$drink_source_open_well_distr<- NULL
```

```r
#-----------------IHS11-------------
#NSO WorldBank IHS 2011 Data
#After reshaping data, the data need to be PCoded
#This is why currently this part is commented

# library(foreign) #for .dta files
#
# #2010 SURVEY
# wd11 <- 'xvariables/IHS1011/Household/'
# HH_MOD_A <- read.dta(paste0(wd11, 'HH_MOD_A_FILT.dta'))
# HH_MOD_F <- read.dta(paste0(wd11, 'HH_MOD_F.dta'))
# HH_MOD_H <- read.dta(paste0(wd11, 'HH_MOD_H.dta'))
# HH_MOD_T <- read.dta(paste0(wd11, 'HH_MOD_T.dta'))
#
# #SELECT RELEVANT VARIABLES
# HH_MOD_A <- HH_MOD_A[, c(1,2,6,8)]
# HH_MOD_F <- HH_MOD_F[, c(1,11,12,14,16,47,51,53,56:61,63,67)]
# HH_MOD_H <- HH_MOD_H[, c(1,4,12)]
# HH_MOD_T <- HH_MOD_T[, c(1,8:10)]
# #HH_MOD_U <- HH_MOD_U[, c(1,3,6,7)]
#
# #MERGE FILES
# ihs_11 <- merge(HH_MOD_A, HH_MOD_F, by = 'case_id')
# ihs_11 <- merge(ihs_11, HH_MOD_H, by = 'case_id')
# ihs_11 <- merge(ihs_11, HH_MOD_T, by = 'case_id')
#
# #CHANGE NAMES
# survey_names <- as.data.frame(colnames(ihs_11))
# colnames(ihs_11) <- c('case_id', 'ea_id', 'district', 'ta', 'dwelling type',
#                       'wall type', 'roof type', 'floor type', 'main telephone line',
#                       'no of cellphones', 'drinking water source',
#                       'walking distance main water source', '
#                       walking distance main water source min/hour', 'Use water source',
#                       'main drinking source rest of season', 'toilet facility type',
#                       'toilet facility use','rubbish disposal',
#                       'hh contains children below 5yo','food worries last 7 days',
#                       'food shortage last 12months', 'poverty guess HH',
#                       'poverty guess neighbours', 'poverty guess friends')
#
# compared_names <- cbind(survey_names, as.data.frame(colnames(ihs_11)))
#
```

```
#-----------------IHS11 END-------------

#READ PCODED IHS11 FILE INSTEAD
ihs_11 <- read.csv('xvariables/IHS1011/ihs11_pcoded.csv', sep = ';')

#DELETE VARIABLES THAT WE WON'T USE
ihs_11[, c(20:29)] <- NULL
ihs_11$main.telephone.line <- NULL
ihs_11$Use.water.source <- NULL
ihs_11$main.drinking.source.rest.of.season <- NULL
ihs_11$toilet.facility.use <- NULL
ihs_11$hh.contains.children.below.5yo <- NULL
ihs_11$dwelling.type <- NULL
ihs_11$floor.type <- NULL
colnames(ihs_11)[[13]] <- "P_CODE_TA"

#RESHAPE THE WALL TYPES
wall <- ihs_11[, c(1:4, 13, 5)]

library(reshape2)
#reshape to columns per case
column_table <- dcast(wall, case_id ~ wall.type, fun.aggregate = length)
columns_ta <- merge(wall[, c("case_id", "P_CODE_TA")], column_table, by = 'case_id')
columns_ta$total <- 1 #create ones, so you can count total
columns_ta$`Other (specify)` <- NULL #Because it hasn't been specified

#aggregate on TA level
library(dplyr)
agg_columns <- as.data.frame(columns_ta[, c(2:11)] %>% group_by(P_CODE_TA)
                             %>% summarise_each(funs(sum)))

#Calculate Percentage brick & concrete wall types
agg_columns$brick_concrete_wall <- agg_columns$`Burnt bricks` + agg_columns$Concrete
agg_columns$perc_walltypes <- agg_columns$brick_concrete_wall/agg_columns$total
```

```
#Mobile Phone Use IHS11
mobile_tel <- as.data.frame(ihs_11[, c(13, 7)] %>% group_by(P_CODE_TA)
                            %>% summarise_each(funs(sum)))

#FLOOD MAPS
flood_area <- read.csv('xvariables/Flood/floodarea_per_admint.csv', sep = ';')
flood_area <- flood_area[, c(2,12)]

#Read hand etc - Aki
flood <- read.csv('xvariables/Flood/flood.csv', sep = ';')
flood <- flood[, c(1,4:8)]

#World Poverty Index:
#www.worldpop.org.uk/data/summary/?contselect=Africa&countselect=Malawi&typeselect=Poverty
poverty <- read.csv('xvariables/worldpop/poverty_qgis.csv', sep = ';')

#Health Facilities:
#http://www.masdap.mw/layers/geonode:mw_health_wgs84
health_facilities <- read.csv('xvariables/masdap/health_facilities.csv', sep = ';')
```

```
#DATA OSM
roads <- read.csv('xvariables/osm/roads.csv', sep = ';')

#ADD ALL XDATA
datax <- merge(datax, flood, by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)
datax <- merge(datax, flood_area, by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)
datax <- merge(datax, poverty[, c(1,4)], by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)
datax <- merge(datax, health_facilities, by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)
datax <- merge(datax, roads[, c(1,6,7)], by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)
#datax <- merge(datax, roads[, c(1,6,7)], by.x = 'P_CODE_TA', by.y='P_CODE', all.x = T)

#IHS11 DATA
datax <- merge(datax, agg_columns[, c(1,12)], by = 'P_CODE_TA', all.x = T)
datax <- merge(datax, mobile_tel, by = 'P_CODE_TA', all.x = T)
```

```
#------YDATA------
y_var <- read.csv('yvariables/workingfolder/yvar_cat.csv', sep = ';')

#CREATE FIRST SUBSET
data_sample <- merge(datax, y_var, all.y = T)
data_sample$P_CODE_REGION <- NULL

#NUMBER OF VARIABLES
no_of_var <- dim(data_sample)[[2]]

#DELETE IRRELEVANT AREAS SUCH AS LAKES AND FORESTS & POPULATION = NULL
data_sample <- data_sample[!data_sample$POP2008 == 'NULL' ,]
data_sample <- data_sample[!(data_sample$TRAD_AUTH == 'Lengwe National Park' |
                               data_sample$TRAD_AUTH == 'Mwabvi Game Reserve'),]
```

```
#-------MODELS--------
#CHECK CLASS & SET TO INTEGER
#This is done in 2 steps, otherwise the numbers of population for example change
classes <- sapply(data_sample[, c(6:no_of_var)], class)
data_sample[, c(6:no_of_var)] <- sapply(data_sample[, c(6:no_of_var)], as.character)
data_sample[, c(6:no_of_var)] <- sapply(data_sample[, c(6:no_of_var)], as.numeric)
classes <- sapply(data_sample, class) #Check if classes were changed correctly

#Deal with missing values - median imput per column or for factor mode (most freq)
library(randomForest)
data_sample[, c(6:no_of_var)] <- na.roughfix(data_sample[, c(6:no_of_var)])
# head(data_sample)
```

```
#-----CART-----
library(rpartScore) #misclassification cost is default
set.seed(123)
cart_full <- rpartScore(ygroup ~ ., data = data_sample[, c(6:no_of_var)],
                        minsplit = 10) #adding cp=0 gives the same tree
# cart_full$cptable
cart_default <- rpartScore(ygroup ~ ., data = data_sample[, c(6:no_of_var)])

#PRUNE THE TREE USING THE SE-1 RULE
cart_default$cptable
pfit<- prune(cart_full, cp=0.02) #pruned tree, min xerror
```

```r
#PLOT TREE
# Manual at: http://www.milbo.org/rpart-plot/prp.pdf
library(rpart.plot)
rpart.plot(cart_default, uniform=TRUE, branch.lty=2, shadow.col="gray",
           nn=TRUE, main="CART Default")
```

```r
rpart.plot(pfit, uniform=TRUE, branch.lty=2, shadow.col="gray",
           nn=TRUE, main="Pruned Tree")
```

```r
#PREDICTION
tree_prediction <- function(tree){
  predict = predict(tree)
  results <- as.data.frame(cbind(actual = as.numeric(data_sample$ygroup),
                                 predict = predict))
  correct <-  sum(results$actual==results$predict)/dim(results)[[1]]
  output <- list(correct=correct, confusionmatrix=table(actual = data_sample$ygroup,
                                                        predict))
  return(output)
}

#tree_prediction(cart_full)
tree_prediction(cart_default)
tree_prediction(pfit)
```

```r
#library(party)
library(partykit)
#https://www.r-bloggers.com/a-brief-tour-of-the-trees-and-forests/
#http://www.exegetic.biz/blog/2013/05/package-party-conditional-inference-trees/

data_sample$ygroup <- as.factor(data_sample$ygroup)
CT_default <- ctree(ygroup ~ ., data = data_sample[, c(6:no_of_var)])
CT_30 <- ctree(ygroup ~ ., data = data_sample[, c(6:no_of_var)],
               control = ctree_control(mincriterion = 0.70))

#plot
plot(CT_default)
```

```r
plot(CT_30)
```

```r
#Results & Confusion Matrix
tree_prediction(CT_default)
tree_prediction(CT_30)

#RF has difficulties with col names, so change here
colnames(data_sample)[6:27] <- c("Population", "Access.Improved.Water.Sanitation",
                                 "Child.Mortality.Rate", "Infant.Mortality.Rate",
                                 "Life.Expectancy", "Drinking.Source.Nature",
                                 "Drinking.Source.Piped.Dwelling",
                                 "Drinking.Source.Piped.Communal",
                                 "Drinking.Source.Protected.Well",
                                 "Drinking.Source.Open.Well",
                                 "Contributing.Area", "Stream.Order", "HAND",
                                 "Slope", "Accumulated.Rainfall", "Flood.Percentage",
                                 "Poverty", "Number.of.Health.Institutions",
```

```
                                "Road.Density", "Road.Length",
                                "Percentage.Good.Wall.Type", "Number.of.Cellphones")
```

```
#-------RF-CART------------
library(randomForest)
data_sample$ygroup <- as.factor(data_sample$ygroup)

#Create a vector with OOB error rate per mtry
rf_oob <- as.data.frame(matrix(nrow = no_of_var-6, ncol = 1))
for(i in 1:(no_of_var-6)){
  set.seed(123)
  output.forest <- randomForest(ygroup ~ .,data = data_sample[, c(6:no_of_var)], mtry= i)
  rf_oob[i ,] <- as.data.frame(output.forest$err.rate[500,1])
}

rf_mtry <- 21

#Create RF with best mtry
set.seed(123)
output.forest <- randomForest(ygroup ~ .,data = data_sample[, c(6:no_of_var)],
                              importance=T, mtry= rf_mtry)
output.forest

#Variable Importance
importance(output.forest)
varImpPlot(output.forest,type=2) #mean decrease in node impurity
```

```
varImpPlot(output.forest,type=1) #mean decrease in accuracy
```

```
#-------RF-CTREE------------
detach("package:partykit", unload=TRUE)
library(party)
data_sample$ygroup <- as.factor(data_sample$ygroup)

#Function prediction for ctree
cprediction <- function(tree){
  cpredict <- predict(tree, OOB=TRUE)
  results <- as.data.frame(cbind(as.numeric(as.character(data_sample$ygroup)),
                                 as.numeric(as.character(cpredict))))
  correct <-  sum(results$V1==results$V2)/dim(results)[[1]]
  output <- list(correct=correct, confusionmatrix=table(actual = data_sample$ygroup,
                                                        predict = cpredict))
  return(output)
}

#Create a vector with OOB error rate per mtry
crf_oob <- as.data.frame(matrix(nrow = no_of_var-6, ncol = 1))
for(i in 1:(no_of_var-6)){
  set.seed(123)
  data.controls <- cforest_unbiased(mtry=i)
  condit_forest <- cforest(ygroup ~ .,data = data_sample[, c(6:no_of_var)],
                           control = data.controls)
  crf_oob[i ,] <- 1-cprediction(condit_forest)[[1]]
}
```

```
crf_mtry <- which.min(crf_oob$V1) #Best mtry, lowest OOB Error rate

#PLOT OOB error rate & show best mtry
plot(crf_oob$V1, col=ifelse(index(crf_oob$V1) == 6, "red", "black"),
     xlab = 'Value of mtry', ylab = 'OOB Error Rate', type = 'b')
```

```
set.seed(123)
data.controls <- cforest_unbiased(mtry = crf_mtry)
condit_forest <-cforest(ygroup ~ ., data = data_sample[, c(6:no_of_var)],
                        control = data.controls)
cprediction(condit_forest)

#mean decrease in accuracy, with permutation importance
varimp_cforest <- as.data.frame(varimp(condit_forest))

#Plot results cforest
cforestImpPlot <- function(x) {
  cforest_importance <<- v <- varimp(x)
  dotchart(v[order(v)])
}

cforestImpPlot(condit_forest)
```

```
#-----CROSS VALIDATED RESULTS-----
library(plyr) #for progress bar

#REPEATED CROSS VALIDATION FUNTION
repeated_cv <- function(no_of_repeats, kfold, model_no){

  #REPEATED CROSS VALIDATION
  repeatedcv_correct <- data.frame()
  prec0 <- data.frame()
  recall0 <- data.frame()
  prec1 <- data.frame()
  recall1 <- data.frame()
  prec2 <- data.frame()
  recall2 <- data.frame()
  acc <- data.frame()

  #Creating a progress bar to know the status of CV
  progress.bar <- create_progress_bar("text")
  progress.bar$init(no_of_repeats)

  acc_confmatrix <- matrix(0, nrow = 3, ncol = 3)
  single_confmatrix <- matrix(0, nrow = 3, ncol = 3)

  ############START REPEATED CROSS VALIDATION##########
  for(j in 1:no_of_repeats){

    #------CROSS VALIDATION------
    #TP: Fill up missing values
    data_sample[, c(6:no_of_var)] <- na.roughfix(data_sample[, c(6:no_of_var)])
```

```
library(plyr)
data <- data_sample[, c(6:no_of_var)]

# sample from 1 to k, nrow times (the number of observations in the data)
set.seed(j)
data$id <- sample(1:kfold, nrow(data), replace = TRUE)
list <- 1:kfold

# prediction and testset data frames that we add to with each iteration over
# the folds
prediction <- data.frame()
testsetCopy <- data.frame()

#----------START KFOLD CROSS VALIDATION----------
for (i in 1:kfold){
  # remove rows with id i from dataframe to create training set
  # select rows with id i to create test set
  trainingset <- subset(data, id %in% list[-i])
  testset <- subset(data, id %in% c(i))

  #IF clause for parameters RF or Dec Tree
  set.seed(123)

  if (model_no == 1){
    mymodel <- rpartScore(as.numeric(as.character(ygroup)) ~ ., data = trainingset)
  } else if (model_no == 2){
    mymodel <- ctree(as.factor(ygroup) ~ ., data = trainingset)
  } else if (model_no == 3){
    mymodel <- randomForest(as.factor(trainingset$ygroup) ~ ., data = trainingset,
                            mtry = rf_mtry)
  } else if (model_no == 4){
    data.controls <- cforest_unbiased(mtry = crf_mtry)
    mymodel <-cforest(ygroup ~ ., data = trainingset, control = data.controls)
  }

  # remove response column ygroup
  temp <- as.data.frame(predict(mymodel, newdata = testset[,-23]))
  # append this iteration's predictions to the end of the prediction data frame
  prediction <- rbind(prediction, temp)

  #The testset is to keep the ygroup, so we can do a rbind.
  testsetCopy <- rbind(testsetCopy, as.data.frame(testset[,23]))

}
#----------END KFOLD CROSS VALIDATION----------


# add predictions and actual ygroup values
result <- cbind(testsetCopy[, 1], prediction)
names(result) <- c("Actual", "Predicted")
acc_confmatrix <- acc_confmatrix + table(result)
single_confmatrix <- table(result)
```

```
  correct <-  sum(result$Actual==result$Predicted)/dim(result)[[1]]
  correct

  repeatedcv_correct[j,1] <- correct

  #CREATE MICRO AVERAGES PRECISION & RECALL
  tp0 <- single_confmatrix[1,1]
  fp0 <- single_confmatrix[2,1] + single_confmatrix[3,1]
  fn0 <- single_confmatrix[1,2] + single_confmatrix[1,3]
  tp1 <- single_confmatrix[2,2]
  fp1 <- single_confmatrix[1,2] + single_confmatrix[3,2]
  fn1 <- single_confmatrix[2,1] + single_confmatrix[2,3]
  tp2 <- single_confmatrix[3,3]
  fp2 <- single_confmatrix[1,3] + single_confmatrix[2,3]
  fn2 <- single_confmatrix[3,1] + single_confmatrix[3,2]

  prec0[j,1] <- tp0/(tp0+fp0)
  recall0[j,1] <- tp0/(tp0+fn0)
  prec1[j,1] <- tp1/(tp1+fp1)
  recall1[j,1] <- tp1/(tp1+fn1)
  prec2[j,1] <- tp2/(tp2+fp2)
  recall2[j,1] <- tp2/(tp2+fn2)

  #Set missing values to 0 (when category 1 isn't predicted, for example with ctree)
  prec1[j,1][is.na(prec1[j,1])] <- 0

  #accuracy
  acc[j,1] <- (tp0 + tp1 + tp2)/sum(single_confmatrix)

  progress.bar$step()
}
############END REPEATED CROSS VALIDATION###########

#CREATE OUTCOME TABLE WITH MICRO & MACRO PRECISION & RECALL
outcome_table <- data.frame()
#Micro precision
outcome_table[1,1] <- mean(prec0[,1])
outcome_table[2,1] <- mean(prec1[,1])
outcome_table[3,1] <- mean(prec2[,1])
#Micro recall
outcome_table[1,2] <- mean(recall0[,1])
outcome_table[2,2] <- mean(recall1[,1])
outcome_table[3,2] <- mean(recall2[,1])
#Macro Precision
outcome_table[1,3] <- acc_confmatrix[1,1]/sum(acc_confmatrix[,1])
outcome_table[2,3] <- acc_confmatrix[2,2]/sum(acc_confmatrix[,2])
outcome_table[3,3] <- acc_confmatrix[3,3]/sum(acc_confmatrix[,3])
#Macro Recall
outcome_table[1,4] <- acc_confmatrix[1,1]/sum(acc_confmatrix[1,])
outcome_table[2,4] <- acc_confmatrix[2,2]/sum(acc_confmatrix[2,])
outcome_table[3,4] <- acc_confmatrix[3,3]/sum(acc_confmatrix[3,])

rownames(outcome_table) <- c("0", "1", "2")
```

```
  colnames(outcome_table) <- c("micro precision", "micro recall", "macro precision",
                               "macro recall")

  correct_rcv <- mean(repeatedcv_correct[,1])
  output <- list(Run_accuracy = acc, prec_recall_table = outcome_table,
                 Accuracy = correct_rcv, conf = acc_confmatrix)
  return(output)
}
```

```
#REPEATED CROSS VALIDATED OUTPUTS
load("output_CART.RData")
load("output_CTREE.RData")
load("output_RF.RData")
load("output_CFOREST.RData")

# output_CART <- model_output <- repeated_cv(no_of_repeats = 100,
# kfold = 5, model_no = 1)
# output_CTREE <- model_output <- repeated_cv(no_of_repeats = 100,
# kfold = 5, model_no = 2)
# output_RF <- model_output <- repeated_cv(no_of_repeats = 100,
# kfold = 5, model_no = 3)
# output_CFOREST <- model_output <- repeated_cv(no_of_repeats = 100,
# kfold = 5, model_no = 4)
#
# save(output_CART, file="output_CART.RData")
# save(output_CTREE, file="output_CTREE.RData")
# save(output_RF, file="output_RF.RData")
# save(output_CFOREST, file="output_CFOREST.RData")

#ACCURACY VARIANCE PER MODEL
var(output_CART$Run_accuracy)
var(output_CTREE$Run_accuracy)
var(output_RF$Run_accuracy)
var(output_CFOREST$Run_accuracy)
```