MACHINE LEARNING METHODOLOGIES TO ASSES DEBRIS EXTENT AFTER EARTHQUAKE

Supervisor:
Prof. Gian Paolo Cimellaro

UCLA Supervisor:
Prof. Henry Burton

Dissertation of:
Luca Rampini 223989

Academic Year 2017/2018
Alla mia famiglia
ACKNOWLEDGEMENTS

First and foremost, I would like to thank Professor Gian Paolo Cimellaro who gave me the chance to write my master thesis and the opportunity to have this remarkable work experience in a prestigious and important university such as UCLA. I will always be grateful to him also for the advices and the knowledge that directly derives from his collaboration.

I am deeply indebted to Professor Burton to give me a great chance of learning, to guide me in all the process necessary to finish my work and to host me at UCLA; I really appreciate his competence, kindness and availability for all the period that I have been in Los Angeles.

I would also like to extend my thanks to Professor Taciroglu, to give me the possibility to work with his research group, that turned out a continuous source of knowledge.

A huge thank you goes to Antonio and Matteo, who have helped me a lot to improve my knowledge about Machine Learning and the use of Python in general; and made my experience in Los Angeles crisper.

Tutta la mia gratitudine è rivolta ai miei genitori, Brunella e Roberto, per avermi dato non solo la possibilità di perseguire i miei studi al Politecnico, ma anche questa preziosa opportunità di studio all’estero senza far mai mancare il loro supporto.

Un ringraziamento speciale va a mio fratello Giacomo, sempre pronto a dispensare preziosi consigli e che ha sempre rappresentato per me un modello da seguire.

Voglio estendere i miei ringraziamenti anche ai miei amici di sempre, Mirko, Silvio, Nicola, Simone, Andrea, Antonio, Matteo, Nicodemo, Fabio, Paride, Emanuele, Giacomo, Niko e Gabriele; i quali mi sono stati vicini, nonostante la distanza, e che mi hanno reso in larga parte la persona che sono oggi.

Un sentito ringraziamento ad Alessandro, Mattia, Stefano e Daniele, per avermi fatto passare gli ultimi mesi a Torino con una sintonia tale da credere che fossimo sempre vissuti insieme.

Un sentito grazie va inoltre a Miriam, Claudia, Martina, Aurelio, Ludovico, Pierpaolo, Fabio, Desideria, Giulio, Francesca, Vittoria, Simone, Livio, Arturo, Enrico e a tutti gli altri ragazzi che ho avuto la fortuna e l’onore di conoscere in
questi anni a Torino, per avermi accompagnato in questo viaggio rendendolo sicuramente più piacevole ed elettrizzante. Un caloroso grazie lo devo anche ad Alessio, per avermi sopportato per ben due anni come coinquilino e per essersi rivelato un amico sempre pronto ad aiutare in ogni momento. Un grazie speciale anche a Mattia R. per aver sempre creduto in me e per la sua amicizia che, sono sicuro, non verrà mai a mancare.
Each significant seismic event is an opportunity to assess the performance of our built environment. After such events, a large quantity of varied information is often collected within a short time period.

Therefore, a quick and reliable method for detecting damage to buildings is required for disaster response with the intent of reducing both human and economic losses.

In the last few years, the increasing advancement of technologies and Computer Science has permitted the adoption of Machine Learning techniques in various fields, included Civil Engineering and Urban Resilience.

One of the most influence parameters in the evaluation of seismic loss is the extension of the debris and their associated effects on critical infrastructure. Debris accumulation can result in disruption of the road network and compromise rescue operations. This implies an overall increase in the average number of people who have difficulty evacuating, with a significant risk that some residents cannot evacuate at all.

Starting from the newest Machine Learning algorithms and using Python programming language, the purpose of this study is to collect different features and data from the main seismic events that occurred in the past several decades and use them to forecast the extension of building-rubble after earthquakes. This type of model would ultimately allow civil protection agencies to plan their rescue operations while reducing the risk of getting stuck by extended rubble.

Finally, the PHI Challenge sponsored by PEER, gives a chance to introduce Deep Learning methodologies for Image Recognition tasks related to Earthquake Reconnaissance. This topic indeed, is one of the most popular argument in the state of the art that connects Machine Learning techniques and Civil Engineering.
Ogni evento sismico di una certa rilevanza rappresenta una opportunità per valutare le prestazioni dei nostri edifici e delle nostre infrastrutture. Infatti, dopo tali eventi, una grande quantità di informazioni è raccolta entro breve tempo.

Pertanto, una rapida ed affidabile metodologia è necessaria per valutare velocemente i danni strutturali, allo scopo di ridurre le perdite umane ed economiche.

Negli ultimi anni, la rapida crescita delle tecnologie e della Computer Science, ha permesso l’utilizzo di tecniche di Machine Learning nei più svariati campi, incluso l’Ingegneria Civile e la Resilienza Urbanistica.

Uno dei parametri più influenti per la valutazione delle perdite, dovute ad eventi sismici, è l’estensione dei detriti e gli effetti associati ad essa sulle infrastrutture di importanza strategica. L’accumulo di macerie può infatti provocare l’interruzione della rete stradale, o parte di essa. Tutto ciò implica un generale aumento delle persone che incontrano difficoltà ad evacuare determinate zone, con l’effettivo rischio che alcune persone non riescano affatto ad evacuare in tempo.

Utilizzando gli algoritmi di Machine Learning più aggiornati e il linguaggio di programmazione Python, lo scopo di questa tesi è di raccogliere dati ottenuti dagli eventi sismici precedenti, per predire l’estensione dei detriti. Questo modello può, come fine ultimo, permettere agli enti di Protezione Civile di pianificare gli interventi di salvataggio, riducendo il rischio di essere bloccati dalle macerie durante il percorso.

Infine, la PHI Challenge organizzata dal PEER, fornisce l’occasione giusta per introdurre le tecniche di Deep Learning applicate al riconoscimento automatico di Immagini da dividere in categorie legate all’Ingegneria Sismica. Tale argomento infatti, risulta essere uno dei più popolari e discussi nello stato dell’arte attuale della ricerca che collega Machine Learning all’Ingegneria Civile.
# TABLE OF CONTENTS

List of Figures .................................................................................................................. V
List of Tables ......................................................................................................................... VII

Chapter 1 Introduction ........................................................................................................ 1
  1.1 Python, tools and modules .......................................................................................... 2
  1.2 Layout of the Thesis ................................................................................................... 3

Chapter 2 DATA COLLECTION ...................................................................................... 5
  2.1 Dataset sources ......................................................................................................... 5
  2.2 Earthquakes ............................................................................................................... 8
    2.2.1 Central Italy ......................................................................................................... 8
    2.2.2 Cephalonia ......................................................................................................... 9
    2.2.3 Christchurch ....................................................................................................... 10
    2.2.4 Ecuador ............................................................................................................... 10
    2.2.5 India .................................................................................................................. 11
    2.2.6 Mexico Central .................................................................................................. 12
    2.2.7 Nepal ................................................................................................................ 12
    2.2.8 South Napa Valley ............................................................................................ 13
    2.2.9 Southern Taiwan ............................................................................................... 14
  2.3 Dataset features ....................................................................................................... 14
    2.3.1 Debris extension ............................................................................................... 15
  2.4 Dataset ..................................................................................................................... 18
Chapter 3 MACHINE LEARNING METHODOLOGIES ........................................ 21

3.1 Machine Learning .................................................................................. 21

3.2 Machine Learning algorithms ................................................................. 22

3.2.1 k-Nearest Regressor ........................................................................... 23

3.2.2 Linear models .................................................................................... 24

3.2.3 Decision Trees ................................................................................... 26

3.2.4 Support Vector Regression (SVR) ....................................................... 28

3.2.5 Multilayer Perceptrons (MLP) Regressor .......................................... 29

Chapter 4 DATA VISUALIZATION AND PREPROCESSING .................. 31

4.1 Data visualization .................................................................................... 31

4.1.1 Stories-EOD ....................................................................................... 31

4.1.2 Year-EOD ......................................................................................... 32

4.1.3 Magnitude-EOD ................................................................................ 33

4.1.4 Distance from Epicenter-EOD ............................................................ 33

4.1.5 Height-EOD ...................................................................................... 34

4.1.6 Logarithmic values ............................................................................ 34

4.1.7 t-Distributed Stochastic Neighbor Embedding (t-SNE) ................... 35

4.2 Preprocessing data ................................................................................ 36

Chapter 5 DATA TESTING AND RESULTS ........................................... 37

5.1.1 Data tuning ....................................................................................... 37

5.2 Data testing ............................................................................................ 39

5.2.1 R-squared ........................................................................................ 39

5.2.2 Mean Absolute Relative Distance (MARD) .................................... 40

5.3 Results ................................................................................................. 41

5.3.1 KNR ................................................................................................. 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>Linear models</td>
<td>43</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Decision Tree</td>
<td>47</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Random forest</td>
<td>49</td>
</tr>
<tr>
<td>5.3.5</td>
<td>SVR</td>
<td>50</td>
</tr>
<tr>
<td>5.3.6</td>
<td>MLP regressor</td>
<td>52</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Scores report</td>
<td>53</td>
</tr>
</tbody>
</table>

Chapter 6 PHI CHALLENGE AND IMAGE RECOGNITION .................................. 55

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>55</td>
</tr>
<tr>
<td>6.2</td>
<td>PHI Challenge tasks</td>
<td>56</td>
</tr>
<tr>
<td>6.3</td>
<td>Deep Learning models and Transfer Learning</td>
<td>60</td>
</tr>
<tr>
<td>6.4</td>
<td>Google Cloud Platform (GCP)</td>
<td>61</td>
</tr>
<tr>
<td>6.5</td>
<td>Training and results</td>
<td>62</td>
</tr>
</tbody>
</table>

Conclusions.................................................................................................. 64

Appendix............................................................................................................. 66

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Dataset</td>
<td>66</td>
</tr>
<tr>
<td>8.2</td>
<td>Extension of debris evaluation</td>
<td>71</td>
</tr>
<tr>
<td>8.3</td>
<td>Data visualization script</td>
<td>76</td>
</tr>
<tr>
<td>8.4</td>
<td>Training and test script</td>
<td>78</td>
</tr>
<tr>
<td>8.5</td>
<td>Results visualization script</td>
<td>82</td>
</tr>
<tr>
<td>8.6</td>
<td>PHI Challenge labeling algorithm</td>
<td>88</td>
</tr>
</tbody>
</table>

References...................................................................................................... 93
LIST OF FIGURES

Figure 2.1 Instance template of the images forming the dataset ........................................ 7
Figure 2.2 Torre dell'Orologio in Poggio Renatico (FE).......................................................... 9
Figure 2.3 Damaged building in Cephalonia ........................................................................... 9
Figure 2.4 Cathedral of the Blessed Sacrament Catholic in Christchurch.............................. 10
Figure 2.5 Concrete building damaged in Ecuador ................................................................. 11
Figure 2.6 Hotel damaged during the Earthquake occurred in India ........................................ 11
Figure 2.7 A building collapse during the Central Mexico earthquake .................................... 12
Figure 2.8 Damage in Nepal building after the seismic event .................................................. 13
Figure 2.9 Damaged building in South Napa after earthquake ............................................... 13
Figure 2.10 Removal of debris caused by a damaged building after the seismic event in Southern Taiwan ............................................................................................................. 14
Figure 2.11 The width of a car is used as reference measure ..................................................... 16
Figure 2.12 Evaluation of the debris extension ........................................................................ 16
Figure 3.1 Scikit-learn diagram to choose the correct algorithms ............................................. 22
Figure 3.2 Predictions made by k=1 ..................................................................................... 23
Figure 3.3 Predictions made by k=3 ..................................................................................... 24
Figure 3.4 Predictions of a linear model .................................................................................. 25
Figure 3.5 Decision tree learned on the data .......................................................................... 27
Figure 3.6 Instance of SVR fit line and boundaries ................................................................. 28
Figure 3.7 Illustration of a multilayer perceptron with a single hidden layer ......................... 29
Figure 4.1 Data representation: Number of stories-EOD .......................................................... 32
Figure 4.2 Data representation: Year of construction-EOD ..................................................... 32
Figure 4.3 Data representation: Magnitude-EOD ..................................................................... 33
Figure 4.4 Data representation: Distance from epicenter-EOD .............................................. 33
Figure 4.5 Data representation: Height-EOD .......................................................... 34
Figure 4.6 Data representation: Height-EOD with logarithmic values .................. 35
Figure 4.7 t-SNE data visualization..................................................................... 36
Figure 5.1 R-squared score for testing set about (a)15% (b)20% (c) 33%.............. 42
Figure 5.2 MARD score for testing set about (a)15% (b)20% (c) 33%............... 43
Figure 5.3 Lasso-R-squared score for testing set about (a)15% (b)20% (c) 33% ..... 43
Figure 5.4 Elastic Net-R-squared score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 44
Figure 5.5 Ridge-R-squared score for testing set about (a)15% (b)20% (c) 33%...... 45
Figure 5.6 Lasso-MARD score for testing set about (a)15% (b)20% (c) 33% ...... 45
Figure 5.7 Elastic Net-MARD score for testing set about (a)15% (b)20% (c) 33% . 46
Figure 5.8 Ridge-MARD score for testing set about (a)15% (b)20% (c) 33% ........ 47
Figure 5.9 Decision tree-R-squared score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 48
Figure 5.10 Decision tree-MARD score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 48
Figure 5.11 Random forest-R-squared score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 49
Figure 5.12 Random forest-MARD score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 50
Figure 5.13 SVR-R-squared score for testing set about (a)15% (b)20% (c) 33%...... 51
Figure 5.14 SVR-MARD score for testing set about (a)15% (b)20% (c) 33%........ 51
Figure 5.15 MLP regressor-R-squared score for testing set about (a)15% (b)20% (c) 33% ................................................................................................................. 52
Figure 5.16 MLP regressor-MARD score for testing set about (a)15% (b)20% (c) 33%................................................................................................................. 53
LIST OF TABLES

Table 2-1 Number of photographs for each earthquake event........................................ 7
Table 2-2 Parameters to evaluate the extension of debris in the instance id.061 .......... 17
Table 2-3 Quantity of samples for each feature............................................................ 18
Table 5-1 Alpha values for linear models regressor..................................................... 38
Table 5-2 R-squared and MARD scores........................................................................ 54
Table 6-1 Summary of Images in the training process.................................................. 59
Table 6-2 Summary of frozen layers .............................................................................. 61
Table 6-3 Nvidia v100 GPU characteristics................................................................. 61
Table 6-4 Summary of the training and testing process................................................. 63
CHAPTER 1
INTRODUCTION

After every disaster, a huge number of images are collected by reconnaissance teams formed by professional engineers, academic researchers and graduate students. All of them are charged with collecting perishable data to be used for:

- learn as much as possible about the nature of the event and the extent of the consequences;
- identify potential gaps in existing research or in the application of engineering knowledge;
- make recommendations regarding the need of further investigations and/or changes to codes and laws.

All this collected data can be used to reduce social and economic losses that follow strong ground motions. Indeed, after a seismic event, often a large amount of debris is generated that might be a critical obstacle to emergency operations and evacuations.

Starting from existing collections of data, the purpose of this thesis is to forecast the extension of the debris and, eventually, to determine whether a road is blocked or not.

In this way, it is possible to lead a procedure that can predict the best path for rescue operations and increase the possibility for a better action in term of human security and time safe.

Using the methodologies of modern Machine Learning (ML), it is possible to evaluate the accuracy of our predictions and increasing constantly the precision adding more and new data that could be collected in the future events.
Because most data collected by reconnaissance teams are pictures, it was considerate appropriate obtain evaluations about the amount of rubble from them. Indeed, this parameter is not directly reported inside the reports that are compiled around the world and the future availability of new data, taken directly on field, could improve significantly the results of this study. Once the extension of debris is extracted from the dataset (as shown in Chapter 2), the most famous and important ML algorithms is used to forecast a prediction. ML indeed, represents one of the most powerful and modern instruments to make predictions and is widely used around the world in different fields. The first result of the present research is to forecast, for new data, the amount of rubble, giving some inputs as magnitude, height, material and distance from epicenter. Finally, it is described another important aspect that connects ML and Earthquake resilience: Image Recognition (IR). This aspect continues to take the field in the Earthquake Engineering investigations and proof of this is the challenge sponsored by PEER. Using Google Cloud Project (GCP) and the deep learning most modern models, the concerned thesis covers also this aspect.

1.1 Python, tools and modules

Running ML algorithms required a good knowledge of programming language and Computer Science. One of the powerful and widely used programming language is Python, an open-source project that spreads out in the last few years. The power of this language lies on his flexibility due to the versatility of his modules and tools. Unlike MATLAB, indeed, a lot of functionalities could be downloaded as different packs and installed separately from the main program. This allows the availability of a huge number of different packages, easy to install and to use for every purpose. In the list below is listed the modules used in this research and what is their function:
• Scikit-learn: this module is constantly being developed and improved, thanks to a very active community. It contains several state-of-art ML algorithms, as well as comprehensive documentation about each algorithm.

• Numpy: surely one of the fundamental packages for data analysis in Python. It contains functionality for multidimensional arrays, high-level mathematical functions such as linear algebra operations.

• Scipy: is a collection of functions for scientific computing. It provides special mathematical functions and statistical distribution.

• Matplotlib: is the primary scientific plotting library in Python. It provides functions for making publication-quality visualizations such as line charts, scatter plots, and so on.

• Tensorflow: is a powerful library used for train the Convolutional Neural Network (CNN) models for the PHI Challenge

• Keras: is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. All ingredients within CNN mentioned in previous paragraphs are available in Keras, one only need to call those function in a reasonable order.

1.2 Layout of the Thesis

The present work is divided into seven chapters, described below:

- Chapter 2 shows how the dataset is built and how the extension of the debris is calculated for each sample.

- Chapter 3 presents the description of all the ML algorithm that are used to train, test and validate the data.

- Chapter 4 describes how the training and preprocessing operations are conducted and visualizes the dataset to give important insights for how to use them.
- *Chapter 5* summarizes the testing phase and give the results in terms of accuracy for each ML algorithm.

- *Chapter 6* describes the PHI Challenge organized by the Pacific Earthquake Engineering Research (PEER) about Image classification for detecting damage in buildings from pictures taken after seismic event.

- *Conclusions* at the end resume what has been done in the present work, running through the main features. This chapter gives an interpretation about the results as well as final consideration about how improves them and how could be conducted further researches on this field.
CHAPTER 2
DATA COLLECTION

This chapter describes in detail the process of creating the dataset used for train and test the ML algorithms. The sources of the images are mentioned as well as the methodology used to extracting information from that.

2.1 Dataset sources

In a typical seismic evaluation mission, a group is dispatched to a region where a seismic event has taken place. The teams may follow established guidelines, that can vary nation by nation, to provide a well-informed dataset, where it is possible to obtain important information as location of the damaged buildings, severity of the event, materials involved in the damaged and so on.

In the recent years many databases, publicly available, come out and they are usable for every kind of research and activities.

The dataset used for this thesis comes from these main sources:

- EERI clearinghouse: The Earthquake Engineering Research Institute (EERI) is a national, non-profit, technical society of engineers, geoscientists, architects, planners, public officials, and social scientists. EERI members include researchers, practicing professionals, educators, government officials, and building code regulators. The objective of the Earthquake Engineering Research Institute is to reduce earthquake risk by (1) advancing the science and practice of earthquake engineering, (2) improving understanding of the impact of earthquakes on the physical, social, economic, political, and
cultural environment, and (3) advocating comprehensive and realistic measures for reducing the harmful effects of earthquakes.

- Datacenterhub.org: as written on the internet site: “Too often, the results of years of engineering and scientific inquiry are stored on media that become outdated, broken, misplaced, or are simply not accessible. Even when these data are accessible, it can be difficult to interpret them. DataCenterHub seeks to alleviate this problem by providing a simple, standardized yet flexible platform to preserve and share data. In the future, this platform will offer data visualization tools and the ability to compare directly data from different sources. We plan to archive each uploaded dataset on Purdue’s institutional repositories (FORTRESS and PURR) to ensure the longevity of the data contributed to DataCenterHub. While funding is available, we shall also maintain a more readily accessible copy of all data on DataCenterHub.

- Db.concretecoalition.org: This database contains over 50 case studies on concrete buildings damaged in over 20 earthquakes. Each case study provides detailed information about the building and photographs of damage.

- GEER: The Geotechnical Extreme Events Reconnaissance (GEER) Association was formed as an outgrowth of grassroots efforts to investigate and document the geotechnical impacts of the 1989 Loma Prieta Earthquake, 1994 Northridge Earthquake, and 1995 Kobe Earthquake. Following these earthquakes, members of the geotechnical earthquake engineering community responded with ad hoc reconnaissance teams that relied on past personal and professional relationships.

These databases provide thousands of images (around 15000), but the purpose of the thesis is to study only pictures where the extent of the rubble is clearly visible and could be measured and compared with other elements present inside the photographs (as shown in the instance below). For that reason, a big effort was required to select manually each picture from these sources to build the dataset necessary for the concerned research.
Following this approach, a database of 198 photographs after past earthquakes events is gathered from these sources and each picture shows an entire building (or, at least, a good portion of it) that was damaged or collapsed by an earthquake event and his extension of rubble after damage.

The pictures cover an amount of 14 different earthquakes as shown in Table 2-1:

<table>
<thead>
<tr>
<th>Earthquake</th>
<th>Pictures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Italy</td>
<td>38</td>
</tr>
<tr>
<td>Cephalonia (Greece)</td>
<td>6</td>
</tr>
<tr>
<td>Christchurch (New Zealand)</td>
<td>9</td>
</tr>
<tr>
<td>Ecuador</td>
<td>38</td>
</tr>
<tr>
<td>India</td>
<td>6</td>
</tr>
</tbody>
</table>
In the following paragraphs, a brief description about the earthquakes as well as some useful considerations, are provided for each earthquake (except for Loma Prieta, Northern Iran, Northridge and Turkey, where less than 6 samples are taken from these earthquakes).

### 2.2 Earthquakes

#### 2.2.1 Central Italy

The Central Italy earthquake it happened in two different events, the first on 24 August 2016 and the second on 26 October 2016. The magnitude of these Earthquakes was around 5.7. The main part of building hit by the ground motion was built in masonry and dates from more than one hundred years ago.
2.2.2 Cephalonia

In 2014 a shake hit the island of Cephalonia with a Magnitude of 6. The general damage was not too heavy, but some buildings lost some debris and create rubble on the streets.
2.2.3 Christchurch

An earthquake with a magnitude of 6.3 hit the city of Christchurch in New Zealand on 22 February of 2011. This event caused a lot of building damage, including the Cathedral, and an amount of 185 victims.

![Figure 2.4 Cathedral of the Blessed Sacrament Catholic in Christchurch](image)

2.2.4 Ecuador

The 16\textsuperscript{th} of April 2016 a 7.8 Magnitude seismic event occurred in Ecuador. It is one of the most tragic events for the nation since 1987 and the number of damage and victims is high. A lot of buildings, both in masonry and in reinforced concrete, were damaged, as well as some infrastructures.
2.2.5 India

The 2001 India earthquake, occurred in 26\textsuperscript{th} January, had a magnitude of 7.7. It caused more than 10’000 victims and destroyed a lot of buildings and infrastructures.
2.2.6 Mexico Central

A 7.1 of magnitude earthquake hit the central part of Mexico on 19 September 2017. The earthquakes caused damage in the states of Puebla and Morelos and around 370 people was killed during the seismic event.

2.2.7 Nepal

In 2015, on 25\textsuperscript{th} of April, a seismic event with a magnitude of 8.0 hit the nation of Nepal. It was the worst natural disaster from 1934 and caused thousands of victims and around $10 billion of total damage. Almost all the buildings were made in masonry, so the amount of debris was very huge in this case.
2.2.8 South Napa Valley

On 24th August 2014, in the south of Napa, an earthquake of 6.0 magnitude was occurred. The event was the biggest in the San Francisco Bay Area since the 1989 Loma Prieta earthquake.
2.2.9 Southern Taiwan

A 6.4 magnitude earthquake occurred in the Southern Taiwan on 6th February 2016. Tainan was the city most hit by the event with a lot of building damage.

![Image of debris removal](image)

*Figure 2.10 Removal of debris caused by a damaged building after the seismic event in Southern Taiwan*

2.3 Dataset features

Quite possibly the most important part in ML process is understanding the data you are working with and how it relates to the task you want to solve.

In a larger context, the algorithms and methods in ML are only one part of a greater process to solve a problem, and it is always good to keep the big picture in mind. Indeed, no ML algorithm will be able to make a prediction on data for which it has no information.

For this reason, it is extremely important to feed the dataset with several features that allow to treat the problem and forecast a solution.
Looking at the purpose of this thesis, the following attributes have been kept in consideration because strictly connected with the resultant amount of debris that has been collected for each sample:

- **Material**: the buildings considered in this study are made in reinforced concrete or in masonry. The behavior of the two materials is quite different in term of extension of debris.
- **Stories**: tallest buildings could produce a bigger amount of rubble and could be interest see if there is a relation between height and extension of debris.
- **Year of construction**: construction technologies and methodologies can vary during the years because of both different regulations and construction technique.
- **Magnitude**: stronger is the ground motion and more are the possibilities to damage a building.
- **Distance from epicenter**: as the magnitude, buildings close to epicenter suffer from a stronger ground motion.
- **Height**: This parameter influences the seismic force (and, in consequence, the damages) that is felt by the structure since the Natural Period depending strongly from this value.

The complete table with all the pictures’ information is shown in the appendix 8.1.

### 2.3.1 Debris extension

The purpose of this thesis is to forecast the extension of debris after earthquake based on different parameters of the buildings. Unfortunately, this information is not reported inside the report that usually are written after every event.

A method to evaluate the rubble’s dimension from pictures is requested as explained in the next paragraphs.

Starting from a photograph, it is identified an object or a dimension, which the measure is well known; in the instance below, which correspond to the sample id.061 of the dataset, the standard width of a car is used as reference.
Figure 2.11 The width of a car is used as reference measure

Using PhotoFiltre, a photo editing program, it is possible evaluate the coordinates of the pixel of the two extreme points of a diagonal in the red square (both x axis and y axis).
Same operation is done to calculate the extension of the debris inside the picture.

Figure 2.12 Evaluation of the debris extension
Using a proportion (equation (1.1)), is possible to obtain an estimation of the rubble’s amount along one axis (x-axis in the instance):

\[
(1.1) d = p \cdot \frac{P}{D}
\]

Where:
- \( P = x_{i_r} - x_{2_r} \) (\( x_{i_r} \) and \( x_{2_r} \) are the pixel coordinates (x or y) of the reference element)
- \( p = x_{i_d} - x_{2_d} \) (\( x_{i_d} \) and \( x_{2_d} \) are the pixel coordinates (x or y) of the debris extension)
- \( D \) is the value in meters of the principal measure of the reference element
- \( d \) is the value in meters of the debris extension

Completing the instance, the parameters used to evaluate the rubble is summarize in the Table 2-2:

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{i_r} )</td>
<td>502</td>
</tr>
<tr>
<td>( x_{2_r} )</td>
<td>334</td>
</tr>
<tr>
<td>( x_{i_d} )</td>
<td>720</td>
</tr>
<tr>
<td>( x_{2_d} )</td>
<td>371</td>
</tr>
<tr>
<td>( p )</td>
<td>349</td>
</tr>
<tr>
<td>( P )</td>
<td>168</td>
</tr>
<tr>
<td>( D )</td>
<td>1.8 m</td>
</tr>
<tr>
<td>( d )</td>
<td>3.629 m</td>
</tr>
</tbody>
</table>

Once the extension of the debris is evaluated for all possible pictures, the estimation that is obtained is divided by the height of the building.
This operation is called “normalization” and is very common when samples, used for a ML study, are collected. This process is very useful for a series of reasons that are listed here:

- Makes training data less sensitive to the scale of features: in this study the height of a building heavily influences the general amount of the rubble.
- Minimize the variance of the dataset
- Consistency for comparing results across models: if other studies decide to cover this argument with other methods, with normalization is easier to compare results.
- Makes optimization well-conditioned: most of the ML optimizations are solved using gradient descent and the speed of convergence depends on the scaling of features.

For completeness, all the parameters used to calculate the extension of the debris are reported in the appendix section 8.2.

### 2.4 Dataset

Once the features and the methodologies required to build the dataset are shown, it is appropriate to summarize for each feature how many samples are available. Indeed, not for all the pictures it was possible to collect every parameter because the reports written all around the world are made by different teams and people, with different methodologies and instruments. Therefore, the next Table 2-3 is helpful to understand and see, for each feature, how many samples are collected.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Quantity of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>198</td>
</tr>
<tr>
<td>Stories</td>
<td>198</td>
</tr>
<tr>
<td>Year of construction</td>
<td>43</td>
</tr>
</tbody>
</table>
If there is a greater interest in the values and the composition of data, the complete dataset is available in the appendix section 8.1.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnitude</td>
<td>198</td>
</tr>
<tr>
<td>Distance from epicenter</td>
<td>170</td>
</tr>
<tr>
<td>Height</td>
<td>198</td>
</tr>
</tbody>
</table>
CHAPTER 3
MACHINE LEARNING
METHODOLOGIES

3.1 Machine Learning

Arthur Samuel in 1959 defined Machine Learning as:

“Field of study that gives computers the ability to learn without being explicitly programmed”

As the definition suggest, ML is about extracting knowledge from data. It is a research field at the intersection of Artificial Intelligence (AI), statistics and Informatics.

From the beginning of Computer Science and programming the idea behind ML was always well known by the scientists in the field of Informatic Engineer. However, the limited capacity both in term of calculation power and storage data capacity of the computers never allowed a practical and fruitful use of this camp. Only recently, ML methodologies has become ubiquitous in everyday life, thanks to the great improvement registered in the field of Informatic and computers.

From detecting spam, recognizing people inside photos and automatic recommendations of which movies to watch, many modern website and company have ML algorithms at their core.

The most successful kinds of ML methodologies are those that automate decision-making processes by generalizing from known examples, which is known as supervised learning. The concerned study provides the algorithm with pairs of input
and desired outputs in order to allow the algorithm to find a way to produce a desired output given an input. The concerned study is treated as a supervised learning problem with a regression task, where the answer to be learned is a continuous value.

### 3.2 Machine Learning algorithms

There are a lot of algorithms available in literature, so it is important to understand which one could be useful for the purpose (and is also important to have a deep comprehension of the data by visualizing them, as it will be described in Chapter 4). For this reason, Scikit-learn provides the chart shown in Figure 3.1 that help to choose the correct algorithm based on the dimension of the dataset and the problem that is dealt with.

![Scikit-learn algorithm cheat-sheet](image)

*Figure 3.1 Scikit-learn diagram to choose the correct algorithms.*

In the next paragraphs, all the different typologies of algorithms are introduced and explained as well as strengths and weakness for each method.
3.2.1 k-Nearest Regressor

The k-Nearest Neighbours (KNR) algorithm is arguably one of the simplest. It makes prediction for a new data point starting from the closest data in the training datasets, its “nearest neighbours”.

The k stands for how many samples are used to evaluate the prediction; in the following Figure 3.2 and Figure 3.3 is shown how change the prediction varying the value of k from 1 to 3.

In the first case, the target value is the same of the closest data. In the latter, the prediction is given by the mean of the relevant neighbours.

![Figure 3.2 Predictions made by k=1](image-url)
In conclusion, there are two important parameters for the KNR algorithm: the k value and how you measure distance between data points. Usually, the number of neighbours used is not more than 5 (to avoid overfitting, as it will be discussed in the next Chapter) and the lengths are measured by the Euclidean distance.

One of the strengths of KNR is that model is very easy to use and understand, and often give good result without a lot of adjustment.

On the other hand, with a large training set, the prediction can be slow, and the reliability of the result is less accurate.

### 3.2.2 Linear models

Linear models are the oldest method and is widely used for his simplicity and robustness.

The general prediction formula in this case, is represented by the following (3.1) equations:

\[
(3.1) \quad y = w_0 \cdot x_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n + b
\]
Where \( x_0 \) to \( x_n \) denotes the features and \( w \) and \( b \) are parameters of the model that are learned.

In conclusion, linear models try to predict labels approaching the training data as a straight line.

![Figure 3.4 Predictions of a linear model](image)

In literature, there are a lot of different linear models and their difference lies in how the parameters \( w \) and \( b \) are learned from the training data.

In this concerned study, the most popular linear regression models are used to predict the extension of debris.

### 3.2.2.1 Ridge regression

In ridge regression, the coefficients \( w \) are chosen not only so that they predict well on the training data, but also to fit an additional constraint. The target is to have the magnitude of coefficients to be as small as possible and so, all entries of \( w \) should be close to zero. In other words, this means each feature should have as little effect on the outcome as possible (which translates to having a small slopes), while still predicting well. This is a clear example of “regularization”, where the model is restricted to avoid overfitting.
How much simplicity is placed in the model is decided by the alpha parameter; increasing alpha, indeed, forces coefficients to move more toward zero, which decreases training set performances but might help generalization.

### 3.2.2.2 Lasso
The last linear model that is presented is Lasso. As with Ridge regression, also lasso restricts coefficients to be close to zero, but in a slightly different way. Indeed, in this case some of the coefficients are exactly zero. This means some features are completely ignored by the model; and this be a form of automatic feature selection. Also lasso has a regularization parameter alpha, that controls how strongly coefficients are pushed toward zero.

### 3.2.2.3 Elastic Net
Elastic net is a hybrid of ridge regression and lasso regularization. Like lasso, elastic net can generate reduced models by generating zero-valued coefficients. Empirical studies have suggested that the elastic net technique can outperform lasso on data with highly correlated predictors.

### 3.2.2.4 Linear models’ strengths and weaknesses
Linear models are very fast to train, and fast to predict. They scale to very large datasets and work well with sparse data. Another strength of linear models is that they make it relatively easy to understand how a prediction is made, using the formulas we saw earlier for regression and classification. Unfortunately, it is often not entirely clear why coefficients are the way they are.

### 3.2.3 Decision Trees
Decision trees are widely used models for regression tasks. Essentially, they are learning a sequence of if/else questions that gets us to the true answer most quickly. In the ML settings, these questions are called “tests”.
To build a tree, the algorithm searches over all possible tests and finds the one that is most informative about the target variable (see Figure 3.5).
Typically, using pure leaves (means that all the training data are at the end of the tree) leads the model to overfit the training data. To prevent that, there are two commons strategies: stopping the creation of the tree early (also called “pre-pruning”) or building the tree but then removing or collapsing nodes that contain little information (also called “post-pruning” or just pruning).

Decision trees have two advantages over many of the algorithms that are discussed: the resulting model can easily be visualized and understood by nonexperts, and the algorithms are completely invariant to scaling of the data.

The weakness of this method otherwise, still lies in the risk of overfitting data providing poor generalization, despite the use of pre-pruning or post-pruning.

Therefore, in most applications, the ensemble methods are usually used in place of a single decision tree.

3.2.3.1 Random Forests

As we just observed, a main drawback of decision trees is that they tend to overfit the training data. Random forests are one way to address this problem. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting but will likely overfit on part of the data. If we build many
trees, all of which work well and overfit in different ways, we can reduce the amount of overfitting by averaging their results.

To implement this strategy, we need to build many decision trees. Each tree should do an acceptable job of predicting the target and should also be different from the other trees. Random forests get their name from injecting randomness into the tree building to ensure each tree is different. There are two ways in which the trees in a random forest are randomized: by selecting the data points used to build a tree and by selecting the features in each split test.

### 3.2.4 Support Vector Regression (SVR)

In the SVR it is defined as hyper plane the line that will predict the continuous value that is the subject of the research. On the other hand, two boundary lines are situated at the distance ‘\( \varepsilon \)’ from the hyper plane.

The SVR algorithm counts the number of data that are inside the boundaries and the best fit line is the one that contain the maximum number of data.

![Figure 3.6 Instance of SVR fit line and boundaries](image-url)
3.2.5 Multilayer Perceptrons (MLP) Regressor

In the last few years a new methodology spreads out and it is so important that is considered a new field of ML, under the name “deep learning”. The main task for this method is Image Recognition where, thanks to the huge improvements in terms of both hardware power and number of data, in the last decade impressive results are reached.

MLP is a Convolutional Neural Network (CNN) that try to simulate the same procedures that occur in the neurons of the human brain. Each processing units indeed, behaves like neuron and therefore calculate a weighted sum of its inputs. Then, uses an activation function to form the single output value.

Usually, in a CNN we have an input layer where the features are collocated, then other layer called hidden, where the weights are calculated step by step and finally the output value.

Figure 3.7 Illustration of a multilayer perceptron with a single hidden layer
4.1 Data visualization

The data representation is one of the main tasks of a data scientist and ML practitioners trying to solve and it is known as feature engineering. Indeed, visualize the data in an efficient way can have a bigger influence on the results than the parameters tuning.

The purpose of this research is to estimate the extension of debris (EOD), so, in the next charts, is shown this quantity along the y-axis, meanwhile, in the x-axis are represented 5 different features (one for each graph).

In each graph, the data are divided in two categories: masonry and reinforced concrete building, as indicated in the label.

4.1.1 Stories-EOD

The number of stories of a building is not a continuous number, but a discrete one. For this reason, all the sample are situated only in the integer positions of the chart below.
4.1.2 Year-EOD

The year of construction is an interesting feature but unfortunately is very difficult to find this kind of information. Indeed, the number of samples of this graph is the smallest.
4.1.3 Magnitude-EOD

The data are grouped in 11 x coordinates because these are the magnitude of the different earthquakes.

![Magnitude-EOD](image)

*Figure 4.3 Data representation: Magnitude-EOD*

4.1.4 Distance from Epicenter-EOD

The distance from epicenter is expressed in kilometers and is measured taking the geographic coordinates of building and epicenter (so it’s measured in the surface).

![Distance from epicenter-EOD](image)

*Figure 4.4 Data representation: Distance from epicenter-EOD*
4.1.5 Height-EOD

This is the features with more and well distributed data.

4.1.6 Logarithmic values

Visualize the data help to understand if a correlation between features and target is possible. Looking to this data it is possible to say that linear correlation will not fit properly the dataset so the accuracy of the most part of the algorithm will not be high. Therefore, to improve the accuracy of our forecast, it is appropriate to use the logarithmic values of our data with the purpose to get a better linearization of the problem. In Figure 4.6 it is shown the new distribution of the data and it is clearly visible a higher linearization than the previous chart in Figure 4.5.
4.1.7 t-Distributed Stochastic Neighbor Embedding (t-SNE)

Since our problem is a multifeatured regression in 6 dimensions is useful utilize some tools that allow a sort of 2D visualization of the data in order to make more comprehensive the distribution of them. In 2008 an algorithm called t-SNE was developed to visualize high dimensional data using Barnes-Hut approximations. Using this algorithm for the dataset related to this research, the result is shown in Figure 4.7.

Looking at the graph it is possible to say that the R-squared value hardly could has a high score because for the same x-value on the graph we have more than one data in the y-axis (forward is explained better how works the R-squared coefficient).

Its useful repeat that the graph has only a visual importance, therefore the values in the axes have not any meaning for the concerned problem.
4.2 Preprocessing data

Once the data are collected it is important to scale them to avoid overfitting or other problems. Overfitting, or high variance, is when an algorithm, which is used to fit a training set, can have a high accuracy for the train set but a lower one for the predictions. In other words, overfitting occurs when the algorithm may fit the training set very well but fail to generalize to new examples.

In Scikit Learn there are many possibilities to scale the data but in this research the following are used:

- StandardScaler; this preprocessing ensures that for each feature the mean is 0 and the variance is 1, bringing all the features to the same magnitude. In this way it is possible to avoid odd data points, called outliers, that could create problem for the accuracy.
- MinMaxScaler; this procedure shifts the data such that all features are exactly between 0 and 1. For a two-dimensional dataset this means all the data is contained within the maximum and the minimum value of the feature.
The chapter is dedicated to the comparison of all the results obtained, in order to show which parameter are used to evaluate the accuracy and the reliability of the prediction.

### 5.1.1 Data tuning

For each different algorithm, several parameters or coefficients are needed to tune in order to obtain the best result in terms of R-squared and MARD.

In the next paragraphs the parameter necessary to find the best fit regression are described. For further information it is recommended to consult the documentation of Scikit-learn since in the concerned research only the parameters and the options modified are described but, for each algorithm, there are more coefficients to tune in general.

#### 5.1.1.1 KNR

For the KNR algorithm the parameter to tune is just one, that is “n_neighbors”. This represents the number of neighbors taken into consideration to evaluate the predictions for the X test. A value of 5 it is used to obtain the best results.

#### 5.1.1.2 Linear Models

For all the 3 linear models (Ridge, Elastic Net and Lasso) the parameter to tune is one, that is alpha. It gives an idea of how many samples are not considered by the
DATA TESTING AND RESULTS

model to avoid overfitting. Since the dataset is not huge the values of alpha are quite small.

Table 5-1 Alpha values for linear models regressor

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ridge</td>
<td>0.1</td>
</tr>
<tr>
<td>Elastic Net</td>
<td>0.01</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.01</td>
</tr>
</tbody>
</table>

5.1.1.3 Decision Tree

The coefficients used for Decision Tree are 3:

- Min_samples_split: gives the minimum number of samples required to split an internal node and it is set to 25.
- Min_samples_leaf: is the minimum number of samples required to be at a leaf node. A value of 10 it is used, it means that a split point at any depth will only be considered if it leaves at least 10 training samples in each of the left and right branches.
- Presort: setting this option to ‘True’ the data are presorted to speed up the finding of best splits in fitting. This is widely used for small dataset.

5.1.1.4 Random Forest

For the Random Forest algorithm, the parameters tuned are 3:

- Max_depth: indicates the maximum depth of the tree. It is set to 10 to avoid overfitting.
- N_estimators: gives the number of trees in the forest. A value of 20 it is used.
- Min_sample_split: has the same meaning as in the decision tree algorithm and it is set to 40.

5.1.1.5 SVR

The options tuned for the SVR are 3. C and epsilon, respectively 0.1 and 0.001, are parameters to produce no penalty associated in training loss function for points that are predicted within a certain distance (evaluated using C and epsilon) from the
actual value. Finally, the Kernel type ‘rbf’, which stands for Radial Basis Function, is used.

5.1.1.6 MLP Regressor

CNN required a lot of parameters to tune due to their nature of multilayer connective framework. In the concerned research the parameters set are 7:

- Alpha = 0.1; is the regularization parameter
- Batch_size = 10; is the number of samples included in the minibatches to train the model.
- Hidden_layer_sizes = 1000; is the number of neurons in the i-th hidden layer
- Learning_rate = ‘constant’;
- Learning_rate_init = 0.001;
- Activation = ‘relu’; defines the activation function for the hidden layer. ‘relu’ stands for Rectified Linear Unit Function.
- Solver = ‘adam’; defines the solver for weight optimization. ‘adam’ refers to a stochastic gradient-based optimizer proposed by Kingma, Diederik and Jimmy Ba.

5.2 Data testing

Once the data are preprocessed the next step is to tune the parameter for each algorithm in order to obtain the best accuracy for the task. But how is measured the accuracy?

For regression problems doesn’t exist a unique parameter to evaluate quality of our prediction, therefore in the concerned research are used the two most used parameters: R-squared and Mean Absolute Error (MARD).

5.2.1 R-squared

The R-squared value, also known as coefficient of correlation, provides a measure of how well future samples are likely to be predicted by the model valuating how much the scatter points are distant from the regression fit line calculated by the algorithm.
Best possible score is 1.0 and it can be negative (that means, for instance that the model is predicting descending values while the true data are growing). Furthermore, a constant model that always predicts the expected value of $y$, disregarding the input features, would get a R-squared score of 0.

The statistic definition of the coefficient of correlation is the following:

If $y_i$ is the predicted value of the $i$-th sample and $y_i^*$ is the corresponding true value, then the score $R^2$ estimated over $n_{\text{samples}}$ is defined as:

\[
R^2 (y, y) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - y_i^*)^2}{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \bar{y})^2}
\]

Where \(\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i \) (0.2).

### 5.2.2 Mean Absolute Relative Distance (MARD)

MARD is the average vertical distance between each point and the regression line. MARD is also the average horizontal distance between each point and the best fit line.

Therefore, lower is the value of MARD and better are the predictions.

In statistic the definition of MARD is the following:

If $y_i$ is the predicted value of the $i$-th sample, and $y_i^*$ is the corresponding true value, then the MARD estimated over $n_{\text{samples}}$ is defined as:

\[
MARD(y, y) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \left| \frac{y_i - y_i^*}{y_i^*} \right|
\]
5.3 Results

In the next paragraphs are shown the R-squared and MARD values for each algorithm. In the charts, shown for each algorithm, 3 colors are used, where;

- Blue bars refer to the score with a test size data about 15% of the dataset;
- Red bars refer to the score with a test size data about 20% of the dataset
- Green bars refer to the score with a test size data about 33% of the dataset

These quantities are the most common when an algorithm is trained. Furthermore, each algorithm is run with a different number of features. That because the number of samples are different if we consider this 3 groups of multiple features. Indeed, as shown in Table 2-1, if we consider all the features, the dataset it is composed by 43 features; if the “year of construction” is ignored, the dataset is formed by 170 samples and if also the “distance from epicenter” is ignored, all the dataset and his 198 samples are used.

5.3.1 KNR

Using the parameters described in 5.1.1.1, the KNR algorithm gives the following R-squared scores, shown in Figure 5.1.
The best score of 0.32 is obtained for the case (a), considering 3 features. In Figure 5.2 are shown the MARD score for each case. The best score is reached for the case (b) with 3 features.
5.3.2 Linear models

Linear models present, except for very few percentages of difference, the same scores both in terms of R-squared and MARD. In Figure 5.3, Figure 5.4 and Figure 5.5 are shown the R-squared scores for Lasso, Elastic Net and Ridge algorithms.

Figure 5.2 MARD score for testing set about (a) 15% (b) 20% (c) 33%

Figure 5.3 Lasso-R-squared score for testing set about (a) 15% (b) 20% (c) 33%
Figure 5.4 Elastic Net-R-squared score for testing set about (a) 15% (b) 20% (c) 33%
Figure 5.5 Ridge-R-squared score for testing set about (a) 15% (b) 20% (c) 33%

Linear models do not give good results in terms of R-squared and also the MARD scores are worse as it is possible to see in Figure 5.6, Figure 5.7, Figure 5.8.

Figure 5.6 Lasso-MARD score for testing set about (a) 15% (b) 20% (c) 33%
Figure 5.7 Elastic Net-MARD score for testing set about (a) 15% (b) 20% (c) 33%
5.3.3 Decision Tree

With decision tree algorithm the scores are better as it is shown in Figure 5.9.
The best score is in the case (a) with all the features. The same case gives also the best result in terms of MARD with a value of 0.23 as shown in Figure 5.10.

Figure 5.9 Decision tree-R-squared score for testing set about (a)15% (b)20% (c) 33%

Figure 5.10 Decision tree-MARD score for testing set about (a)15% (b)20% (c) 33%
5.3.4 Random forest

Random Forest gives the best scores both in terms of R-squared and MARD. Indeed, the R-squared score reach a total of 0.52 while the MARD score is 0.22. In the following Figure 5.11 and Figure 5.12 are shown the overall scores for each case.

(a)  
(b)  
(c)

*Figure 5.11 Random forest-R-squared score for testing set about (a) 15% (b) 20% (c) 33%*
DATA TESTING AND RESULTS

5.3.5 SVR

In the following graphs are reported the results for the SVR algorithm. In Figure 5.13 are shown the R-squared scores.
Figure 5.13 SVR-R-squared score for testing set about (a) 15% (b) 20% (c) 33%

In Figure 5.14 instead, are shown the MARD scores.

Figure 5.14 SVR-MARD score for testing set about (a) 15% (b) 20% (c) 33%
5.3.6 MLP regressor

The last algorithm used is the MLP regressor that gives good results. For larger dataset this algorithm could give improved results.

![MLP regressor-R-squared scores](image)

(a) ![MLP regressor-R-squared scores](image)

(b) ![MLP regressor-R-squared scores](image)

(c)

*Figure 5.15 MLP regressor-R-squared score for testing set about (a) 15% (b) 20% (c) 33%*

In Figure 5.15 it is possible to see R-squared scores, while in Figure 5.16 are reported the MARD values.
Figure 5.16 MLP regressor-MARD score for testing set about (a) 15% (b) 20% (c) 33%

5.3.7 Scores report

In Table 5-2 are reported all the values in numerical term for 3 different test set (15%, 20% and 33%) and for 3 different groups of features.
Table 5-2 R-squared and MARD scores

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of features</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>15%</td>
<td>20%</td>
<td>33%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNR</td>
<td>R²</td>
<td>0.42</td>
<td>0.47</td>
<td>0.36</td>
<td>0.38</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.32</td>
<td>0.35</td>
<td>0.34</td>
<td>0.34</td>
<td>0.36</td>
</tr>
<tr>
<td>Lasso</td>
<td>R²</td>
<td>0.54</td>
<td>0.53</td>
<td>0.47</td>
<td>0.54</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.53</td>
<td>0.60</td>
<td>0.50</td>
<td>0.53</td>
<td>0.60</td>
</tr>
<tr>
<td>Elastic Net</td>
<td>R²</td>
<td>0.54</td>
<td>0.54</td>
<td>0.48</td>
<td>0.54</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.53</td>
<td>0.60</td>
<td>0.50</td>
<td>0.53</td>
<td>0.60</td>
</tr>
<tr>
<td>Ridge</td>
<td>R²</td>
<td>0.54</td>
<td>0.54</td>
<td>0.48</td>
<td>0.54</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.52</td>
<td>0.58</td>
<td>0.55</td>
<td>0.51</td>
<td>0.57</td>
</tr>
<tr>
<td>Random Forest</td>
<td>R²</td>
<td>0.61</td>
<td>0.48</td>
<td>0.42</td>
<td>0.59</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.56</td>
<td>0.58</td>
<td>0.52</td>
<td>0.55</td>
<td>0.61</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>R²</td>
<td>0.56</td>
<td>0.56</td>
<td>0.43</td>
<td>0.52</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.60</td>
<td>0.63</td>
<td>0.55</td>
<td>0.59</td>
<td>0.63</td>
</tr>
<tr>
<td>SVR</td>
<td>R²</td>
<td>0.43</td>
<td>0.36</td>
<td>0.29</td>
<td>0.47</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.53</td>
<td>0.56</td>
<td>0.48</td>
<td>0.50</td>
<td>0.55</td>
</tr>
<tr>
<td>MLP Regressor</td>
<td>R²</td>
<td>0.49</td>
<td>0.54</td>
<td>0.48</td>
<td>0.50</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>MARD</td>
<td>0.53</td>
<td>0.61</td>
<td>0.54</td>
<td>0.53</td>
<td>0.61</td>
</tr>
</tbody>
</table>
CHAPTER 6
PHI CHALLENGE AND IMAGE RECOGNITION

6.1 Introduction

Another relevant aspect of ML that is spread out in the last years (from 2011) is Convolutional Neural Network (CNN). This topic has assumed such importance that it has earned its own name: Deep Learning. The models and the algorithms included in this category are improved every year and a big amount of challenge and competition comes out in different fields: from entertainment (company such as Netflix, Spotify made their own competition) to Financial online security.

One of the most popular field where Deep Learning is widely used is for the Image Recognition. With this term it is indicated all the methods that allow computer CPU or, above all, GPU to classify pictures in different categories, which implies to “understand” what is represented in the pictures relying on the categories.

There have been several recent image-based recognition competitions (such as the PASCAL VOC, ImageNet, and COCO challenges) based on natural objects and scenes.

On 18th January 2018 the Pacific Earthquake Engineering Research (PEER) center announced the PEER Hub ImageNet (PHI) Challenge that provided the use of
Computer Science and Deep Learning techniques to organize the first image-based structural damage recognition competition. As it is possible to read in the main online page of the competition: “In the PHI Challenge, PEER will provide a large image dataset which is relevant to the field of structural engineering, and will design several detection tasks, which will contribute to the establishment of automated vision-based structural health monitoring. The goal of the PHI challenge is to evaluate algorithms for structural image classification using a large-scale dataset based on service conditions and past reconnaissance efforts and laboratory experiments for conditions of extreme events. The state-of-the-art algorithms to be tested in the PHI challenge are expected to enhance the accuracy and the generalization of vision-based approaches. These approaches will aim towards the construction of a big structural image dataset to solve societal-scale problems of structural health monitoring and assessment of the built environment.”

The concerned study belongs to the classification problems (different from the regression one, which we treated in the other paragraph). Even if it is still a supervised learning problem, classification problem is different from regression because now the parameter that is tried to forecast is not continuous but relies on different categories that are individuated before the training phase.

### 6.2 PHI Challenge tasks

In this competition, all training images were labeled by Pacific Earthquake Engineering Center, and each task has at most 4 categories. All images are assumed to be labeled well without any outlier in any category. All images have been resized to 224 x 224 x 3, which means its width, height, and channel (RGB).

The tasks have different weights for the overall score based on difficulty factors. The tasks are:

**Easy:**

- **Task 1**: Scene classification, 3 classes (pixel/object/structural levels);
- **Task 2**: Damage check, 2 classes (yes/no);

  ![](undamaged.png)  ![damaged.png]

  Undamaged  Damaged

- **Task 3**: Spalling condition, 2 classes (yes/no);

  ![](no_spalling.png)  ![spalling.png]

  No spalling  Spalling

- **Task 4**: Material type, 2 classes (steel/others);
Medium:

- **Task 5**: Collapse check, 3 classes (no/partial collapse/collapse);

- **Task 6**: Component type, 4 classes (beam/column/wall/else);
• **Task 7**: Damage level, 4 classes (no/minor/moderate/heavy damage);

No damage  Minor damage  Moderate damage  Heavy damage

Hard:

• **Task 8**: Damage type, 4 classes (no/flexural/shear/combined);

No damage  Flexural damage  Shear damage  Combined damage

In our training process, 80% of images will be the training set, the rest 20% will be separated as the validation and test set equally (Table 6-1).

*Table 6-1 Summary of Images in the training process.*

<table>
<thead>
<tr>
<th>Task1</th>
<th>Task2</th>
<th>Task3</th>
<th>Task4</th>
<th>Task5</th>
<th>Task6</th>
<th>Task7</th>
<th>Task8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Set</strong></td>
<td>13,939</td>
<td>4,730</td>
<td>3,294</td>
<td>3,469</td>
<td>412</td>
<td>2,104</td>
<td>2,105</td>
</tr>
<tr>
<td><strong>Validation Set</strong></td>
<td>1,742</td>
<td>591</td>
<td>329</td>
<td>433</td>
<td>51</td>
<td>263</td>
<td>263</td>
</tr>
</tbody>
</table>
After the algorithm is trained, a database of unlabeled images is provided so that predications of labels and annotation can be generated.

All the competition takes place in the Kaggle platform, which is defined in Wikipedia as “an online community of data scientists and machine learners, owned by Google, Inc. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.”

Through this platform it is possible to update in real time the prediction of the test and see the accuracy of the model.

### 6.3 Deep Learning models and Transfer Learning

The general definition of Transfer Learning (TL) is written in [15] (see the references):

“Given a source domain and its learning task, a target domain and its target task, the objective of TL is to help improve the prediction function in learning target task using the knowledge from source domain with source target.”

In the concerned study, TL is practiced within the training process so that the lower level features (e.g., color, corner, and edge) can be used directly from pre-trained models (e.g., VGG16, VGG19, Inception, ResNet), which effectively reduce the training time with large-scale image set.

To obtain the baseline model, 4 state-of-the-art models have been chosen based on their performance in ImageNet Challenge, which includes VGG16, VGG19, Inception, and Xception. According to TL, lower convolutional bocks that have predictors (weights and biases) to capture common features such as edge, colour, and corner. Within the training process, those features do not need to be learned so that
the overall training time can be reduced. In order to skip the training for lower features, the number of layers that need to be frozen must be defined initialized so that those weights and biases will not be updated during the backward propagation process. The summary of frozen layers for different models in our training is shown in Table 6-2. The layer indicated in the table including convolutional layer, Rectified Linear Units (ReLU), max pooling layer, and fully connected layer.

Table 6-2 Summary of frozen layers

<table>
<thead>
<tr>
<th>Model</th>
<th>VGG16</th>
<th>VGG19</th>
<th>Inception</th>
<th>ResNet50</th>
<th>Xception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frozen Layers</td>
<td>25</td>
<td>20</td>
<td>197</td>
<td>160</td>
<td>54</td>
</tr>
</tbody>
</table>

6.4 Google Cloud Platform (GCP)

The dataset provided by PEER is formed by images with a dimension of 224x224 pixel. Applying this dimension for all the pictures gives an idea about the huge amount of work that is required for the algorithm to make a prediction. Therefore, treating this data needs a powerful machine that permits to run different CNN models in order to get the best accuracy in terms of predictions. Using commercial computer, even most powerful, could give memory error working with such big database. For that reason, to run the models in very powerful virtual machines it is used Google Cloud Platform (GCP), which is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. In this way, it was possible use a computer with the v100 GPU having the following characteristics:

Table 6-3 Nvidia v100 GPU characteristics

| Characteristic | Value |
### PHI CHALLENGE AND IMAGE RECOGNITION

<table>
<thead>
<tr>
<th>Transistors</th>
<th>21.1 billion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUDA cores</td>
<td>5,120</td>
</tr>
<tr>
<td>Tensor cores</td>
<td>640</td>
</tr>
<tr>
<td>GPU clock speed</td>
<td>1200MHz base, 1455MHz boost</td>
</tr>
<tr>
<td>Memory capacity</td>
<td>12GB HBM2</td>
</tr>
<tr>
<td>Memory clock</td>
<td>850MHz</td>
</tr>
<tr>
<td>Memory interface</td>
<td>3,072-bit</td>
</tr>
<tr>
<td>Total memory bandwidth</td>
<td>652.8GBps</td>
</tr>
<tr>
<td>Texture units</td>
<td>320</td>
</tr>
<tr>
<td>Power</td>
<td>250W TDP via 1x 6-pin and 1x 8-pin power connectors</td>
</tr>
<tr>
<td>Ports</td>
<td>3x DisplayPort, 1x HDMI</td>
</tr>
<tr>
<td>Price</td>
<td>$3,000 on Nvidia.com</td>
</tr>
</tbody>
</table>

#### 6.5 Training and results

For preprocessing, all input images were assigned as 224 x 224 x 3 at the beginning. However, each model has its own default input size. For example, VGG16, VGG19, and ResNet require image size 224 x 224 x 3. Both InceptionV3 and Xception require image size 299 x 299 x 3. Besides the modification of size, normalization is also implemented for all images. That is, the pixel must be divided by 255 so that its value is between 0 and 1. Based on the observation, the given images have some rotation and flip that can be confusing during the learning process. As a result, 10% of input images will be randomly rotated and the other 10% will be randomly flipped to make sure that those confusing example in the real test set can be captured.

The overall hyperparameters that utilized in our training process are shown in Table 6-4. The initial learning rate is fixed to all tasks but with different decay rate and epoch based on the size of images. The decay rate can make sure the learning process
will not go too fast to skip the global solution. The accuracy of each task is the result by voting the most predicting category among all models.

Table 6-4 Summary of the training and testing process.

<table>
<thead>
<tr>
<th></th>
<th>Task1</th>
<th>Task2</th>
<th>Task3</th>
<th>Task4</th>
<th>Task5</th>
<th>Task6</th>
<th>Task7</th>
<th>Task8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mini-Batch</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td><strong>Initial Learning Rate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>Learning Rate Decay</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
<td>1e -03</td>
</tr>
<tr>
<td><strong>Epoch</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>6,000</td>
<td>5,000</td>
<td>5,000</td>
<td>2,000</td>
<td>3,000</td>
<td>3,000</td>
<td>3,000</td>
<td>3,000</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.907</td>
<td>0.791</td>
<td>0.83</td>
<td>0.967</td>
<td>0.625</td>
<td>0.684</td>
<td>0.608</td>
<td>0.559</td>
<td></td>
</tr>
</tbody>
</table>
The subject of this thesis was the development of a methodology to assess the extension of the debris caused by earthquake damage. The amount of rubble is estimated from pictures taken in the field and several modern Machine Learning algorithms are evaluated based on their prediction accuracy in terms of R-squared and Mean Absolute Relative Distance (MARD). The work begins with a collection of 198 pictures (and related data) from 14 different earthquakes. Using this dataset and 6 features, 8 different algorithms were trained and their performance in terms of predicting the extension of the debris was evaluated. During the training process, some parameters, different for each algorithm, were altered and tested in order to achieve the best score. Looking at the results, the main features of the present research can be summarized as follows:

- KNR algorithm gives the best MARD score (0.32) but the low value of R-squared (0.42) means that with more data the algorithm, probably, will generalize too much.
- Linear models furnish similar results;
- Random forest gives the best R-squared but a high mean absolute relative deviation (MARD) (0.56). The predictive performance can be improved if more data is collected;
- MLP Regressor, which is a type of Convolutional Neural Network, does not achieve good results because of the size of the dataset that does not allow the network to exploit his power.
- The best score, both for R-squared and MARD, are reached considering all the features and a testing set of 15%. This is due to the medium-small
dimension of the dataset. If more data are collected, maybe, better results can be reached for bigger testing data (which could give more consistence values).

In conclusion Random forest and KNR are the best choices for predicting the debris-extension and the performances suggest that there is some potential for using machine learning methodologies in this field. It is appropriate to underline that collecting more pictures and data from other Earthquakes in future could significantly improve the results.

In the last part of the concerned thesis, it is described the PHI Challenge, organized by PEER.

In this competition, state-of-the-art CNN and Transfer Learning are implemented to classify the structural image into multi-category which includes the damage status before and after the disaster as wells the materials and structural components. Due to the pre-trained models that are available in Keras/TensorFlow, the training process can be reduced to focus on the higher-level features for a specific category. GCP also provides a friendly environment for the beginner of Deep Learning who may not have enough computing power to fulfill large-scale image classification, which is also utilized in this competition.

The overall performance and the process give a good chance for civil engineers to learn the new technology and utilize it to improve current research topics.
APPENDIX

8.1 Dataset

Legend:
1. ID = Identification number of the picture
2. Earthquake = Earthquake location
3. Mat. = Material:
   - 0 = Masonry
   - 1 = Reinforced Concrete
4. Stories
5. Year
6. Magnitude
7. Distance = Distance from epicenter in kilometers
8. Log(Height) = logarithm of the buildings’ heights
9. Log(EOD Norm.) = Logarithmic value of Extension Of Debris normalized with the heights of the buildings
10. d = Extent of debris calculated following the procedure in the paragraph 2.3.1

Note: A missing data is replaced with the value -1.

<table>
<thead>
<tr>
<th>ID</th>
<th>Earthquake</th>
<th>Mat</th>
<th>Stories</th>
<th>Year</th>
<th>Magnitude</th>
<th>Distance [km]</th>
<th>Log(Height) [m]</th>
<th>Log(EOD Norm.)</th>
<th>d [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>id.001</td>
<td>Mexico City</td>
<td>1</td>
<td>8</td>
<td>-1</td>
<td>8.1</td>
<td>60</td>
<td>0.826075</td>
<td>-0.4973</td>
<td>1.9091</td>
</tr>
<tr>
<td>id.002</td>
<td>Mexico City</td>
<td>0</td>
<td>12</td>
<td>-1</td>
<td>8.1</td>
<td>51</td>
<td>0.812913</td>
<td>0.093912</td>
<td>7.4483</td>
</tr>
<tr>
<td>id.003</td>
<td>Mexico City</td>
<td>1</td>
<td>7</td>
<td>-1</td>
<td>8.1</td>
<td>52</td>
<td>0.778151</td>
<td>-0.47716</td>
<td>2</td>
</tr>
<tr>
<td>id.004</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>8.1</td>
<td>48</td>
<td>1.30103</td>
<td>-0.80272</td>
<td>2.8356</td>
</tr>
<tr>
<td>id.006</td>
<td>Mexico City</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>8.1</td>
<td>54</td>
<td>0.90309</td>
<td>-0.41016</td>
<td>3.5</td>
</tr>
<tr>
<td>id.007</td>
<td>Northwestern Armenia</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.9</td>
<td>47</td>
<td>0.94483</td>
<td>-0.66655</td>
<td>1.9397</td>
</tr>
<tr>
<td>id.009</td>
<td>Northwestern Armenia</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>6.9</td>
<td>168</td>
<td>0.778151</td>
<td>-0.37069</td>
<td>2.5556</td>
</tr>
<tr>
<td>id.010</td>
<td>Northwestern Armenia</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>6.9</td>
<td>167</td>
<td>1.255273</td>
<td>-0.68909</td>
<td>3.6833</td>
</tr>
<tr>
<td>id.011</td>
<td>Northwestern Armenia</td>
<td>0</td>
<td>5</td>
<td>-1</td>
<td>6.9</td>
<td>170</td>
<td>0.939519</td>
<td>-0.57955</td>
<td>2.3697</td>
</tr>
<tr>
<td>id.012</td>
<td>Northwestern Armenia</td>
<td>0</td>
<td>5</td>
<td>-1</td>
<td>6.9</td>
<td>170</td>
<td>0.90309</td>
<td>-0.13829</td>
<td>6.5455</td>
</tr>
<tr>
<td>id.013</td>
<td>Loma Prieta</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.1</td>
<td>175</td>
<td>1.041393</td>
<td>-0.93033</td>
<td>1.0568</td>
</tr>
<tr>
<td>id.014</td>
<td>Loma Prieta</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>1</td>
<td>-0.8617</td>
<td>1.2371</td>
</tr>
<tr>
<td>id.015</td>
<td>Loma Prieta</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.1</td>
<td>17</td>
<td>1.255273</td>
<td>-1.51428</td>
<td>0.55</td>
</tr>
<tr>
<td>id.016</td>
<td>Loma Prieta</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>7.1</td>
<td>16</td>
<td>0.954243</td>
<td>-0.18092</td>
<td>5.9333</td>
</tr>
<tr>
<td>id.017</td>
<td>Loma Prieta</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>7.1</td>
<td>17</td>
<td>1.255273</td>
<td>-0.80715</td>
<td>2.8056</td>
</tr>
<tr>
<td>id.019</td>
<td>Northern Iran</td>
<td>0</td>
<td>6</td>
<td>-1</td>
<td>7.7</td>
<td>14</td>
<td>0.954243</td>
<td>-0.96417</td>
<td>0.9778</td>
</tr>
<tr>
<td>id.020</td>
<td>Northridge</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.8</td>
<td>13</td>
<td>1.20412</td>
<td>-0.95117</td>
<td>1.678</td>
</tr>
<tr>
<td>id.021</td>
<td>Northridge</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.8</td>
<td>15</td>
<td>0.845098</td>
<td>-0.51727</td>
<td>1.8235</td>
</tr>
<tr>
<td>id.022</td>
<td>Central Italy</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.1</td>
<td>36</td>
<td>0.845098</td>
<td>-0.55533</td>
<td>1.6702</td>
</tr>
<tr>
<td>id.023</td>
<td>Central Italy</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.1</td>
<td>35</td>
<td>0.778151</td>
<td>-0.63097</td>
<td>1.4033</td>
</tr>
<tr>
<td>id.024</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>1935</td>
<td>6.1</td>
<td>-1</td>
<td>0.982271</td>
<td>-0.22863</td>
<td>5.3165</td>
</tr>
<tr>
<td>id.025</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>1935</td>
<td>6.1</td>
<td>-1</td>
<td>1</td>
<td>-0.25524</td>
<td>5</td>
</tr>
<tr>
<td>id.026</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>6</td>
<td>1935</td>
<td>6.1</td>
<td>-1</td>
<td>1.255273</td>
<td>-0.08197</td>
<td>14.9032</td>
</tr>
<tr>
<td>id.028</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.1</td>
<td>34</td>
<td>0.819544</td>
<td>-0.49785</td>
<td>1.9068</td>
</tr>
<tr>
<td>id.029</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.1</td>
<td>33</td>
<td>0.799341</td>
<td>-0.83062</td>
<td>0.886</td>
</tr>
<tr>
<td>id.030</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.1</td>
<td>37</td>
<td>0.954243</td>
<td>-0.86012</td>
<td>1.2424</td>
</tr>
<tr>
<td>id.032</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>1864</td>
<td>6.1</td>
<td>48</td>
<td>0.954243</td>
<td>-0.83893</td>
<td>1.3043</td>
</tr>
<tr>
<td>id.033</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>1864</td>
<td>6.1</td>
<td>48</td>
<td>0.90309</td>
<td>-1.07935</td>
<td>0.75</td>
</tr>
<tr>
<td>id.035</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>5</td>
<td>1744</td>
<td>6.1</td>
<td>57</td>
<td>1.176091</td>
<td>-0.92812</td>
<td>1.77</td>
</tr>
<tr>
<td>id.036</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.1</td>
<td>35</td>
<td>1.012837</td>
<td>-0.59965</td>
<td>2.2629</td>
</tr>
<tr>
<td>id.039</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>5</td>
<td>1401</td>
<td>6.1</td>
<td>-1</td>
<td>1.176091</td>
<td>-0.75979</td>
<td>2.6086</td>
</tr>
<tr>
<td>id.040</td>
<td>Emilia Romagna</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.1</td>
<td>69</td>
<td>0.944483</td>
<td>-0.74256</td>
<td>1.6279</td>
</tr>
<tr>
<td>id.043</td>
<td>Emilia Romagna</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.1</td>
<td>198</td>
<td>0.875061</td>
<td>-0.80052</td>
<td>0.95</td>
</tr>
<tr>
<td>id.044</td>
<td>Emilia Romagna</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>6.1</td>
<td>200</td>
<td>0.954243</td>
<td>-1</td>
<td>0.9</td>
</tr>
<tr>
<td>id.045</td>
<td>South Napa Valley</td>
<td>0</td>
<td>3</td>
<td>1940</td>
<td>6</td>
<td>28</td>
<td>0.968483</td>
<td>-0.54668</td>
<td>2.5556</td>
</tr>
<tr>
<td>id.046</td>
<td>South Napa Valley</td>
<td>0</td>
<td>3</td>
<td>1940</td>
<td>6</td>
<td>29</td>
<td>0.968483</td>
<td>-0.5373</td>
<td>2.6122</td>
</tr>
<tr>
<td>id.049</td>
<td>South Napa Valley</td>
<td>0</td>
<td>3</td>
<td>1905</td>
<td>6</td>
<td>30</td>
<td>9</td>
<td>0.121727</td>
<td>1.095541</td>
</tr>
<tr>
<td>id.050</td>
<td>South Napa Valley</td>
<td>0</td>
<td>2</td>
<td>1940</td>
<td>6</td>
<td>31</td>
<td>0.939519</td>
<td>-0.20363</td>
<td>5.6316</td>
</tr>
<tr>
<td>id.051</td>
<td>South Napa Valley</td>
<td>0</td>
<td>2</td>
<td>1940</td>
<td>6</td>
<td>33</td>
<td>0.778151</td>
<td>-0.65995</td>
<td>1.3125</td>
</tr>
<tr>
<td>id.053</td>
<td>South Napa Valley</td>
<td>0</td>
<td>1</td>
<td>1940</td>
<td>6</td>
<td>27</td>
<td>0.826075</td>
<td>-0.285</td>
<td>3.1125</td>
</tr>
<tr>
<td>id.054</td>
<td>South Napa Valley</td>
<td>0</td>
<td>1</td>
<td>1940</td>
<td>6</td>
<td>30</td>
<td>0.69897</td>
<td>-0.18329</td>
<td>3.2784</td>
</tr>
<tr>
<td>id.055</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.4</td>
<td>202</td>
<td>1.255273</td>
<td>-0.85356</td>
<td>2.5213</td>
</tr>
<tr>
<td>id.056</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.4</td>
<td>-1</td>
<td>0.973128</td>
<td>-0.58939</td>
<td>2.3163</td>
</tr>
<tr>
<td>id.057</td>
<td>Puebla</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>1.176091</td>
<td>-1.07314</td>
<td>1.2676</td>
</tr>
<tr>
<td>id.059</td>
<td>Puebla</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>1</td>
<td>-0.52476</td>
<td>2.6886</td>
</tr>
<tr>
<td>id.060</td>
<td>Puebla</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>0.755875</td>
<td>-0.1224</td>
<td>4.5263</td>
</tr>
<tr>
<td>id.061</td>
<td>Puebla</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>0.954243</td>
<td>-0.38143</td>
<td>3.7393</td>
</tr>
<tr>
<td>id.062</td>
<td>Puebla</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.1</td>
<td>-1</td>
<td>0.954243</td>
<td>-0.39448</td>
<td>3.6289</td>
</tr>
<tr>
<td>id.064</td>
<td>Christchurch</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>-1</td>
<td>1.255273</td>
<td>-0.51089</td>
<td>5.5514</td>
</tr>
<tr>
<td>id.068</td>
<td>Christchurch</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>411</td>
<td>1.518514</td>
<td>-0.35018</td>
<td>14.7342</td>
</tr>
<tr>
<td>id.069</td>
<td>Christchurch</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>403</td>
<td>0.845098</td>
<td>-0.27368</td>
<td>3.195</td>
</tr>
<tr>
<td>id.070</td>
<td>Christchurch</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>-1</td>
<td>0.69897</td>
<td>-0.21254</td>
<td>3.6783</td>
</tr>
<tr>
<td>id.079</td>
<td>Christchurch</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>160</td>
<td>0.934498</td>
<td>-0.57073</td>
<td>2.4187</td>
</tr>
<tr>
<td>id.081</td>
<td>Christchurch</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>158</td>
<td>0.799341</td>
<td>-0.35566</td>
<td>2.6452</td>
</tr>
<tr>
<td>id.082</td>
<td>Christchurch</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>162</td>
<td>0.778151</td>
<td>-0.00753</td>
<td>5.8966</td>
</tr>
<tr>
<td>id.084</td>
<td>Christchurch</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>230</td>
<td>0.954243</td>
<td>-0.31292</td>
<td>4.3784</td>
</tr>
<tr>
<td>id.086</td>
<td>Christchurch</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>229</td>
<td>1.255273</td>
<td>-0.70708</td>
<td>3.5342</td>
</tr>
<tr>
<td>id.087</td>
<td>Cephalonia</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6</td>
<td>232</td>
<td>0.778151</td>
<td>-1.03953</td>
<td>0.5479</td>
</tr>
<tr>
<td>id.088</td>
<td>Cephalonia</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6</td>
<td>228</td>
<td>0.954243</td>
<td>-1.04048</td>
<td>0.8203</td>
</tr>
<tr>
<td>id.089</td>
<td>Cephalonia</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6</td>
<td>233</td>
<td>1.255273</td>
<td>-1.43181</td>
<td>0.6667</td>
</tr>
<tr>
<td>id.090</td>
<td>Cephalonia</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6</td>
<td>230</td>
<td>1.255273</td>
<td>-1.39794</td>
<td>0.7204</td>
</tr>
<tr>
<td>id.091</td>
<td>Cephalonia</td>
<td>1</td>
<td>3</td>
<td>2007</td>
<td>6</td>
<td>-1</td>
<td>0.954243</td>
<td>-0.96019</td>
<td>0.9861</td>
</tr>
<tr>
<td>id.092</td>
<td>Cephalonia</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>6</td>
<td>250</td>
<td>0.69897</td>
<td>-0.24558</td>
<td>2.8403</td>
</tr>
<tr>
<td>id.093</td>
<td>Central Italy</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>6.2</td>
<td>244</td>
<td>1.255273</td>
<td>-0.87299</td>
<td>2.4118</td>
</tr>
<tr>
<td>id.094</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>246</td>
<td>0.98677</td>
<td>-0.72955</td>
<td>1.6774</td>
</tr>
<tr>
<td>id.095</td>
<td>Central Italy</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>6.2</td>
<td>247</td>
<td>0.863323</td>
<td>-0.28659</td>
<td>3.1011</td>
</tr>
<tr>
<td>id.096</td>
<td>Central Italy</td>
<td>0</td>
<td>1</td>
<td>1701</td>
<td>6.2</td>
<td>-1</td>
<td>0.973128</td>
<td>-0.56431</td>
<td>2.4545</td>
</tr>
<tr>
<td>id.097</td>
<td>Central Italy</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>240</td>
<td>0.845098</td>
<td>-0.97516</td>
<td>0.6353</td>
</tr>
<tr>
<td>id.098</td>
<td>Central Italy</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.2</td>
<td>238</td>
<td>1.176091</td>
<td>-0.88107</td>
<td>1.9728</td>
</tr>
<tr>
<td>id.100</td>
<td>Central Italy</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>6.2</td>
<td>235</td>
<td>1.079181</td>
<td>-0.96738</td>
<td>1.2938</td>
</tr>
<tr>
<td>id.101</td>
<td>Central Italy</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>6.2</td>
<td>233</td>
<td>0.778151</td>
<td>-0.32523</td>
<td>2.8377</td>
</tr>
<tr>
<td>id.102</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>234</td>
<td>0.954243</td>
<td>-0.72102</td>
<td>1.7109</td>
</tr>
<tr>
<td>id.103</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>240</td>
<td>1.623249</td>
<td>-1.45967</td>
<td>1.4583</td>
</tr>
<tr>
<td>id.104</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>239</td>
<td>0.812913</td>
<td>-0.33106</td>
<td>2.7994</td>
</tr>
<tr>
<td>id.106</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>235</td>
<td>0.954243</td>
<td>-0.81987</td>
<td>1.3622</td>
</tr>
<tr>
<td>id.107</td>
<td>Central Italy</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>6.2</td>
<td>235</td>
<td>0.995635</td>
<td>-0.63714</td>
<td>2.0755</td>
</tr>
<tr>
<td>id.112</td>
<td>India</td>
<td>1</td>
<td>11</td>
<td>1995</td>
<td>7.7</td>
<td>-1</td>
<td>1.623249</td>
<td>-1.09313</td>
<td>3.3913</td>
</tr>
<tr>
<td>id.114</td>
<td>India</td>
<td>1</td>
<td>4</td>
<td>-1</td>
<td>6.9</td>
<td>160</td>
<td>1.380211</td>
<td>-1.22695</td>
<td>1.4237</td>
</tr>
<tr>
<td>id.116</td>
<td>India</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>6.9</td>
<td>157</td>
<td>1.255273</td>
<td>-0.98548</td>
<td>1.8613</td>
</tr>
<tr>
<td>id.118</td>
<td>India</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>6.9</td>
<td>222</td>
<td>1.518514</td>
<td>-1.60033</td>
<td>0.8296</td>
</tr>
<tr>
<td>id.120</td>
<td>India</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>6.9</td>
<td>227</td>
<td>1.518514</td>
<td>-1.48545</td>
<td>1.0777</td>
</tr>
<tr>
<td>id.122</td>
<td>India</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>6.9</td>
<td>225</td>
<td>1.053078</td>
<td>-0.95429</td>
<td>1</td>
</tr>
<tr>
<td>id.123</td>
<td>Mexico City</td>
<td>1</td>
<td>14</td>
<td>1962</td>
<td>8.1</td>
<td>48</td>
<td>1.079181</td>
<td>-0.07438</td>
<td>10.1111</td>
</tr>
<tr>
<td>id.129</td>
<td>Turkey</td>
<td>1</td>
<td>3</td>
<td>1999</td>
<td>6.4</td>
<td>20</td>
<td>0.832509</td>
<td>-1.15181</td>
<td>0.4227</td>
</tr>
<tr>
<td>id.133</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>223</td>
<td>0.778151</td>
<td>-0.77806</td>
<td>1</td>
</tr>
<tr>
<td>id.134</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>227</td>
<td>0.845098</td>
<td>-0.81588</td>
<td>0.9169</td>
</tr>
<tr>
<td>id.135</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>225</td>
<td>0.954243</td>
<td>-0.77806</td>
<td>1.5</td>
</tr>
<tr>
<td>id.136</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>224</td>
<td>1.041393</td>
<td>-1.10735</td>
<td>0.859</td>
</tr>
<tr>
<td>id.137</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>60</td>
<td>0.832509</td>
<td>-0.82304</td>
<td>0.9016</td>
</tr>
<tr>
<td>id</td>
<td>Location</td>
<td>n1</td>
<td>n2</td>
<td>n3</td>
<td>magnitude</td>
<td>depth (km)</td>
<td>r s</td>
<td>r f</td>
<td>Ic</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----------</td>
<td>------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>138</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>62</td>
<td>0.778151</td>
<td>-0.80024</td>
<td>0.9501</td>
</tr>
<tr>
<td>139</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>58</td>
<td>0.954243</td>
<td>-1.04191</td>
<td>0.8175</td>
</tr>
<tr>
<td>140</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>63</td>
<td>1</td>
<td>-0.96658</td>
<td>0.9717</td>
</tr>
<tr>
<td>141</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>59</td>
<td>1.477121</td>
<td>-1.21183</td>
<td>1.8409</td>
</tr>
<tr>
<td>142</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>61</td>
<td>0.929419</td>
<td>-0.69789</td>
<td>1.8045</td>
</tr>
<tr>
<td>143</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>387</td>
<td>0.939519</td>
<td>-0.69702</td>
<td>1.8084</td>
</tr>
<tr>
<td>144</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>407</td>
<td>0.792392</td>
<td>-0.52827</td>
<td>1.7778</td>
</tr>
<tr>
<td>145</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>386</td>
<td>0.845098</td>
<td>-0.5583</td>
<td>1.6591</td>
</tr>
<tr>
<td>146</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>390</td>
<td>0.954243</td>
<td>-0.70202</td>
<td>1.7872</td>
</tr>
<tr>
<td>147</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>414</td>
<td>1.079181</td>
<td>-0.88339</td>
<td>1.5701</td>
</tr>
<tr>
<td>148</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>392</td>
<td>0.954243</td>
<td>-0.99654</td>
<td>0.9074</td>
</tr>
<tr>
<td>149</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>395</td>
<td>1.079181</td>
<td>-0.48945</td>
<td>3.8879</td>
</tr>
<tr>
<td>150</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>400</td>
<td>1.477121</td>
<td>-0.87452</td>
<td>4.0055</td>
</tr>
<tr>
<td>151</td>
<td>Ecuador</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>7.8</td>
<td>404</td>
<td>1.322219</td>
<td>-1.25806</td>
<td>1.1584</td>
</tr>
<tr>
<td>152</td>
<td>Ecuador</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>7.8</td>
<td>3</td>
<td>0.954243</td>
<td>-0.92885</td>
<td>1.0601</td>
</tr>
<tr>
<td>153</td>
<td>Ecuador</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>7.8</td>
<td>2</td>
<td>1.176091</td>
<td>-1.31426</td>
<td>0.7281</td>
</tr>
<tr>
<td>154</td>
<td>Ecuador</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>7.8</td>
<td>12</td>
<td>1.176091</td>
<td>-1.23062</td>
<td>0.8814</td>
</tr>
<tr>
<td>155</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>11</td>
<td>1</td>
<td>-0.81361</td>
<td>1.536</td>
</tr>
<tr>
<td>156</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>9</td>
<td>1.079181</td>
<td>-0.84345</td>
<td>1.7203</td>
</tr>
<tr>
<td>157</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>8</td>
<td>0.845098</td>
<td>-0.66254</td>
<td>1.3048</td>
</tr>
<tr>
<td>158</td>
<td>Ecuador</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>7.8</td>
<td>7</td>
<td>1.556303</td>
<td>-1.41341</td>
<td>1.3896</td>
</tr>
<tr>
<td>159</td>
<td>Ecuador</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>7.8</td>
<td>10</td>
<td>1.176091</td>
<td>-1.20482</td>
<td>0.9364</td>
</tr>
<tr>
<td>160</td>
<td>Ecuador</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>7.8</td>
<td>10</td>
<td>1.302111</td>
<td>-1.56067</td>
<td>0.6606</td>
</tr>
<tr>
<td>161</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>16</td>
<td>1.079181</td>
<td>-1.14691</td>
<td>0.8554</td>
</tr>
<tr>
<td>162</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>14</td>
<td>0.954243</td>
<td>-0.74136</td>
<td>1.6327</td>
</tr>
<tr>
<td>163</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>15</td>
<td>0.792392</td>
<td>-0.53417</td>
<td>1.7536</td>
</tr>
<tr>
<td>164</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>17</td>
<td>0.778151</td>
<td>-0.914</td>
<td>0.7312</td>
</tr>
<tr>
<td>165</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>147</td>
<td>1.041393</td>
<td>-1.11805</td>
<td>0.8385</td>
</tr>
<tr>
<td>166</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>153</td>
<td>1.079181</td>
<td>-1.1831</td>
<td>0.7869</td>
</tr>
<tr>
<td>167</td>
<td>Ecuador</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>98</td>
<td>0.653213</td>
<td>-0.78675</td>
<td>0.7354</td>
</tr>
<tr>
<td>168</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>102</td>
<td>0.90309</td>
<td>-0.75995</td>
<td>1.5638</td>
</tr>
<tr>
<td>169</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>1.021189</td>
<td>-0.73212</td>
<td>1.6677</td>
</tr>
<tr>
<td>170</td>
<td>Ecuador</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>101</td>
<td>1</td>
<td>-0.73142</td>
<td>1.6704</td>
</tr>
<tr>
<td>171</td>
<td>Nepal</td>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>65</td>
<td>0.929419</td>
<td>-0.32587</td>
<td>4.25</td>
</tr>
<tr>
<td>172</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>5.7</td>
<td>126</td>
<td>1.556303</td>
<td>-0.91686</td>
<td>4.3608</td>
</tr>
<tr>
<td>173</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>5.7</td>
<td>124</td>
<td>1.278754</td>
<td>-0.78755</td>
<td>2.9361</td>
</tr>
<tr>
<td>174</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>5.7</td>
<td>123</td>
<td>1.176091</td>
<td>-0.99353</td>
<td>1.5219</td>
</tr>
<tr>
<td>175</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>5.7</td>
<td>127</td>
<td>1</td>
<td>-0.83003</td>
<td>1.3311</td>
</tr>
<tr>
<td>176</td>
<td>Mexico City</td>
<td>1</td>
<td>6</td>
<td>-1</td>
<td>5.7</td>
<td>125</td>
<td>0.982271</td>
<td>-0.77989</td>
<td>1.4944</td>
</tr>
<tr>
<td>177</td>
<td>Central Italy</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>6.2</td>
<td>-1</td>
<td>1.477121</td>
<td>-0.82016</td>
<td>4.5396</td>
</tr>
<tr>
<td>178</td>
<td>Central Italy</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>6.2</td>
<td>-1</td>
<td>0.977724</td>
<td>-0.27597</td>
<td>4.7676</td>
</tr>
<tr>
<td>id</td>
<td>Country</td>
<td>Pop.</td>
<td>Magnitude</td>
<td>Depth</td>
<td>Year</td>
<td>Moment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>Central Italy</td>
<td>1</td>
<td>6.2</td>
<td>63</td>
<td>0.963788</td>
<td>-0.47716</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>Central Italy</td>
<td>0</td>
<td>6.2</td>
<td>67</td>
<td>0.653213</td>
<td>-0.1986</td>
<td>2.8486</td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>Central Italy</td>
<td>1</td>
<td>6.2</td>
<td>66</td>
<td>1.033424</td>
<td>-0.94962</td>
<td>1.0104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>Central Italy</td>
<td>0</td>
<td>6.2</td>
<td>65</td>
<td>0.939519</td>
<td>-0.84497</td>
<td>1.2861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>Central Italy</td>
<td>0</td>
<td>6.2</td>
<td>64</td>
<td>1.079181</td>
<td>-1.37986</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>Central Italy</td>
<td>0</td>
<td>6.2</td>
<td>62</td>
<td>0.977724</td>
<td>-0.61618</td>
<td>2.1776</td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>Central Italy</td>
<td>0</td>
<td>6.2</td>
<td>70</td>
<td>0.838849</td>
<td>-0.67778</td>
<td>1.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>6.4</td>
<td>1985</td>
<td>1.623249</td>
<td>-1.18046</td>
<td>2.7707</td>
<td></td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>6.4</td>
<td>1975</td>
<td>1.255273</td>
<td>-0.66696</td>
<td>3.875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>6.4</td>
<td>1975</td>
<td>1.518514</td>
<td>-0.91757</td>
<td>3.9911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1981</td>
<td>0.819544</td>
<td>-0.72677</td>
<td>1.1255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>226</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>1.477121</td>
<td>-0.88874</td>
<td>3.8773</td>
<td></td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.954243</td>
<td>-0.99055</td>
<td>0.9194</td>
<td></td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>Southern Taiwan</td>
<td>0</td>
<td>6.4</td>
<td>1971</td>
<td>0.838849</td>
<td>-0.64187</td>
<td>1.3685</td>
<td></td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1984</td>
<td>1.079181</td>
<td>-1.57349</td>
<td>0.3207</td>
<td></td>
<td></td>
</tr>
<tr>
<td>234</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1984</td>
<td>1.518514</td>
<td>-1.57025</td>
<td>0.8881</td>
<td></td>
<td></td>
</tr>
<tr>
<td>235</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1984</td>
<td>1.380211</td>
<td>-1.40012</td>
<td>0.955</td>
<td></td>
<td></td>
</tr>
<tr>
<td>236</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1984</td>
<td>1.079181</td>
<td>-1.57349</td>
<td>0.3207</td>
<td></td>
<td></td>
</tr>
<tr>
<td>238</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1985</td>
<td>0.845098</td>
<td>-0.21439</td>
<td>3.6622</td>
<td></td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1984</td>
<td>1.079181</td>
<td>-0.78068</td>
<td>1.9881</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.653213</td>
<td>-0.21424</td>
<td>3.6636</td>
<td></td>
<td></td>
</tr>
<tr>
<td>241</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.954243</td>
<td>-0.50626</td>
<td>2.8056</td>
<td></td>
<td></td>
</tr>
<tr>
<td>242</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>1.079181</td>
<td>-1.26841</td>
<td>0.6466</td>
<td></td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.698979</td>
<td>-0.98047</td>
<td>0.6278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>244</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.954243</td>
<td>-1.098</td>
<td>0.7186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>248</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.778151</td>
<td>-0.969</td>
<td>0.6445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>249</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>1.612784</td>
<td>-1.75203</td>
<td>0.7269</td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.832509</td>
<td>-0.67882</td>
<td>1.2571</td>
<td></td>
<td></td>
</tr>
<tr>
<td>251</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.954243</td>
<td>-0.78383</td>
<td>1.4803</td>
<td></td>
<td></td>
</tr>
<tr>
<td>254</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.903099</td>
<td>-0.82681</td>
<td>1.3412</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>Southern Taiwan</td>
<td>1</td>
<td>6.4</td>
<td>1982</td>
<td>0.954243</td>
<td>-1.28483</td>
<td>0.4675</td>
<td></td>
<td></td>
</tr>
<tr>
<td>257</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>100</td>
<td>1.113943</td>
<td>-1.00789</td>
<td>1.1786</td>
<td></td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>100</td>
<td>0.954243</td>
<td>-0.44491</td>
<td>3.2308</td>
<td></td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>90</td>
<td>1.623249</td>
<td>-1.10458</td>
<td>3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>85</td>
<td>0.812913</td>
<td>-0.26098</td>
<td>3.2897</td>
<td></td>
<td></td>
</tr>
<tr>
<td>261</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>110</td>
<td>0.778151</td>
<td>-0.56511</td>
<td>1.6331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>262</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>110</td>
<td>0.78533</td>
<td>-0.40938</td>
<td>2.3377</td>
<td></td>
<td></td>
</tr>
<tr>
<td>263</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>95</td>
<td>0.929419</td>
<td>-0.61943</td>
<td>2.0417</td>
<td></td>
<td></td>
</tr>
<tr>
<td>264</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>121</td>
<td>0.812913</td>
<td>-0.14291</td>
<td>4.3178</td>
<td></td>
<td></td>
</tr>
<tr>
<td>266</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>118</td>
<td>0.778151</td>
<td>-0.28997</td>
<td>3.0771</td>
<td></td>
<td></td>
</tr>
<tr>
<td>267</td>
<td>Nepal</td>
<td>0</td>
<td>7.8</td>
<td>123</td>
<td>0.954243</td>
<td>-0.42899</td>
<td>3.3517</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id.268</td>
<td>Nepal</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>117</td>
<td>0.778151</td>
<td>-0.68825</td>
<td>1.2303</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>id.269</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>122</td>
<td>1.041393</td>
<td>-0.98843</td>
<td>1.1298</td>
</tr>
<tr>
<td>id.270</td>
<td>Nepal</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>119</td>
<td>0.845098</td>
<td>-0.41976</td>
<td>2.2823</td>
</tr>
<tr>
<td>id.271</td>
<td>Nepal</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>120</td>
<td>0.778151</td>
<td>-0.99654</td>
<td>0.6048</td>
</tr>
<tr>
<td>id.272</td>
<td>Nepal</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>7.8</td>
<td>98</td>
<td>0.778151</td>
<td>-0.59912</td>
<td>1.5101</td>
</tr>
<tr>
<td>id.273</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>102</td>
<td>1.060698</td>
<td>-0.94923</td>
<td>1.0118</td>
</tr>
<tr>
<td>id.274</td>
<td>Nepal</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>7.8</td>
<td>99</td>
<td>1.041393</td>
<td>-0.98843</td>
<td>1.6747</td>
</tr>
<tr>
<td>id.275</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>101</td>
<td>0.60206</td>
<td>-0.15889</td>
<td>2.0809</td>
</tr>
<tr>
<td>id.276</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>97</td>
<td>0.763428</td>
<td>-0.39147</td>
<td>2.436</td>
</tr>
<tr>
<td>id.277</td>
<td>Nepal</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>1.255273</td>
<td>-1.27165</td>
<td>0.963</td>
</tr>
<tr>
<td>id.278</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>103</td>
<td>0.799341</td>
<td>-0.67695</td>
<td>1.2627</td>
</tr>
<tr>
<td>id.279</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.280</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.281</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.282</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.283</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.284</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.285</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.286</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.287</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.288</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.289</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.290</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.291</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.292</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.293</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.294</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.295</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.296</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.297</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.298</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.299</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.300</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.301</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
<tr>
<td>id.302</td>
<td>Nepal</td>
<td>0</td>
<td>3</td>
<td>-1</td>
<td>7.8</td>
<td>100</td>
<td>0.799341</td>
<td>-0.40949</td>
<td>2.3369</td>
</tr>
</tbody>
</table>

### 8.2 Extension of debris evaluation

Note: A missing data is replaced with the value -1.

The meaning of the different parameters is well described in the paragraph 2.3.1.

The reference column shows the object or the dimension that is taken as comparison (with well-known measure) in the proportion (1.1) to evaluate the extent of the debris.
## Appendix

<table>
<thead>
<tr>
<th>ID</th>
<th>Y1</th>
<th>Y2</th>
<th>y1</th>
<th>y2</th>
<th>D [m]</th>
<th>P</th>
<th>p</th>
<th>d [m]</th>
<th>EOD norm.</th>
<th>Reference</th>
<th>Coordinate Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>id.001</td>
<td>301</td>
<td>312</td>
<td>299</td>
<td>313</td>
<td>1.5</td>
<td>11</td>
<td>14</td>
<td>1.9090</td>
<td>0.31818</td>
<td>Sidewalk</td>
<td>y</td>
</tr>
<tr>
<td>id.002</td>
<td>95</td>
<td>153</td>
<td>152</td>
<td>164</td>
<td>36</td>
<td>58</td>
<td>12</td>
<td>7.4482</td>
<td>1.24138</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.003</td>
<td>280</td>
<td>289</td>
<td>285</td>
<td>297</td>
<td>1.5</td>
<td>9</td>
<td>12</td>
<td>2</td>
<td>0.33333</td>
<td>Sidewalk</td>
<td>y</td>
</tr>
<tr>
<td>id.004</td>
<td>12</td>
<td>304</td>
<td>169</td>
<td>307</td>
<td>6</td>
<td>292</td>
<td>138</td>
<td>2.8356</td>
<td>0.15753</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.006</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3.5</td>
<td>0.38889</td>
<td>Sidewalk</td>
<td>y</td>
</tr>
<tr>
<td>id.007</td>
<td>162</td>
<td>278</td>
<td>163</td>
<td>213</td>
<td>4.5</td>
<td>116</td>
<td>50</td>
<td>1.9396</td>
<td>0.21552</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.009</td>
<td>415</td>
<td>451</td>
<td>282</td>
<td>305</td>
<td>4</td>
<td>36</td>
<td>23</td>
<td>2.5555</td>
<td>0.42593</td>
<td>Wall</td>
<td>x</td>
</tr>
<tr>
<td>id.010</td>
<td>14</td>
<td>295</td>
<td>95</td>
<td>164</td>
<td>15</td>
<td>281</td>
<td>69</td>
<td>3.6832</td>
<td>0.20463</td>
<td>Height</td>
<td>x</td>
</tr>
<tr>
<td>id.011</td>
<td>74</td>
<td>312</td>
<td>240</td>
<td>287</td>
<td>12</td>
<td>238</td>
<td>47</td>
<td>2.3697</td>
<td>0.26331</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.012</td>
<td>78</td>
<td>210</td>
<td>239</td>
<td>263</td>
<td>36</td>
<td>132</td>
<td>24</td>
<td>6.5454</td>
<td>0.72727</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.013</td>
<td>333</td>
<td>377</td>
<td>326</td>
<td>357</td>
<td>1.5</td>
<td>44</td>
<td>31</td>
<td>1.0568</td>
<td>0.11742</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.014</td>
<td>172</td>
<td>269</td>
<td>296</td>
<td>320</td>
<td>5</td>
<td>97</td>
<td>24</td>
<td>1.2371</td>
<td>0.13746</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.015</td>
<td>380</td>
<td>410</td>
<td>401</td>
<td>412</td>
<td>1.5</td>
<td>30</td>
<td>11</td>
<td>0.55</td>
<td>0.03056</td>
<td>Sidewalk</td>
<td>y</td>
</tr>
<tr>
<td>id.016</td>
<td>138</td>
<td>198</td>
<td>360</td>
<td>449</td>
<td>4</td>
<td>60</td>
<td>89</td>
<td>5.9333</td>
<td>0.65926</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id.017</td>
<td>376</td>
<td>484</td>
<td>227</td>
<td>429</td>
<td>1.5</td>
<td>108</td>
<td>202</td>
<td>2.8055</td>
<td>0.15586</td>
<td>Sidewalk</td>
<td>x</td>
</tr>
<tr>
<td>id.019</td>
<td>212</td>
<td>302</td>
<td>121</td>
<td>143</td>
<td>4</td>
<td>90</td>
<td>22</td>
<td>0.9777</td>
<td>0.10864</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.020</td>
<td>150</td>
<td>209</td>
<td>290</td>
<td>345</td>
<td>1.8</td>
<td>59</td>
<td>55</td>
<td>1.6779</td>
<td>0.11186</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id.021</td>
<td>543</td>
<td>594</td>
<td>366</td>
<td>428</td>
<td>1.5</td>
<td>51</td>
<td>62</td>
<td>1.8235</td>
<td>0.30392</td>
<td>Sidewalk</td>
<td>x</td>
</tr>
<tr>
<td>id.022</td>
<td>1243</td>
<td>1555</td>
<td>1870</td>
<td>2063</td>
<td>2.7</td>
<td>312</td>
<td>193</td>
<td>1.6701</td>
<td>0.27837</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id.023</td>
<td>1221</td>
<td>1550</td>
<td>1645</td>
<td>1816</td>
<td>2.7</td>
<td>329</td>
<td>171</td>
<td>1.4033</td>
<td>0.23389</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id.024</td>
<td>45</td>
<td>203</td>
<td>283</td>
<td>311</td>
<td>30</td>
<td>158</td>
<td>28</td>
<td>5.3164</td>
<td>0.59072</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.025</td>
<td>143</td>
<td>239</td>
<td>41</td>
<td>57</td>
<td>30</td>
<td>96</td>
<td>16</td>
<td>5</td>
<td>0.55556</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.026</td>
<td>21</td>
<td>176</td>
<td>183</td>
<td>260</td>
<td>30</td>
<td>155</td>
<td>77</td>
<td>14.903</td>
<td>0.82796</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.028</td>
<td>416</td>
<td>534</td>
<td>1332</td>
<td>1407</td>
<td>3</td>
<td>118</td>
<td>75</td>
<td>1.9067</td>
<td>0.3178</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id.029</td>
<td>40</td>
<td>147</td>
<td>326</td>
<td>405</td>
<td>1.2</td>
<td>107</td>
<td>79</td>
<td>0.8859</td>
<td>0.14766</td>
<td>Window</td>
<td>y</td>
</tr>
<tr>
<td>id.030</td>
<td>354</td>
<td>453</td>
<td>993</td>
<td>1034</td>
<td>3</td>
<td>99</td>
<td>41</td>
<td>1.2424</td>
<td>0.13805</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.032</td>
<td>168</td>
<td>260</td>
<td>152</td>
<td>182</td>
<td>4</td>
<td>92</td>
<td>30</td>
<td>1.3043</td>
<td>0.14493</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.033</td>
<td>44</td>
<td>268</td>
<td>173</td>
<td>215</td>
<td>4</td>
<td>224</td>
<td>42</td>
<td>0.75</td>
<td>0.08333</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.035</td>
<td>200</td>
<td>300</td>
<td>70</td>
<td>129</td>
<td>3</td>
<td>100</td>
<td>59</td>
<td>1.77</td>
<td>0.118</td>
<td>Lamp</td>
<td>y</td>
</tr>
<tr>
<td>id.036</td>
<td>50</td>
<td>225</td>
<td>146</td>
<td>212</td>
<td>6</td>
<td>175</td>
<td>66</td>
<td>2.2628</td>
<td>0.25143</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.039</td>
<td>155</td>
<td>225</td>
<td>49</td>
<td>132</td>
<td>2.2</td>
<td>70</td>
<td>83</td>
<td>2.6085</td>
<td>0.1739</td>
<td>Signal</td>
<td>y</td>
</tr>
<tr>
<td>id.040</td>
<td>71</td>
<td>157</td>
<td>201</td>
<td>236</td>
<td>4</td>
<td>86</td>
<td>35</td>
<td>1.6279</td>
<td>0.18088</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.043</td>
<td>88</td>
<td>148</td>
<td>329</td>
<td>348</td>
<td>3</td>
<td>60</td>
<td>19</td>
<td>0.95</td>
<td>0.15833</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.044</td>
<td>51</td>
<td>191</td>
<td>151</td>
<td>193</td>
<td>3</td>
<td>140</td>
<td>42</td>
<td>0.9</td>
<td>0.1</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id.045</td>
<td>173</td>
<td>245</td>
<td>140</td>
<td>186</td>
<td>4</td>
<td>72</td>
<td>46</td>
<td>2.5555</td>
<td>0.28395</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.046</td>
<td>44</td>
<td>191</td>
<td>233</td>
<td>329</td>
<td>4</td>
<td>147</td>
<td>96</td>
<td>2.6122</td>
<td>0.29025</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.050</td>
<td>216</td>
<td>368</td>
<td>199</td>
<td>413</td>
<td>4</td>
<td>152</td>
<td>214</td>
<td>5.6315</td>
<td>0.62573</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id.051</td>
<td>304</td>
<td>368</td>
<td>292</td>
<td>348</td>
<td>1.5</td>
<td>64</td>
<td>56</td>
<td>1.3125</td>
<td>0.21875</td>
<td>Sidewalk</td>
<td>x</td>
</tr>
<tr>
<td>id.053</td>
<td>247</td>
<td>279</td>
<td>174</td>
<td>257</td>
<td>1.2</td>
<td>32</td>
<td>83</td>
<td>3.1125</td>
<td>0.51875</td>
<td>Window</td>
<td>y</td>
</tr>
<tr>
<td>id.054</td>
<td>26</td>
<td>179</td>
<td>0</td>
<td>418</td>
<td>1.2</td>
<td>153</td>
<td>418</td>
<td>3.2784</td>
<td>0.65569</td>
<td>Window</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>404</td>
<td>849</td>
<td>78</td>
<td>265</td>
<td>6445</td>
<td>187</td>
<td>2.5213</td>
<td>0.14007</td>
<td>Height</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>237</td>
<td>882</td>
<td>187</td>
<td>436</td>
<td>645</td>
<td>249</td>
<td>2.3162</td>
<td>0.25736</td>
<td>Height</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>156</td>
<td>298</td>
<td>30</td>
<td>60</td>
<td>142</td>
<td>30</td>
<td>1.2676</td>
<td>0.08451</td>
<td>Height</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>84</td>
<td>163</td>
<td>251</td>
<td>369</td>
<td>1.8</td>
<td>79</td>
<td>118</td>
<td>2.6886</td>
<td>0.29873</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>876</td>
<td>933</td>
<td>598</td>
<td>770</td>
<td>1.5</td>
<td>57</td>
<td>172</td>
<td>4.5263</td>
<td>0.75439</td>
<td>Sidewalk</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>334</td>
<td>502</td>
<td>371</td>
<td>720</td>
<td>1.8</td>
<td>168</td>
<td>349</td>
<td>3.7392</td>
<td>0.41548</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>462</td>
<td>436</td>
<td>0</td>
<td>704</td>
<td>2</td>
<td>388</td>
<td>704</td>
<td>3.6288</td>
<td>0.40321</td>
<td>Signal</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>175</td>
<td>282</td>
<td>219</td>
<td>549</td>
<td>1.8</td>
<td>107</td>
<td>330</td>
<td>5.5514</td>
<td>0.30841</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>465</td>
<td>544</td>
<td>287</td>
<td>481</td>
<td>6</td>
<td>79</td>
<td>194</td>
<td>14.734</td>
<td>0.44649</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>271</td>
<td>351</td>
<td>149</td>
<td>362</td>
<td>1.2</td>
<td>80</td>
<td>213</td>
<td>3.195</td>
<td>0.5325</td>
<td>Rail</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>44</td>
<td>159</td>
<td>222</td>
<td>269</td>
<td>9</td>
<td>115</td>
<td>47</td>
<td>3.6782</td>
<td>0.61304</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>214</td>
<td>508</td>
<td>490</td>
<td>589</td>
<td>61</td>
<td>294</td>
<td>99</td>
<td>20.540</td>
<td>2.28231</td>
<td>Lenght</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>219</td>
<td>342</td>
<td>455</td>
<td>540</td>
<td>3.5</td>
<td>123</td>
<td>85</td>
<td>2.4186</td>
<td>0.26874</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>5</td>
<td>284</td>
<td>315</td>
<td>438</td>
<td>6</td>
<td>279</td>
<td>123</td>
<td>2.6451</td>
<td>0.44086</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>280</td>
<td>367</td>
<td>239</td>
<td>524</td>
<td>1.8</td>
<td>87</td>
<td>285</td>
<td>5.8965</td>
<td>0.98276</td>
<td>Car</td>
<td>x</td>
</tr>
<tr>
<td>id</td>
<td>217</td>
<td>328</td>
<td>171</td>
<td>333</td>
<td>3</td>
<td>111</td>
<td>162</td>
<td>4.3783</td>
<td>0.48649</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>187</td>
<td>260</td>
<td>174</td>
<td>260</td>
<td>3</td>
<td>73</td>
<td>86</td>
<td>3.5342</td>
<td>0.19635</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>99</td>
<td>318</td>
<td>206</td>
<td>246</td>
<td>3</td>
<td>219</td>
<td>40</td>
<td>0.5479</td>
<td>0.09132</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>213</td>
<td>341</td>
<td>227</td>
<td>262</td>
<td>3</td>
<td>128</td>
<td>35</td>
<td>0.8203</td>
<td>0.09115</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>26</td>
<td>314</td>
<td>351</td>
<td>415</td>
<td>3</td>
<td>288</td>
<td>64</td>
<td>0.6666</td>
<td>0.03704</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>49</td>
<td>326</td>
<td>309</td>
<td>382</td>
<td>3</td>
<td>304</td>
<td>73</td>
<td>0.7203</td>
<td>0.04002</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>51</td>
<td>483</td>
<td>1</td>
<td>143</td>
<td>3</td>
<td>432</td>
<td>142</td>
<td>0.9861</td>
<td>0.10957</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>35</td>
<td>392</td>
<td>421</td>
<td>590</td>
<td>6</td>
<td>357</td>
<td>169</td>
<td>2.8403</td>
<td>0.56807</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>191</td>
<td>395</td>
<td>22</td>
<td>186</td>
<td>3</td>
<td>204</td>
<td>164</td>
<td>2.4117</td>
<td>0.13399</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>352</td>
<td>445</td>
<td>651</td>
<td>677</td>
<td>6</td>
<td>93</td>
<td>26</td>
<td>1.6774</td>
<td>0.18638</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>105</td>
<td>283</td>
<td>52</td>
<td>144</td>
<td>6</td>
<td>178</td>
<td>92</td>
<td>3.1011</td>
<td>0.51685</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>167</td>
<td>277</td>
<td>362</td>
<td>407</td>
<td>6</td>
<td>110</td>
<td>45</td>
<td>2.4545</td>
<td>0.27273</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>299</td>
<td>367</td>
<td>436</td>
<td>472</td>
<td>1.2</td>
<td>68</td>
<td>36</td>
<td>0.6352</td>
<td>0.10588</td>
<td>Window</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>201</td>
<td>385</td>
<td>179</td>
<td>300</td>
<td>3</td>
<td>184</td>
<td>121</td>
<td>1.9728</td>
<td>0.13152</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>417</td>
<td>481</td>
<td>699</td>
<td>735</td>
<td>2.3</td>
<td>64</td>
<td>36</td>
<td>1.2937</td>
<td>0.10781</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>276</td>
<td>353</td>
<td>106</td>
<td>201</td>
<td>2.3</td>
<td>77</td>
<td>95</td>
<td>2.8376</td>
<td>0.47294</td>
<td>Door</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>1</td>
<td>769</td>
<td>184</td>
<td>330</td>
<td>9</td>
<td>768</td>
<td>146</td>
<td>1.7109</td>
<td>0.1901</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>355</td>
<td>643</td>
<td>364</td>
<td>434</td>
<td>6</td>
<td>288</td>
<td>70</td>
<td>1.4583</td>
<td>0.03472</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>104</td>
<td>821</td>
<td>198</td>
<td>533</td>
<td>6</td>
<td>718</td>
<td>335</td>
<td>2.7994</td>
<td>0.46657</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>218</td>
<td>329</td>
<td>73</td>
<td>157</td>
<td>1.8</td>
<td>111</td>
<td>84</td>
<td>1.3622</td>
<td>0.15135</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>224</td>
<td>383</td>
<td>308</td>
<td>418</td>
<td>3</td>
<td>159</td>
<td>110</td>
<td>2.0754</td>
<td>0.23061</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>366</td>
<td>435</td>
<td>240</td>
<td>318</td>
<td>3</td>
<td>69</td>
<td>78</td>
<td>3.3913</td>
<td>0.08075</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>316</td>
<td>375</td>
<td>373</td>
<td>401</td>
<td>3</td>
<td>59</td>
<td>28</td>
<td>1.4237</td>
<td>0.05932</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>180</td>
<td>317</td>
<td>146</td>
<td>231</td>
<td>3</td>
<td>137</td>
<td>85</td>
<td>1.8613</td>
<td>0.10341</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>455</td>
<td>590</td>
<td>147</td>
<td>203</td>
<td>2</td>
<td>135</td>
<td>56</td>
<td>0.8296</td>
<td>0.02514</td>
<td>Signal</td>
<td>y</td>
</tr>
<tr>
<td>id</td>
<td>472</td>
<td>845</td>
<td>644</td>
<td>845</td>
<td>2</td>
<td>373</td>
<td>201</td>
<td>1.0777</td>
<td>0.03266</td>
<td>Signal</td>
<td>y</td>
</tr>
</tbody>
</table>
### Appendix

<table>
<thead>
<tr>
<th>id</th>
<th>218</th>
<th>323</th>
<th>34</th>
<th>69</th>
<th>105</th>
<th>35</th>
<th>1</th>
<th>0.11111</th>
<th>Signal</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>id.122</td>
<td>447</td>
<td>474</td>
<td>329</td>
<td>420</td>
<td>27</td>
<td>91</td>
<td>10.1111</td>
<td>0.84259</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.123</td>
<td>648</td>
<td>868</td>
<td>478</td>
<td>509</td>
<td>220</td>
<td>31</td>
<td>0.4227</td>
<td>0.07045</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.129</td>
<td>1712</td>
<td>3368</td>
<td>864</td>
<td>1416</td>
<td>1656</td>
<td>552</td>
<td>1</td>
<td>0.16667</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.133</td>
<td>1796</td>
<td>3432</td>
<td>976</td>
<td>1476</td>
<td>1636</td>
<td>500</td>
<td>0.9168</td>
<td>0.15281</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.134</td>
<td>2276</td>
<td>3340</td>
<td>3380</td>
<td>3912</td>
<td>1064</td>
<td>532</td>
<td>1.5</td>
<td>0.16667</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.135</td>
<td>1948</td>
<td>3764</td>
<td>1900</td>
<td>2420</td>
<td>1816</td>
<td>520</td>
<td>0.8590</td>
<td>0.07809</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.136</td>
<td>1039</td>
<td>2969</td>
<td>1859</td>
<td>2439</td>
<td>1930</td>
<td>580</td>
<td>0.9015</td>
<td>0.15026</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.137</td>
<td>1024</td>
<td>2388</td>
<td>1708</td>
<td>2140</td>
<td>1364</td>
<td>432</td>
<td>0.9501</td>
<td>0.15836</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.138</td>
<td>1346</td>
<td>2168</td>
<td>1102</td>
<td>1326</td>
<td>822</td>
<td>224</td>
<td>0.8175</td>
<td>0.09084</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.139</td>
<td>1225</td>
<td>2213</td>
<td>3139</td>
<td>3459</td>
<td>988</td>
<td>320</td>
<td>0.9716</td>
<td>0.10796</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.140</td>
<td>2068</td>
<td>2420</td>
<td>1952</td>
<td>2168</td>
<td>352</td>
<td>216</td>
<td>1.8409</td>
<td>0.06136</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.141</td>
<td>481</td>
<td>2230</td>
<td>3158</td>
<td>4210</td>
<td>1749</td>
<td>1052</td>
<td>1.8044</td>
<td>0.2005</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.142</td>
<td>1352</td>
<td>2208</td>
<td>36</td>
<td>552</td>
<td>856</td>
<td>516</td>
<td>1.8084</td>
<td>0.20093</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.143</td>
<td>2088</td>
<td>2844</td>
<td>2780</td>
<td>3228</td>
<td>756</td>
<td>448</td>
<td>1.7778</td>
<td>0.2963</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.144</td>
<td>1396</td>
<td>2452</td>
<td>1868</td>
<td>2452</td>
<td>1056</td>
<td>584</td>
<td>1.6590</td>
<td>0.27652</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.145</td>
<td>703</td>
<td>2338</td>
<td>278</td>
<td>1252</td>
<td>1635</td>
<td>974</td>
<td>1.7871</td>
<td>0.19857</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.146</td>
<td>1870</td>
<td>2426</td>
<td>2528</td>
<td>2819</td>
<td>556</td>
<td>291</td>
<td>1.5701</td>
<td>0.13085</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.147</td>
<td>1664</td>
<td>2312</td>
<td>2038</td>
<td>2234</td>
<td>648</td>
<td>196</td>
<td>0.9074</td>
<td>0.10082</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.148</td>
<td>1956</td>
<td>2402</td>
<td>730</td>
<td>1308</td>
<td>446</td>
<td>578</td>
<td>3.8878</td>
<td>0.32399</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.149</td>
<td>2069</td>
<td>2615</td>
<td>388</td>
<td>1177</td>
<td>546</td>
<td>729</td>
<td>4.0054</td>
<td>0.13352</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.150</td>
<td>4052</td>
<td>5744</td>
<td>724</td>
<td>1284</td>
<td>1692</td>
<td>560</td>
<td>1.1583</td>
<td>0.05516</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.151</td>
<td>1692</td>
<td>3488</td>
<td>4572</td>
<td>5116</td>
<td>1796</td>
<td>544</td>
<td>1.0601</td>
<td>0.11779</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.152</td>
<td>1621</td>
<td>2684</td>
<td>1279</td>
<td>1537</td>
<td>1063</td>
<td>258</td>
<td>0.7281</td>
<td>0.04854</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.153</td>
<td>2552</td>
<td>3117</td>
<td>1903</td>
<td>2069</td>
<td>565</td>
<td>166</td>
<td>0.8814</td>
<td>0.05876</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.154</td>
<td>1988</td>
<td>2488</td>
<td>1508</td>
<td>1764</td>
<td>500</td>
<td>256</td>
<td>1.536</td>
<td>0.1536</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.155</td>
<td>3460</td>
<td>4604</td>
<td>632</td>
<td>1288</td>
<td>1144</td>
<td>656</td>
<td>1.7202</td>
<td>0.14336</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.156</td>
<td>1469</td>
<td>4389</td>
<td>2179</td>
<td>3449</td>
<td>2920</td>
<td>1270</td>
<td>1.3047</td>
<td>0.21747</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.157</td>
<td>4312</td>
<td>5780</td>
<td>3224</td>
<td>3904</td>
<td>1468</td>
<td>680</td>
<td>1.3896</td>
<td>0.0386</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.158</td>
<td>3960</td>
<td>5344</td>
<td>2972</td>
<td>3404</td>
<td>1384</td>
<td>432</td>
<td>0.9364</td>
<td>0.06243</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.159</td>
<td>4464</td>
<td>5572</td>
<td>1596</td>
<td>1840</td>
<td>1108</td>
<td>244</td>
<td>0.6606</td>
<td>0.02753</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.160</td>
<td>2892</td>
<td>3888</td>
<td>8</td>
<td>292</td>
<td>996</td>
<td>284</td>
<td>0.8554</td>
<td>0.07129</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.161</td>
<td>1688</td>
<td>2276</td>
<td>1760</td>
<td>2080</td>
<td>588</td>
<td>320</td>
<td>1.63263</td>
<td>0.18141</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.162</td>
<td>1273</td>
<td>2178</td>
<td>2269</td>
<td>2798</td>
<td>905</td>
<td>529</td>
<td>1.753591</td>
<td>0.29227</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.163</td>
<td>1498</td>
<td>2294</td>
<td>2164</td>
<td>2358</td>
<td>796</td>
<td>194</td>
<td>0.731156</td>
<td>0.12186</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.164</td>
<td>1944</td>
<td>3808</td>
<td>1112</td>
<td>1633</td>
<td>1864</td>
<td>521</td>
<td>0.838519</td>
<td>0.07623</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.165</td>
<td>2244</td>
<td>3686</td>
<td>460</td>
<td>886</td>
<td>1624</td>
<td>426</td>
<td>0.786946</td>
<td>0.06558</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.166</td>
<td>1448</td>
<td>2476</td>
<td>1676</td>
<td>1928</td>
<td>1028</td>
<td>252</td>
<td>0.735409</td>
<td>0.16342</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.167</td>
<td>599</td>
<td>2481</td>
<td>3123</td>
<td>3450</td>
<td>1882</td>
<td>327</td>
<td>1.563762</td>
<td>0.17375</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.168</td>
<td>644</td>
<td>2603</td>
<td>2576</td>
<td>2739</td>
<td>1959</td>
<td>363</td>
<td>1.667688</td>
<td>0.1853</td>
<td>Height</td>
<td>y</td>
</tr>
<tr>
<td>id.169</td>
<td>564</td>
<td>2509</td>
<td>2670</td>
<td>3031</td>
<td>1945</td>
<td>361</td>
<td>1.670437</td>
<td>0.1856</td>
<td>Height</td>
<td>y</td>
</tr>
</tbody>
</table>
| ID  | Height | y | x | Car | x | License plate | Car | x | Car | y | x | Car | y | x | Car | y | x | Door | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Sidewalk | y | x | Side...
8.3 Data visualization script

#import modules in python
import matplotlib.pyplot as plt
import numpy as np

#load dataset
matrix = np.loadtxt('Database_pictures_python_multi.csv',
    usecols=(2,3,4,5,6,7,8),
    delimiter=';', skiprows=1)
index_list = [1,2,3,4,5]

#define graphs properties
titles = ['Stories', 'Year', 'Magnitude', 'Distance from
epicenter', 'Height']
options = [0]
for i in range(len(index_list)):
    f, axs = plt.subplots(1,1,figsize=(10,5))
    plt.title(titles[i])
    plt.ylabel('Extension of debris')
    plt.xlabel(titles[i])
    good_rows_masonry = np.logical_and(matrix[:,
        index_list[i]]>=0,
        matrix[:, 0]==0)
    good_rows_concrete = np.logical_and(matrix[:,
        index_list[i]]>=0,
        matrix[:, 0]==1)
    plt.scatter(matrix[good_rows_masonry, index_list[i]],
        matrix[good_rows_masonry, 6],
        marker = 'v', label='masonry')
    plt.scatter(matrix[good_rows_concrete, index_list[i]],
        matrix[good_rows_concrete, 6],
        marker = 'o', label='concrete')
    plt.legend(loc=1)
import modules in python
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.neighbors import KNeighborsRegressor as KNR
from sklearn.preprocessing import MinMaxScaler
from sklearn import svm
from sklearn.tree import DecisionTreeRegressor
import sklearn
from sklearn.ensemble import RandomForestRegressor
from sklearn import linear_model
from sklearn.linear_model import ElasticNet
from sklearn.linear_model import Ridge
from sklearn.svm import SVR
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.neural_network import MLPRegressor
import seaborn as sns
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error

# import database from folder
db = np.loadtxt('Database_pictures_python_multi.csv',
usecols=(2,3,4,5,6,7,8),
       delimiter = ';', skiprows=1)

index_list = [1,3,4,5]
X = db[np.sum(db[:,index_list],1) > 0,:][:,index_list]

8.4 Training and test script
y = db[:,6]
y = y.reshape(-1,1)

#Preprocessing
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
#X_scaled = preprocessing.scale(X)
#min_max_scaler = preprocessing.MinMaxScaler()
#X_scaled = min_max_scaler.fit_transform(X)
#X_embedded = TSNE(n_components=2).fit_transform(X_scaled)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.33, shuffle=True, random_state=0)

#run algorithms
clf = KNR(n_neighbors=5)
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("KNR")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = linear_model.Lasso(alpha=0.01, max_iter=10)
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("Lasso")

79
from sklearn.linear_model import ElasticNet, Ridge
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error

X_train, y_train = load_data('train')
X_test, y_test = load_data('test')

clf = ElasticNet(alpha=0.01)
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("Elastic Net")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = Ridge(alpha=0.1)
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("Ridge")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = SVR(C=0.1, epsilon=0.001, kernel='rbf')
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("SVR")
Appendix

```python
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = RandomForestRegressor(max_depth=10, n_estimators=20, min_samples_split=40)
clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("random forest")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = tree.DecisionTreeRegressor(min_samples_split=25, min_samples_leaf=10, presort=True)
clf = clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("Tree")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))

clf = MLPRegressor(alpha=0.1, batch_size=10, hidden_layer_sizes=1000,
                   learning_rate='constant',
                   learning_rate_init=0.001, solver='adam',
```

81
activation='relu', random_state=27)
clf = clf.fit(X_train, y_train)
y_predict = clf.predict(X_test)
MARD = mean_absolute_error(y_predict, y_test)
print("MLP Regressor")
print("Train set R^2: {:.2f}".format(clf.score(X_train, y_train)))
print("Test set R^2: {:.2f}".format(clf.score(X_test, y_test)))
print("Mean absolute error {:.2f}".format(MARD))
...
#producing charts
X_embedded = TSNE(n_components=2, random_state=0).fit_transform(X_scaled)
g = sns.scatterplot(X_embedded[:,0], X_embedded[:,1])
g.set(xticks=[])  
g.set(yticks=[])  
g.set_title('t-SNE data visualization')
plt.show()  
#sns.scatterplot(X_embedded[:,0], X_embedded[:,1], hue=y[:,0])

8.5 Results visualization script

import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

mpl.style.use('default')
x = np.arange(3)
plt.bar(x, height= [0.29,0.27,0.26], color='b')
plt.xticks(x, ['5','4','3'])
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('KNR-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
#plt.axes().spines['right'].set_visible(False)
#plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
plt.axes().axhline(linewidth=0, color="k")
plt.tick_params(axis='x',          # changes apply to the x-axis
                which='both',        # both major and minor ticks are
                bottom=False,      # ticks along the bottom edge are off
                top=False,         # ticks along the top edge are off
                labelbottom=True)  # labels along the bottom edge are off
plt.show()

x = np.arange(3)
plt.bar(x, height= [0.33,0.31,0.31], color='b')
plt.xticks(x, ['5','4','3'])

plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('Lasso-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
#plt.axes().spines['right'].set_visible(False)
#plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
```python
plt.axes().axhline(linewidth=0, color="k")
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=True)  # labels along the bottom edge are off
plt.show()

x = np.arange(3)
plt.bar(x, height=[0.33, 0.31, 0.31], color='b')
plt.xticks(x, ['5', '4', '3'])
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('Elastic Net-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
# plt.axes().spines['right'].set_visible(False)
# plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
plt.axes().axhline(linewidth=0, color="k")
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=True)  # labels along the bottom edge are off
```
```python
plt.show()

x = np.arange(3)
plt.bar(x, height= [0.33, 0.31, 0.31], color='b')
plt.xticks(x, ['5','4','3'])
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('Ridge-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
#plt.axes().spines['right'].set_visible(False)
#plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
plt.axes().axhline(linewidth=0, color='k')
plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=True)
plt.show()

x = np.arange(3)
plt.bar(x, height= [0.27, 0.26, 0.26], color='b')
plt.xticks(x, ['5','4','3'])
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('SVR-MARD')
```
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('Random forest-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
```python
axis='x',          # changes apply to the x-axis
which='both',      # both major and minor ticks are
bottom=False,      # ticks along the bottom edge are off
top=False,         # ticks along the top edge are off
labelbottom=True)  # labels along the bottom edge are off
plt.show()

x = np.arange(3)
plt.bar(x, height= [0.23,0.28,0.27], color='b')
plt.xticks(x, ['5','4','3'])

plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('Decision tree-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
#plt.axes().spines['right'].set_visible(False)
# plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
plt.axes().axhline(linewidth=0, color="k")
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are
    bottom=False,      # ticks along the bottom edge are off
top=False,         # ticks along the top edge are off
labelbottom=True)  # labels along the bottom edge are off
plt.show()
```
Appendix

x = np.arange(3)
plt.bar(x, height= [0.24, 0.24, 0.24], color='b')
plt.xticks(x, ["5", '4', '3'])
plt.xlabel('Number of features')
plt.ylabel('R^2')
plt.title('MLP regressor-MARD')
plt.axis([-1, 3, -0.1, 0.6])
plt.grid(False)
plt.axes().spines['right'].set_visible(False)
plt.axes().spines['top'].set_visible(False)
plt.axes().spines['bottom'].set_position(('data', -0.1))
plt.axes().tick_params(axis='x', pad=5)
plt.axhline(y=0, color='k')
plt.axes().axhline(linewidth=0, color="k")
plt.tick_params(axis='x',          # changes apply to the x-axis
which='both',      # both major and minor ticks are
bottom=False,      # ticks along the bottom edge are off
top=False,         # ticks along the top edge are off
labelbottom=True) # labels along the bottom edge are off
plt.show()

8.6 PHI Challenge labeling algorithm

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import argparse
import sys
import tensorflow as tf

parser = argparse.ArgumentParser()
parser.add_argument('--image', required=True, type=str, help='Absolute path to image file.')
parser.add_argument('--num_top_predictions', type=int, default=5, help='Display this many predictions.')
parser.add_argument('--graph', required=True, type=str, help='Absolute path to graph file (.pb)')
parser.add_argument('--labels', required=True, type=str, help='Absolute path to labels file (.txt)')
parser.add_argument('--output_layer', type=str, default='final_result:0', help='Name of the result operation')
parser.add_argument('--input_layer', type=str, default='DecodeJpeg/contents:0', help='Name of the input operation')

def load_image(filename):
"""Read in the image_data to be classified."""
return tf.gfile.FastGFile(filename, 'rb').read()

def load_labels(filename):
    """Read in labels, one label per line."""
    return [line.rstrip() for line in tf.gfile.GFile(filename)]

def load_graph(filename):
    """Unpersists graph from file as default graph."""
    with tf.gfile.FastGFile(filename, 'rb') as f:
        graph_def = tf.GraphDef()
        graph_def.ParseFromString(f.read())
        tf.import_graph_def(graph_def, name='')

def run_graph(image_data, labels, input_layer_name, output_layer_name,
              num_top_predictions):
    with tf.Session() as sess:
        # Feed the image_data as input to the graph.
        # predictions will contain a two-dimensional array,
        # dimension represents the input image count, and the
        # other has
        # predictions per class
        softmax_tensor = sess.graph.get_tensor_by_name(output_layer_name)
        predictions, = sess.run(softmax_tensor, {input_layer_name: image_data})

        # Sort to show labels in order of confidence
        top_k = predictions.argsort()[-num_top_predictions:][::-1]
for node_id in top_k:
    human_string = labels[node_id]
    score = predictions[node_id]
    print('%s (score = %.5f)' % (human_string, score))

return human_string, score

def main(argv):
    """Runs inference on an image."""
    if argv[1:]:
        raise ValueError('Unused Command Line Args: %s' % argv[1:])

    if not tf.gfile.Exists(FLAGS.image):
        tf.logging.fatal('image file does not exist %s', FLAGS.image)

    if not tf.gfile.Exists(FLAGS.labels):
        tf.logging.fatal('labels file does not exist %s', FLAGS.labels)

    if not tf.gfile.Exists(FLAGS.graph):
        tf.logging.fatal('graph file does not exist %s', FLAGS.graph)

    # load image
    image_data = load_image(FLAGS.image)

    # load labels
    labels = load_labels(FLAGS.labels)

    # load graph, which is stored in the default session
    load_graph(FLAGS.graph)
run_graph(image_data, labels, FLAGS.input_layer,
FLAGS.output_layer,
    FLAGS.num_top_predictions)

if __name__ == '__main__':
    FLAGS, unparsed = parser.parse_known_args()
    tf.app.run(main=main, argv=sys.argv[:1]+unparsed)
REFERENCES


