

MDSAA

Master Degree Program in
Data Science and Advanced Analytics

AI BASED EARTHQUAKE FORECASTING IN PORTUGAL

Rita da Silva Soares

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

AI based Earthquake Forecasting in Portugal

by

Rita da Silva Soares

Master Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science.

Supervised by

Leonardo Vanneschi, PhD, NOVA Information Management School

July, 2024

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisbon, 01/07/2024

ABSTRACT

In this work, the potential of artificial intelligence for forecasting earthquakes in Portugal, a region under looked with only the Western Azores area being previously studied, is investigated. The target of this research is set to prediction of an event occurrence with a magnitude higher than 4.0 in the next 10 days. Seismic sequential data retrieved from an open-source catalogue is leveraged through machine learning models (Naïve Bayes, K-Nearest Neighbours, Support Vector Machine, Random Forest and neural networks focusing on the application of LSTM layers) to predict earthquake occurrences. The study involves an extensive data pre-processing with feature extraction by exploiting geophysical formulas and multiple strategies to capture the sequential information. A study on the performance of each strategy and model is conducted in an attempt to identify if a methodology or algorithm should be preferred on earthquake studies in the area under analysis. The final results showcase some advancements in terms of state of the art and introduce new discoveries but still lack the minimum quality to consider the deployment of such warning systems on real-life. This thesis contributes to the development of AI-driven earthquake prediction systems in Portugal and suggests directions for future improvements.

KEYWORDS

Earthquake Prediction; LSTM; Machine Learning; Portugal; Sequential Data

Sustainable Development Goals (SDG):



TABLE OF CONTENTS

1. Introduction.....	1
2. Literature review	4
2.1. Earthquake forecasting through history	4
2.2. Diversity of data types.....	4
2.3. Seismic Indicators	5
2.4. Deep learning studies	6
2.5. Focus on LSTM.....	8
2.6. Regional studies: Portugal	9
2.7. Research Gap.....	10
3. Methodology	11
3.1. Data Source	11
3.2. Pre-Processing: Data Fragmentation and Exploration	11
3.3. Feature Engineering: Input Vector Creation	15
3.3.1.Seismic Indicators.....	16
3.3.2.Labelling.....	19
3.4. Experimental Design.....	20
3.4.1.Machine Learning Exploration	21
3.4.2.Neural Network Exploration.....	22
3.5. Quality Metrics	23
4. Theoretical and Experimental Settings	26
4.1. Machine Learning Methods.....	26
4.1.1.K-Nearest Neighbours	26
4.1.2.Gaussian Naïve Bayes.....	27
4.1.3.Support Vector Machine	28
4.1.4.Random Forest	31
4.2. Neural Networks.....	33
5. Results and Discussion.....	38
5.1. Azores	38
5.2. Mainland and Surroundings	41
6. Conclusion	44
7. Limitation and Future Work	45
Bibliographical References.....	46
Appendix A	50
Appendix B	56

Appendix C	60
Appendix D	64
Annexes	84

LIST OF FIGURES

Figure 1.1 Representation of the main geological structures responsible of Portugal's complex geodynamics. <i>MAR</i> Mid-Atlantic Ridge, <i>ATJ</i> Azores Triple Junction, <i>TR</i> Terceira Ridge, <i>EAF</i> East Azores Fracture Zone, <i>GF</i> Gloria Fault, <i>GB</i> Gorringe Bank, <i>Gs</i> Gibraltar Strait, <i>NA</i> North America plate, <i>Eu</i> Eurasia plate, <i>Nu</i> Nubia plate, <i>P</i> Portugal, <i>S</i> Spain, <i>A</i> Algeria, <i>M</i> Morocco. From "Seismicity along the Azores-Gibraltar region and global plate kinematics" by Bezzeghoud, M., Adam, C., Buorn, E., Borges, J. F., & Caldeira, B. (2014). Journal of Seismology, 18(2), 205–220.	2
Figure 3.1 - Geographical distribution of the retrieved data after application of K-Means with $K = 2$, each point represents an earthquake, grey on the Azores region (Cluster #1) and green on Mainland and surroundings (Cluster #0), and the grey lines the fault lines. Plot computed in Python using Folium library on the function <i>plot_spatial_distributon</i>	12
Figure 3.2 This figure presents the Frequency -Magnitude distribution per cluster, along with the indicated magnitude of completeness (Mc) for each cluster. The gray lines represent Cluster #1 (Azores), and the green lines represent Cluster #0 (Mainland and surroundings). Panel (a) depicts the distribution for the complete dataset, while panel (b) focuses on the data from 2020 onwards. Plots computed using the matplotlib library in Python under the function <i>freq_mag</i>	13
Figure 3.3 Magnitude distribution over the years for each dataset. With magnitude 4.0, the threshold value for label 1, highlighted by a red horizontal line. Plots computed using Python and Matplotlib library under the function <i>plot_mag_time</i>	14
Figure 3.4 Overview of the datasets constructed on the earthquake data for the two regions: Azores and Mainland Portugal. Eight studies were conducted by region, each a combination of time-frame segmentation, sliding window technique and method for b-value estimation. This results in a total of 16 datasets for research on this study.	20
Figure 4.1 (a) Representation of a simple linear binary problem in a 2-dimensional space. (b) Introduction of a soft-margin representation to the problem.	29
Figure 4.2 (a) Representation of a non-linear binary problem in a 2-dimensional space. (b) Representation of the same problem after transformation of the points into a 3-dimensional space.....	30
Figure 4.3 – Representation of the general structure of a decision tree – with depth 2.....	31
Figure 4.4 (a) Representation of an FNN receiving an input vector of size j with 2 hidden layers with k and m neuros, respectively. Each circle represents one neuron and the lines the weight connections between them, each representing one value which at the end of each layer is scaled through an activation function. (b) Representation of an RNN with 1 hidden layer having k neurons. The curved line representing the inner loops.	33

Figure 4.5 Representation of an unfolded RNN layer, representing the inner loop process that each RNN unit goes through. Processing a sequence of size t , where wx , wh and wy represent, respectively, the input, hidden state and prediction weights. The Z symbolizes the hidden state, ht , pre-activation function.	34
Figure 4.6 Representation of an LSTM unit, where the circles represent mathematical operations, sig and tanh, sigmoid and tanh activation functions respectively.	36
Figure 4.7 Description of the RNN general architecture applied to all datasets. With the Bidirectional layer being tuned on whether to be included or not.	36
Figure 5.1 Boxplot of the β -f1 metric on the Azores cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.....	39
Figure 5.2 Boxplot of the pair wise results (P0 and P1, Sensitivity and Specificity) on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.	40
Figure 5.3 Boxplot of the pair wise results (P0 and P1, Sensitivity and Specificity) on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.	41
Figure 5.4 Boxplot of the β -f1 metric on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.	43
Figure A.1 Iteration 1 with a sliding window of size $\Delta t = 1$ using the time method. The pink rectangle corresponding to the current window under analysis and the blue the prediction window. In this step, a vector is created using the past n events – in this case the one in the current window (pink). The label is set to 1 since both conditions are met on the prediction window (blue).	50
Figure A.2 The second iteration has an input vector computed with the information of the current window and a label of zero since one of the conditions is not met - the existence of an event in the prediction window.....	50
Figure A.3 Third iteration has an input vector equal to the previous one, since the n past events are the same (the one on the 02/06/2020). The label is set to zero since no events occurred in the prediction window.....	51

Figure A.4 Fourth iteration has an input vector equal to the last two previous ones, since the n past events are the same. The label is set to zero, even though events occurred in the prediction window the threshold magnitude is not reached.....	51
Figure A.5 Fifth iteration the input vector based on the last 1 event corresponds to the one on the 05/06/2020, the current window. The label is set to zero since no events occurred in the prediction window.....	52
Figure A.6 Sixth iteration has an input vector equal to the last one, since the n past events are the same. The label is set to one since at least one event in the prediction window is higher than 4.0.....	52
Figure A.7 The second iteration has an input vector computed with the information of the current window and a label of zero since one of the conditions is not met - the existence of an event in the prediction window.....	53
Figure A.8 Second iteration since there is at least one event in the current window an input vector is computed using the last n events (n=1). The label is set to zero since there is no event in the prediction window.....	53
Figure A.9 Third iteration there is no event in the current window, as such no input vector creation. No input vector then no label.....	54
Figure A.10 Fourth iteration since there is no event then no input vector calculation is triggered. Then even though there is an event in the prediction window the input vector is not created.....	54
Figure A.12 Sixth iteration does not trigger an input vector creation since there is no event in the current window as such no label is created either.....	55
Figure A.11 Fifth iteration input vector created given there is at least one event in the current window. Label set to zero due to no event present in the prediction window.....	55
Figure D.1 Confusion Matrix results for the Azores [2005, 2024] cluster on the KNN algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	64
Figure D.2 Confusion Matrix results for the Azores [2020, 2024] cluster on the KNN algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	65
Figure D.3 Confusion Matrix results for the Azores [2005, 2024] cluster on the SVM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	66
Figure D.4 Confusion Matrix results for the Azores [2020, 2024] cluster on the SVM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	67

Figure D.5 Confusion Matrix results for the Azores [2005, 2024] cluster on the NB algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	68
Figure D.6 Confusion Matrix results for the Azores [2020, 2024] cluster on the NB algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	69
Figure D.7 Confusion Matrix results for the Azores [2005, 2024] cluster on the RF algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	70
Figure D.8 Confusion Matrix results for the Azores [2020, 2024] cluster on the RF algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	71
Figure D.9 Confusion Matrix results for the Azores [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	72
Figure D.10 Confusion Matrix results for the Azores [2020, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	73
Figure D. 11 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the KNN algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	74
Figure D.12 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the KNN algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	75
Figure D.13 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the SVM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	76
Figure D.14 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the SVM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	77
Figure D.15 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the NB algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	78
Figure D.16 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the NB algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	79

Figure D.17 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the RF algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	80
Figure D.18 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the RF algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	81
Figure D.19 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	82
Figure D.20 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and occ the occurrence. The lsq and mlk identify the estimation method.....	83

LIST OF TABLES

Table 3.1 – Input for data selection on the EMSC earthquake catalogue	11
Table 3.2 General information on the effect each sliding window methods, occurrence and time, have on each dataset in terms of size and percentage of positive events.....	20
Table 4.1 Estimators subject to tuning on the KNN algorithm and the corresponding search space. Resulting in a total of 1024 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20.	27
Table 4.2 Estimators subject to tuning on the NB algorithm and the corresponding search space. Resulting in a total of 400 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20.....	28
Table 4.3 Estimators subject to tuning on the SVM algorithm and the corresponding search space. Resulting in a total of 320 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20.....	31
Table 4.4 Estimators subject to tuning on the RF algorithm and the corresponding search space. Resulting in a total of 720 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20.....	32
Table 4.5 Estimators subject to tuning on the LSTM algorithm per layer and the corresponding search space. With 0* corresponding to the removal of the corresponding layer – Bidirectional. Resulting in a total of 288 combinations.	37
Table B.1 Grid Search results per Mainland cluster dataset on a search space of 1024 combinations on the KNN algorithm.....	56
Table B.2 Grid Search results per Azores cluster dataset on a search space of 1024 combinations on the KNN algorithm.....	56
Table B.3 Grid Search results per Mainland cluster dataset on a search space of 400 combinations on the Gaussian NB algorithm.	56
Table B.4 Grid Search results per Azores cluster dataset on a search space of 400 combinations on the Gaussian NB algorithm.	57
Table B.5 Grid Search results per Mainland cluster dataset on a search space of 320 combinations on the SVM algorithm.	57
Table B.6 Grid Search results per Azores cluster dataset on a search space of 320 combinations on the SVM algorithm.	57
Table B.7 Grid Search results per Mainland cluster dataset on a search space of 720 combinations on the RF algorithm.	58
Table B.8 Grid Search results per Azores cluster dataset on a search space of 720 combinations on the RF algorithm.....	58

Table B.9 Grid Search results per Mainland cluster dataset on a search space of 288 combinations on the LSTM algorithm.....	59
Table B.10 Grid Search results per Azores cluster dataset on a search space of 288 combinations on the LSTM algorithm.....	59
Table C.1 Features selected per model and dataset on the Azores cluster for the KNN and NB algorithms.....	60
Table C.2 Features selected per model and dataset on the Azores cluster for the SVM and RF algorithms.....	61
Table C.3 Features selected per model and dataset on the Mainland and Surroundings cluster for the KNN and NB algorithms.....	62
Table C.4 Features selected per model and dataset on the Mainland and Surroundings cluster for the SVM and RF algorithms.	63
Table D.1 Metrics values for the Azores cluster on the KNN algorithm.	65
Table D.2 Metrics values for the Azores cluster on the SVM algorithm.....	67
Table D.3 Metrics values for the Azores cluster on the NB algorithm.....	69
Table D.4 Metrics values for the Azores cluster on the RF algorithm.	71
Table D.5 Metrics values for the Azores cluster on the LSTM algorithm.....	73
Table D.6 Metrics values for the Mainland and Surroundings cluster on the KNN algorithm.75	75
Table D.7 Metrics values for the Mainland and Surroundings cluster on the SVM algorithm.77	77
Table D.8 Metrics values for the Mainland and Surroundings cluster on the NB algorithm... 79	79
Table D.9 Metrics values for the Mainland and Surroundings cluster on the RF algorithm. .. 81	81
Table D.10 Metrics values for the Mainland and Surroundings cluster on the LSTM algorithm. 83	83

LIST OF ABBREVIATIONS AND ACRONYMS

MAR	Mid-Atlantic Ridge
EFZ	East Fracture Zone
TR	Terceira Rift
GF	Gloria Fault
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
RBF	Radius Basis Function Neural Network
FNN	Feedforward Neural Network
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
NB	Naïve Bayes
LSTM	Long Short-Term Memory
ML	Machine Learning
EMSC	Euro-Mediterranean Seismological Centre
M_c	Magnitude of Completeness
FMD	Frequency-Magnitude Distribution
LSQ	Least Squares
MLK	Maximum Likelihood
USGS	United States Geological Survey

1. INTRODUCTION

The sudden shaking of the ground, due to the passage of seismic waves generated by the Earth's dynamical processes, is known as an Earthquake. This type of catastrophe is particularly dangerous to humankind since it strikes without an apparent warning sign and during its short-lasting period may lead to massive destruction. Additionally, this single event can trigger a chain of secondary disasters, like tsunamis and fires.

Seismic activity is a well-known event to mankind with higher intensity events leaving long-lasting scars in society. The health consequences outcome from these events constitute one of the main areas of impact with numerous losses of lives, physical and psychological injuries on survivors, and a worldwide health impact due for instance to the favorable conditions generated after certain earthquakes to the spread of infectious diseases (Mavrouli et al., 2023).

In addition, the economic sector has been experiencing an increase in losses due to natural disasters since the 1980s (Botzen et al., 2019). The impact in this sector is mainly due to infrastructure losses, building destruction, and disruptions to the supply chain and production capacity which may also affect the economies of dependent nations (Daniell et al., 2012). It is easily understood that the mentioned impacts equate to the social area, since the collapse of buildings may lead to an increase in homelessness and education delay to name a few examples.

A known event, with devastating health and social-economic consequences, is the earthquake of 1755 in Lisbon, the largest in the European records, with an estimated death toll, merely in the city, of 10 000 to 100 000, making it one of the deadliest events as well (Bezzeghoud et al., 2014; Custódio et al., 2015). The largest European intraplate earthquakes, generated within one plate, away from its boundary, were also registered in Portugal, and occurred in the following years and locations; the 1909 in Benavente; 1858 in Setúbal, and the 1531 in Lower Tagus Valley region (Custódio et al., 2015).

Overall, Portugal is considered to have a moderate seismicity with small events intercalated with an occasional major one (Borges et al., 2001). Despite the moderate seismic activity, Portugal's geological story is far more intricate due to its position and different territory, mainland and archipelagos, Azores, and Madeira. The study of Portugal's seismicity is typically done within a specific region or by defining different regions due to the diverse seismicity characteristics and focal mechanisms (Bezzeghoud et al., 2014).

The Azores archipelago, composed of 9 volcanic islands, organized into 3 groups, is known for frequent seismic events of low magnitude. The origin of this high seismicity is mostly its location on the convergence of 3 main plates (Triple Junction), the Eurasian, Nubian, and, the North American as shown in Figure 1.1. The ongoing, complex dynamic between these plates originated multiple geological features, the Mid-Atlantic Ridge (MAR), East Fracture Zone (EFZ)

and the Terceira Rift (TR), the last one still generating debate over its origin and geodynamics. The East Fracture Zone does not have recent activity associated with most earthquakes related to either the MAR or the EFZ (Bezzeghoud et al., 2014; Fontiela et al., 2018). On the other hand, not far from the Azores lays the Gloria Fault, GF (Figure 1.1), an extension of the EAF, associated with low seismic activity but possibly originating large-magnitude events in long time intervals felt on some of the Azores islands (May 8, 1939, e.g.).

When approaching the southern mainland (Figure 1.1), there is no longer a triple junction to be considered but the interaction between two plates, the western Eurasia and Nubian, which determines a region known as Ibero-Maghrebian, also englobing a part of Morocco and Spain. This area, contrary to the previously mentioned, is characterized by moderate-magnitude earthquakes, its complexity leads investigators to further divide it into more zones. Focusing on Portugal's mainland territory multiple potentially active structures are identified such as the Porto-Tomar fault; Lower Tagus Valley fault; and Marquês de Pombal fault to name a few. With more presence in the central and southern areas of the country, where the seismicity risk is also higher (Bezzeghoud et al., 2014; Carvalho & Malfeito, 2018; Custódio et al., 2015).

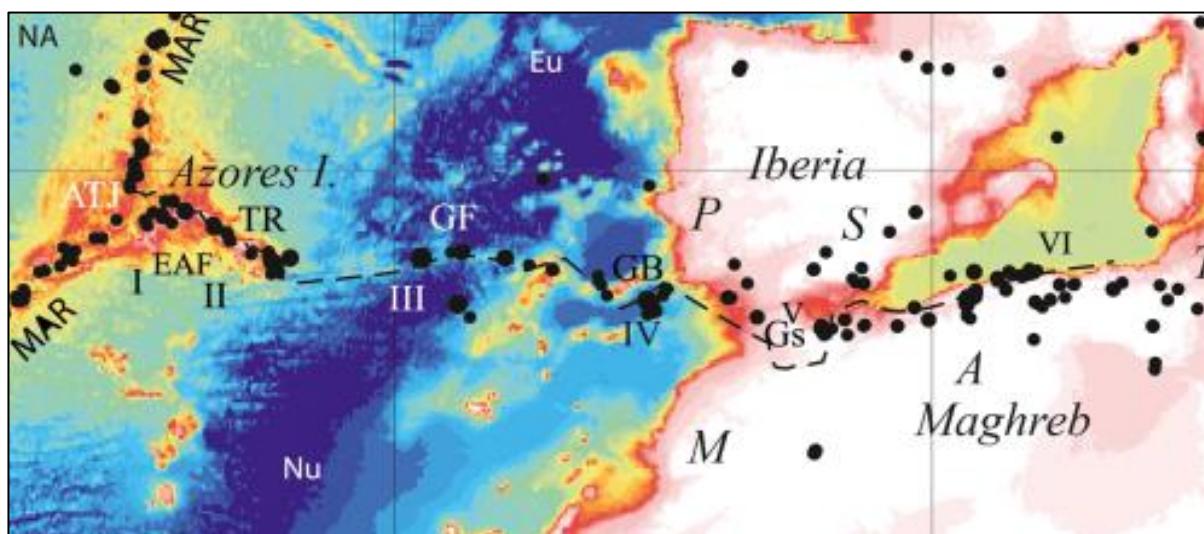


Figure 1.1 Representation of the main geological structures responsible of Portugal's complex geodynamics. *MAR* Mid-Atlantic Ridge, *ATJ* Azores Triple Junction, *TR* Terceira Ridge, *EAF* East Azores Fracture Zone, *GF* Gloria Fault, *GB* Gorringe Bank, *Gs* Gibraltar Strait, *NA* North America plate, *Eu* Eurasia plate, *Nu* Nubia plate, *P* Portugal, *S* Spain, *A* Algeria, *M* Morocco. From (Bezzeghoud et al., 2014).

In conclusion, although Portugal does not constitute a high seismicity risk with frequent earthquakes of moderate to elevated magnitude like other countries, it is a complex geological territory with a record of highly destructive events, making its study an intricate necessary task.

Nowadays, there is an ongoing heated debate on whether such events can be predicted or not, making it a hot topic for research and application of methods from multiple areas

(statistics, geophysics; data science; e.g.) using a variety of data types from multiple regions in the world, some with promising results others not so much. But the potential impact such studies could have in the world should not be overlooked, the investment in such a research area should be seen as a step closer to a future where the catastrophic consequences of an earthquake can be mitigated through an early warning.

Research Questions

Unfortunately, Portugal is a region often underrepresented in earthquake prediction studies despite the described activity. With this in mind, the work here presented attempts to answer the following research questions:

RQ1: Do certain algorithms or methodologies exhibit superior performance in earthquake prediction when applied to data specific to Portugal?

RQ2: Considering the western Azores region has been subject to earthquake prediction studies; can those techniques also be applied to the mainland region of Portugal?

To answer these questions this study conducts exploratory research on Portuguese territory using commonly used algorithms, from machine learning (K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Random Forests and Recurrent Neural Networks) in earthquake forecasting. The method chosen consists of approaching this problem as a supervised classification task. This can be achieved by using time-series records which include the date, magnitude, and location of the events. Following the revised literature, a time-sliding window is utilized to compute an input vector from the initial data, composed of features that describe the geodynamics of the region, to each vector a binary label is associated which represents the occurrence of an event of magnitude higher than 4.0 in the next 10 days. This approach will have four variations to include all methods of input vector creation found in the revised literature.

This project holds the potential for a two-fold contribution. Firstly, while the Azores have received some attention in earthquake prediction research, this project can significantly contribute by applying a wider range of algorithms and methodologies to the region. Secondly, by including the remaining Portuguese territory, a region underrepresented in such studies, this research can introduce this area as a valuable point of interest for future investigations. Additionally, by employing a range of algorithms and exploring various methodologies, the project aims to identify if one method or algorithm proves to be more suitable for earthquake forecasting in the specific geological context of Portugal.

2. LITERATURE REVIEW

2.1. EARTHQUAKE FORECASTING THROUGH HISTORY

Earthquake prediction is a longstanding pursuit, with early seismology students monitoring animal behavior and fluctuations in the magnetic field, among other factors, in the hope of finding a pattern (Geller, 1997). Furthermore, mathematical exploration of correlations and models that could forecast an event has also been widely explored, with the hypothesis that a probabilistic distribution can predict with some degree of confidence the next event. Some recurrently studied models include Poisson and Gamma distributions. (Dasgupta et al., 1998; Roy, 2014)

As the world evolved, powerful technologies emerged. As such, in the late 20th and early 21st centuries, a new branch emerged consisting of the exploration of machine learning algorithms aimed at earthquake predictions. Since then, numerous research studies have been conducted, investigating a broad range of models, data, and parameters to build a trustworthy algorithm for early warning. Although promising, this novel branch cannot be claimed to be superior to other approaches since the outcomes are influenced by various factors that prevent such generalization. This has been shown by Lakshmi and Tiwari, who conducted a comparative study of neural network performance against traditional approaches, holding the factors constant to ensure comparability. It was shown, in the specific scenario and techniques chosen, that the quality of the results is similar for both branches. (Sri Lakshmi & Tiwari, 2009)

In this chapter, a summary of a subset of the multiple possibilities explored in the field of machine learning algorithms is presented via a variety of studies published in recent years, focusing on the most relevant aspects of this study.

2.2. DIVERSITY OF DATA TYPES

Earthquake forecasting offers an extensive option of data types, for instance, geophysical measurements, with some being radon gas concentration, steam pressure, and soil temperature, among others, which were studied by Niksarlioglu using artificial neural networks, ANN, and non-hierarchical clustering for the prediction of earthquake magnitude. (Niksarlioglu & Kulahci, 2013)

Seismic waves, an alternative data type, have been explored through convolutional neural networks, CNN, for earthquake forecasting and detection since machines may misinterpret an event as noise. For this purpose, the labels implemented identified either noise or an event in a specific area, which was obtained via K-means clustering, allowing simultaneous location prediction. The results were positive, achieving a better runtime than current models using signal data. (Perol et al., 2018)

Moustra explored seismic signals against time-series data – the magnitude and location of an event – to study the prediction of earthquakes in Greece. For time-series data, the input vector

of the artificial neural network is of size n , with each entry representing the maximum magnitude recorded in each of the last n days. To improve the prediction of high-magnitude events, the least represented in the dataset, the neural network is trained in two phases: first, only considering the events with a magnitude higher than a threshold, and second, training on the whole dataset but initializing with the weights of the first train. This approach results in a slight increase in accuracy for large earthquakes, which was not considered successful. The unsatisfactory outcome may result from the ambitious goal of forecasting an earthquake within a temporal range of one day.

In the second phase, seismic electrical signals are fed to an artificial neural network to forecast the next major event. Although the predictions have a good accuracy, there is insufficient data on higher-magnitude events, resulting in them not being forecasted. (Moustra et al., 2011)

Despite the potential of geophysical and waveform data, two main things should be considered: first, the retrieval of this information may demand some investment in new technology, and second, the number of registered events may be low since most regions probably acquired the necessary equipment recently or don't have it yet. As is the case of the area analyzed in the previous study. (Moustra et al., 2011)

Time-series data, typically consisting of location, magnitude, and depth, is open-source and has records from a distant past. On top of this, quality results have been achieved with this type of data, either by using the data as input or some researchers have successfully applied geological laws and computed new features, which through this thesis will be refer to as seismic indicators.

2.3. SEISMIC INDICATORS

Panakkat and Adeli introduced one of the most widely used sets of indicators when investigating the prediction of earthquakes in Southern California in a monthly window using different neural networks: recurrent neural network, RNN; radial basis function, RBF; and feedforward neural network, FNN. Each is tuned for the best architecture by a steady increase of hidden layers and nodes until the change in R-score is smaller than a predefined value.

Eight seismic parameters were computed: three based on Gutenberg- Richter's inverse power law, two on the characteristic temporal magnitude distribution, and the rest following general statistical measurements. To better understand the effect these indicators have over the R-score, the authors conducted a study on the effect the removal of one has over it, repeating this process for each one and the three neural networks proposed. Although some regions show R-score fluctuations, indicating some parameters possibly harm the results, there was no further study on feature selection.

The final results obtained on the eight-value vector input present RBF and RNN as the most promising models, yielding high R-Score results. Nonetheless, RNN is considered the better-

performing model since, allied with the high R-scores, it has a low false alarm rate. (Panakkat & Adeli, 2007)

Although not as recognized, in 2013, Reyes presented seven new seismic indicators, with five values being increments of the computed b-value. The sixth value is the maximum magnitude registered in the past seven days so that the ANN can capture the relationship between Omori/Utsu and Bath's laws. The final parameter is determined by evaluating the probability density function, which quantifies the likelihood of an earthquake exceeding a specific threshold value.

The seismic indicators fed an artificial neural network with an output of the maximum magnitude registered in the next five days for Chilean regions, which yielded satisfactory results. The author performed a broader analysis of the efficiency of the indicators, converting the problem into a classification one and using multiple models: K-nearest neighbors (KNN), K-means, and support vector machines (SVM). The performance of the indicators was promising overall, with the ANN algorithm standing out. (Reyes et al., 2013)

A direct comparison between the two studies presented isn't possible since each used distinct model and applied them to different regions, which leads to the question of which set of indicators one should favor. Tackling this issue in 2016, researchers published an exploration of the sensitivity of models to the input indicators mentioned. Although the data used for evaluation was from four Chilean zones, a universal methodology was proposed, generalizing the results.

Summarizing, besides the already briefly explained calculations, a set of new variables is introduced to fine-tune the seismic indicators concerning the magnitudes, time intervals, and number of events. In addition, the b-value is obtained through two different formulas to determine which is better suited to the considered area. The binary label is generated based on whether the maximum magnitude in a prediction horizon is higher than a threshold value.

Regarding feature selection, to gather the best tuning for each indicator per location, each of the five tuning variables is changed while holding the remaining constant, following *ceteris paribus*. This way, seismic indicators with dependencies on those variables are changed. The models chosen to measure the effect of tuning over different combinations of the indicators were KNN, Naïve Bayes (NB), ANN, J48, and SVM. This research showed how fine-tuning the seismicity parameters, following the proposed methodology, enhances the accuracy of the predictions on multiple classifiers. (Asencio-Cortés et al., 2016)

2.4. DEEP LEARNING STUDIES

For most of the previously presented studies, one common aspect is the overperformance of ANNs, most likely due to their ability to capture non-linear complex relationships, as is the case of earthquakes. As such, deep learning quickly became an area that sparked interest in earthquake forecasting.

A study on the Northern Red Sea region for the prediction of the magnitude of the next event used a feedforward neural network, FNN, with a backpropagation learning algorithm. The model outperformed the statistical distributions plotted for quality assessment, making the research considered successful. (Alarifi et al., 2012)

Another exploration of the efficacy of this type of neural network was conducted by Narayananakumar, who forecasted the magnitude range of the next event in the Himalayan regions by feeding a nine-element input vector to the feedforward neural network. The results showed a higher accuracy for lower to moderate magnitude ranges, a common issue attributed to class imbalance. Still the accuracy reached on moderate events is an achievement since, according to the authors, this algorithm outperforms previous studies on events in those magnitude ranges. (Narayananakumar & Raja, 2016)

Exploring a different type of neural network and attempting to mitigate the limited dataset size, which is a common problem related to the difficulty of predicting moderate and high-magnitude events, although results won't be comparable to the previous study due to different data treatment and region, Alexandridis computed a radius basis function neural network, RBF, combined with a fuzzy means training algorithm modified to incorporate the leave-one-out training procedure that allows for more efficient training. In this case, declustering is studied as a pre-processing step and consists of aftershock removal from the catalog.

The described RBF model and a multi-layer perceptron neural network receive as input the seismicity rates and output the predicted elapsed time between two high-magnitude events.

For both models, in a first scenario, raw data is used with no pre-processing, and with a significant earthquake being higher than magnitude 6, the results compared to the second scenario considered, on which the only difference is the addition of the pre-processing step, proves the hypothesis that the removal of aftershocks is an essential pre-processing step when predicting the elapsed time between significant events.

The last case increased the magnitude from which an event is significant, keeping the pre-processing step. The results were worse than in the second scenario, most likely a result of the decrease of events in the training data.

In this research, the radius basis function outperforms the other model, and the declustering technique is introduced as an essential step for temporal predictions using elapsed time. (Alexandridis et al., 2014)

In 2017, a study was conducted over the Hindu Kush area for events with a magnitude higher or equal to 5.5. For this, the authors used tree-based techniques, random forest, and an ensemble of trees using LPBoost, as well as deep learning algorithms, a feedforward neural network, and a recurrent neural network, both utilizing the Levenberg-Marquardt backpropagation method.

The Panakkat parameters were used as input for the different algorithms, and the results show a trade-off between the distinct techniques, which makes it harder to name one as the outperformer. Since although some performed better on the measure of true positives, they underperformed on the false positives, and vice versa. (Asim et al., 2017)

As mentioned, Panakkat explored multiple deep-learning algorithms that didn't perform as desired on moderate earthquakes (Panakkat & Adeli, 2007). In 2009, the authors published research on attempts to improve the accuracy of the RNN for moderate earthquakes in California. In this same study, they explored a new RNN architecture that predicts the time and epicenter of moderate-magnitude and large-magnitude events.

First, the area considered is divided into smaller regions, for which the input parameters are calculated and fed to the same RNN as the one from the previous study. One of the main issues faced is the non-representative data for some regions, which lacks significant events to produce quality results. The results for high magnitudes don't show enough quality for areas with poor data, a non-surprising result. However, for the remaining regions, satisfactory results were obtained even though the time window for earthquake forecasting is smaller than in the previous study. Multiple time frames were tested on those regions to gather which produced better results. Contrary to what could be expected, the prediction of an event in the next two weeks results in better output than in a month, having a loss of accuracy when the window is too big or too narrow.

For the second study, the parameters are not computed using a predefined number of prior events like in previous studies but using all events higher than a threshold magnitude, between two major earthquakes. This information is then fed to two RNN, one outputting a normalized latitude and longitude vector and the other the elapsed time from the last major event to the predicted one. The outcome of both neural networks was considered a success, predicting the location and date of the next four events within an error range that was considered acceptable. Notably, on top of this achievement the model also appears to recognize the different pattern between mainshocks and aftershocks, which validates the decision of not declustering the events, contrary to the previous study mentioned of Alexandridis.

Although, considered a valuable result, some concerns are expressed on the size of the dataset and lack of high-magnitude events which doesn't allow for a detailed quality assessment on all magnitude ranges – the events efficiently predicted were mainly in the moderate category with magnitudes around 5. (Panakkat & Adeli, 2009)

2.5. Focus on LSTM

The encouraging results of recurrent neural networks led to the recent exploration of long short-term memory neural networks, LSTM, which are RNNs where the standard cell is replaced by an LSTM one, allowing the exploration of long-term interval correlations in addition to mitigating the gradient problems.

This model has recently been studied for earthquake predictions, in some cases outperforming other artificial neural networks, as is the case of Berich's research over the Morocco area, which attempted to predict the magnitude, location, and year of the next event. With this purpose, two studies were performed: the first, a tuned LSTM applied to the whole dataset and, the second, two tuned LSTMs received input data from two different datasets, a result of the decomposition of the original data into low magnitude and high magnitude events. For comparison, an ANN was also computed, and, over both input data, the LSTM model outperformed (Berich et al., 2020). Bhargava achieved the same results with his research on the Himalayas region and the comparison between two models, ANN and LSTM. While both failed to produce quality results on high-magnitude events, the LSTM outperformed the others (Bhargava & Pasari, 2022).

Following the same line of thought, Wang focused on the exploration of LSTM using a different input vector that allowed for the study of relationships between events in distinct locations, a spatial-temporal vector. For this, he studied the data from China, comparing the results of a one-dimensional input representing earthquake existence, to a two-dimensional input on each binary entry symbolized event occurrence in a defined region. The results met the hypothesis that a two-input vector can achieve better results by considering simultaneously the frequency over time and multiple locations. (Wang et al., 2020)

2.6. REGIONAL STUDIES: PORTUGAL

Most studies focus on high-activity and intensity regions, but a few have started to explore the Iberian Peninsula, although restricting the Portugal region to a section of the Azores.

In 2006, Alves focused on the similarities between the stock market and earthquakes and adapted finance calculations to produce seven new seismic indicators. After analyzing pattern-matching, six parameters were used as input in an Artificial Neural Network with Backpropagation, enabling the prediction of the location, magnitude, and month of the next seismic event. The results were inspiring, with two events correctly predicted within an error window. (Alves, 2006)

In 2010, Morales-Esteban proposed a study of the Iberian Peninsula seismic patterns through clustering techniques. The removal of aftershocks and foreshocks from the catalog was done beforehand. From the mainshocks left, it was concluded only the Alboran Sea and Western Azores-Gibraltar fault region had enough events to proceed with the study.

The K-Means algorithm performed for those regions over the magnitude, b-value, and date of occurrence after retrieving the ideal number of clusters through the silhouette method, revealed distinct seismic patterns in both regions. The analysis of the results points out that changes in b-values preceding significant earthquakes could be used as a seismic precursor for earthquake forecasting. To validate the significance of the results non-parametric statistical tests were used. (Morales-Esteban et al., 2010)

The same author, in 2013, studied once again the Alboran Sea and Western Azores–Gibraltar fault, now on the complete catalog of events to predict the probability of earthquakes within seven days, higher than a threshold magnitude and inside a defined magnitude range. An Artificial Neural Network with Backpropagation was deployed using the seven seismic parameters (Reyes et al., 2013) and one output, the maximum magnitude observed in the next seven days. Although, the results were satisfactory, to further assess the quality of the proposed model, the output was changed to binary to enable the comparison with classical machine learning algorithms – Support Vector Machine, Naïve Bayes, and M5P. The results support the hypothesis that ANN performs better overall, having a better false alarm rate and sensitivity. (Morales-Esteban et al., 2013)

In the same year, a study on the model sensitivity to the parameters chosen was published using Chile, the Alboran Sea, and the Western Azores–Gibraltar fault. To further understand how feature selection could affect the quality of the predictions, Panakkat (Panakkat & Adeli, 2007) and Reyes (Reyes et al., 2013) parameters were calculated. The feature selection was based on the information gained from each one given a class. The label was a binary value indicating whether an earthquake higher than a threshold magnitude would occur in the following days. The set of best parameters varies with region, typically a mixture of elements of both sets, Reyes and Panakkat.

To further understand the results, multiple models were implemented, for each of the studied locations, using the best set of parameters. In the Chile region, comparisons were made over ANN, Support Vector Machine, Naïve Bayes, and KNN. For the second area, the Iberian Peninsula, the KNN model was replaced by M5P.

In general, the result cemented two main points: first, an increase in performance as a result of feature selection, and, second, the ANN has a promising model out of the studied ones. (Martínez-Álvarez et al., 2013)

2.7. RESEARCH GAP

To the authors knowledge, the mentioned papers were the only studies performed over Portuguese territory. This study aims to proceed the studies referred by exploring a broader set of algorithms has it been has done for other territories, and experiment with multiple methodologies, it is also the author intention to include the whole Portugal territory since some regions on the mainland have moderate seismicity risk and a record of catastrophic events with a devastating impact.

3. METHODOLOGY

This chapter explores into the methodological fundamentals of this study, outlining the chosen approaches and the reasoning behind them. To ensure transparency and reproducibility, the specific steps taken are detailed in this chapter. The code associated with these techniques is also readily available on GitHub for further exploration.

3.1. DATA SOURCE

This study retrieved its data from the Euro-Mediterranean Seismological Centre ¹ (EMSC), which collects worldwide time-series earthquake information. Selection of the region of interest involved choosing an area with a rectangular shape where each pair of vertices corresponds to a coordinate set.

Following the previously introduced seismicity risk studies performed over Portugal, the chosen area also includes a part of Spanish and Moroccan territory (Carvalho & Malfeito, 2018). The coordinates and time frame used are shown in Table 3.1. and a geographical distribution of the data is shown in Figure 3.1.

Table 3.1 – Input for data selection on the EMSC earthquake catalogue

Latitude	Longitude	Time
[32.19; 42.37]	[-34.45; -5.89]	[2024-04-15 23:59:59; 2005-01-01 00:00:00]

3.2. PRE-PROCESSING: DATA FRAGMENTATION AND EXPLORATION

The prediction of an earthquake in such a broad area as the selected one raises multiple concerns. Firstly, without restricting the location of the event, the warning is not very useful. Using coordinates alone cannot tackle this issue, as they offer overly specific information, which harms the training phase by leading it to focus too much on this data (Alarifi et al., 2012). Secondly, the type of activity through Portugal is quite distinct, as explained in a previous chapter; as such, the seismic indicators calculated over the broad region selected may not be good representations of the activity.

To mitigate the presented issues, a clustering step over the coordinates is added to this research; since it's a simple 2-dimensional cluster, K-means is the algorithm chosen.

¹ https://www.emsc-csem.org/Earthquake_information/

The choice of using only 2 clusters outcomes from the trade-off between acquiring a more insightful input vector alongside a more precise location of the predicted earthquake and the number of data points available. The geographical distribution of the different clusters is shown in Figure 3.1.

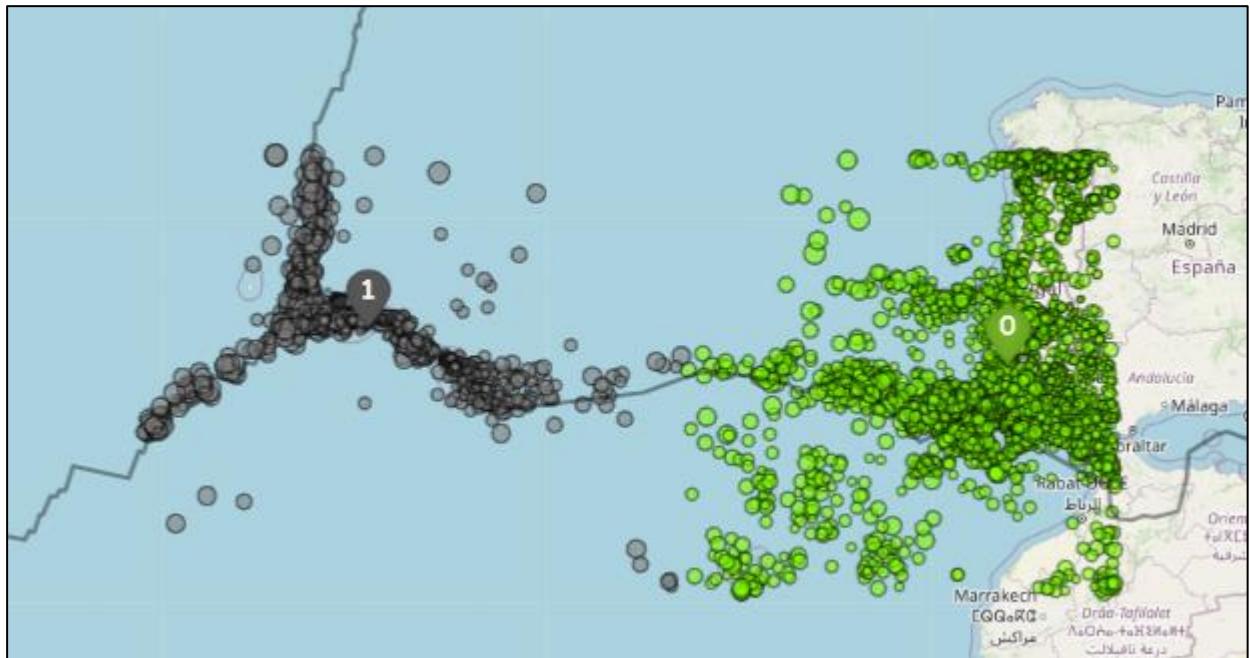


Figure 3.1 - Geographical distribution of the retrieved data after application of K-Means with $K = 2$, each point represents an earthquake, grey on the Azores region (Cluster #1) and green on Mainland and surroundings (Cluster #0), and the grey lines the fault lines. Plot computed in Python using Folium library on the function `plot_spatial_distributon`.

Furthermore, for each of the two clusters, a shorter period was also considered besides the full interval, to assess if the restriction to more recent data has any meaningful impact, the chosen interval was from 2020 to 2024. Summarizing, at this point 4 datasets for study exist, 2 per region – Azores and Mainland, one on the full-time interval, and the remaining one considering only the 4 most recent years.

Additionally, to ensure proper data assembly, the magnitude of completeness, M_c , was studied. This value indicates the minimum magnitude at which a seismic catalogue can be considered complete and of sufficient quality for seismic studies.

Although considerable methods exist to calculate this value, the choice was to follow the revised literature and use the Gutenberg-Ritcher law. This method lies in analyzing the frequency-magnitude distribution, FMD, and identifying the magnitude from which the plot can be explained by a power law, as expected via the law of Gutenberg and Ritcher (Asim et al., 2017). In this study, the frequency analysis results in the grouping of the magnitude in

intervals with an increment of 0.5 between 0.0 and 9.5. For each of the datasets, the FMD plots are shown in Figure 3.2.

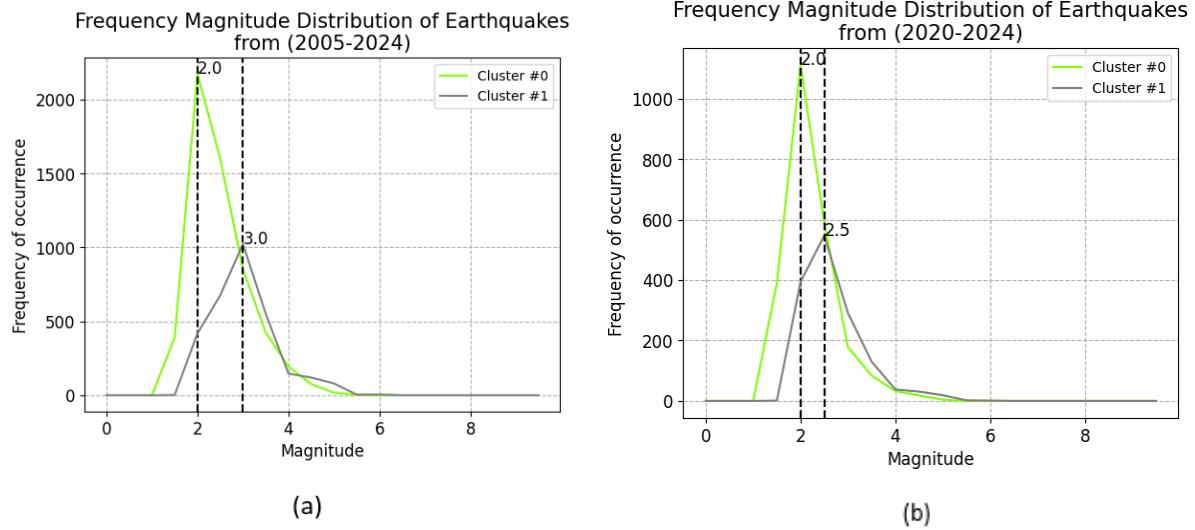


Figure 3.2 This figure presents the Frequency -Magnitude distribution per cluster, along with the indicated magnitude of completeness (M_c) for each cluster. The gray lines represent Cluster #1 (Azores), and the green lines represent Cluster #0 (Mainland and surroundings). Panel (a) depicts the distribution for the complete dataset, while panel (b) focuses on the data from 2020 onwards. Plots computed using the matplotlib library in Python under the function *freq_mag*.

With the regions defined for the remaining of the study and the catalogues complete, a threshold magnitude for the earthquake prediction must be set. The choice of this value, defining the minimum magnitude threshold for predicting earthquake occurrence, is dependent on the data available. Ideally, high-magnitude events would be the target of this study but it must be considered that if there are no events or a too-low amount present then such target is unfeasible.

To ensure an appropriate threshold choice, the magnitude distribution through the years was plotted for each dataset (Figure 3.3) to inspect which magnitude would set a realistic goal, preserving the objective of forecasting events with a high enough magnitude to affect society. The analysis of the plots led to the choice of a threshold magnitude of 4.0, considered a magnitude that balances potential social impact with data availability.

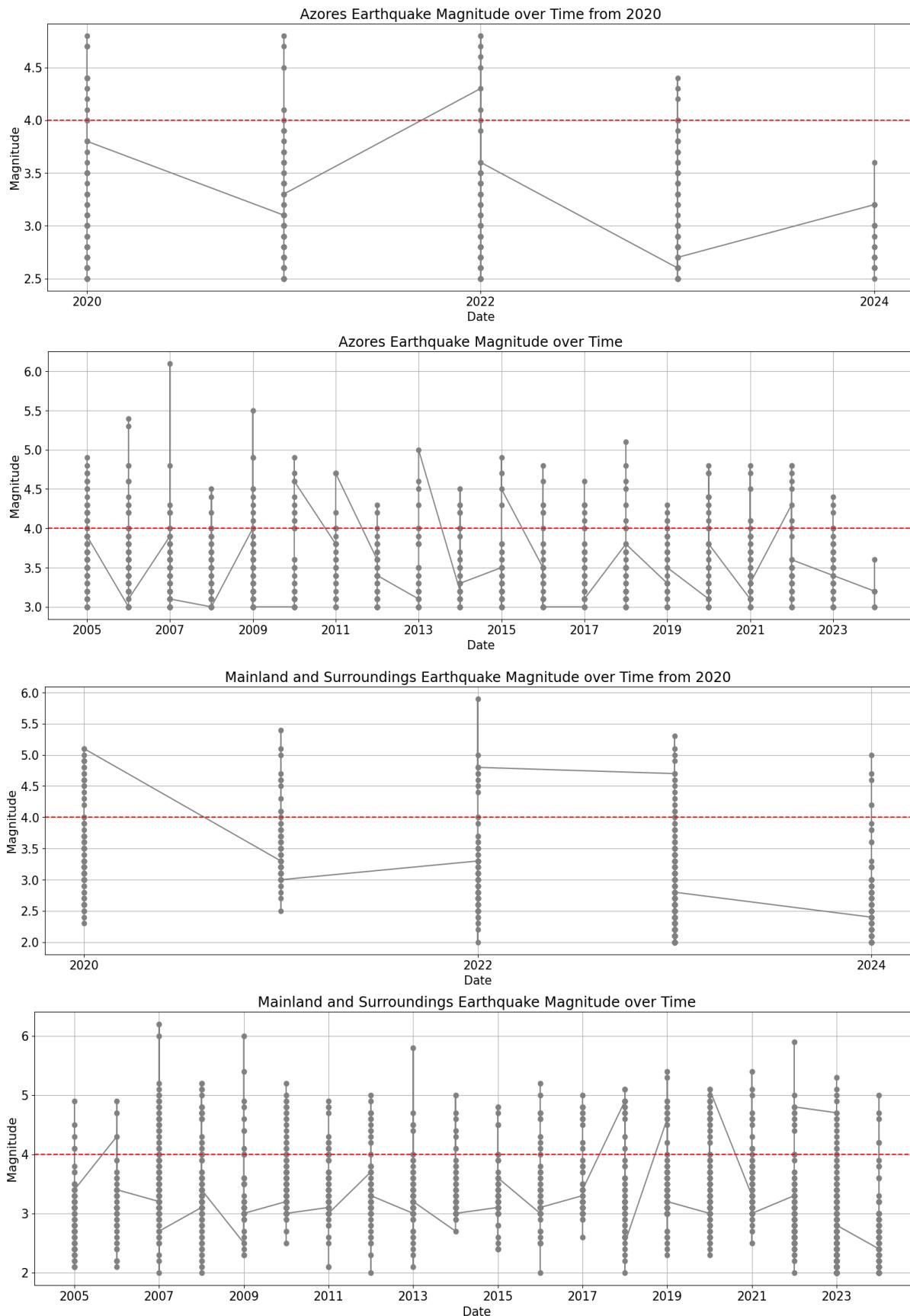


Figure 3.3 Magnitude distribution over the years for each dataset. With magnitude 4.0, the threshold value for label 1, highlighted by a red horizontal line. Plots computed using Python and Matplotlib library under the function *plot_mag_time*.

3.3. FEATURE ENGINEERING: INPUT VECTOR CREATION

Feature extraction, a step in feature engineering, consists of the process of building new features from the available data and pre-processing them to ensure an input vector with relevant information and quality. This time-consuming procedure has a high influence on the final results obtained.

For this study, in this step, multiple new parameters were calculated following the same rationale as some of the literature presented. The details of those calculations and the subsequent treatment applied are presented in this section.

The trigger to the subsequent math can follow two paths (Asim et al., 2018):

- (1) The calculations are executed using the last n events (including the ones in the window) every time there is at least one occurrence in the time window of Δt days under analysis. From now on this approach is denominated occurrence method.
- (2) The calculations are executed independently of a new occurrence in the time window under analysis, even if no event was registered an input vector is still computed using the n most recent earthquakes. From now on this approach is denominated the time method.

Both of the windows under analysis slide through the dataset by adding Δt to the last day. The prior events used in both approaches merely constrain the dataset used for each calculation. In both methods the label is generated considering the following Δt days starting at the end of the time window – this is the next current window.

To ensure the reader has a clear understanding of both methods two simple examples can be found in Appendix A.

The first method is limited by the need for an occurrence to generate an input vector, this is, it is only possible to forecast events if, in the prior Δt days, an event was registered. The second one overcomes this issue since the input vector is independent of new events, but at the cost that it is possible to have the same vector associated with different outcomes, this is generally known for possibly harming the learning process. As such, for the second method 3 extra features are computed to prevent duplicates, these values account for the time – week, month, and year of the window under analysis.

The relative efficacy of each method remains undetermined. In the interest of establishing the most robust approach, this research will leverage both methodologies and conduct a comparative analysis in an attempt to identify if one should be favoured.

3.3.1. Seismic Indicators

The input vector comprises seismic indicators, whose value offer insightful information of the geodynamics of the region based on the most recent n events. The mathematics used in this study aligns with revised literature and utilizes a sliding window of 50 prior events (n) as suggested by other researchers.

Elapsed Time, T

T refers to the time span over the last n events with a magnitude higher than a predefined threshold. In this study, that value will correspond to the cut-off magnitude of the catalogue studied (Last et al., 2016).

This value offers insight into the frequency of earthquakes over time in the seismic region. A small T translates that n number of events occurred in a short time interval.

$$T = T_i - T_1 \quad (3.1)$$

Where i refers to the i^{th} seismic event and 1 to the first on the n-sized sliding window.

Mean Magnitude, M_{mean}

This indicator refers to the mean of the Ritcher magnitudes over the last n events. This indicator is insightful since rising magnitudes can be a precedent for a large-scale earthquake (Asim et al., 2017).

$$M_{mean} = \frac{\sum_{i=1}^n M_i}{n} \quad (3.2)$$

Rate of Square Root of Seismic Energy Released, $dE^{1/2}$

This parameter represents the rate of the square root of energy released over the elapsed time, T. Seismic areas maintain equilibrium on their faults through tension release in the form of frequent small-magnitude earthquakes over time. The absence of these over a long period of time and the low energy released is, in some scenarios, a warning sign for an impending, high-magnitude event (Panakkat & Adeli, 2007).

$$dE^{\frac{1}{2}} = \frac{\sum_{i=1}^n (10^{11.8+1.5M_i})^{\frac{1}{2}}}{T} \quad (3.3)$$

Slope of Gutenberg-Ritcher Curve, b – value

In 1954, Gutenberg and Ritcher, two renowned seismologists, published a study that introduced one of the most well-known geophysical laws. The researchers established a relationship between earthquake frequency and magnitude, described by a logarithm relationship (Gutenberg & Richter, 1944):

$$\log_{10}N_i = a - bM_i \quad (3.4)$$

Where N_i is the number of prior events with magnitude higher or equal to M_i and constants a , y-intercept and b , the slope.

The b-value embodies geophysical aspects of the region, with high values associated with a low resistance area, hence the frequent occurrence of small-magnitude earthquakes. On the other hand, a low value indicates often high-magnitude events; this is a high resistance area (Asencio-Cortés et al., 2017). This indicator has been widely studied as an earthquake precursor, with research stating changes in the b-value, such as an intermediate increase followed by a possible decrease in the value, possibly indicating an impending high-magnitude event (Morales-Esteban et al., 2010).

To determine this parameter and the a-value, the academics followed the least squares method, resulting on the following equations.

$$b_{lsq} = \frac{n \sum_{i=1}^n (M_i \log_{10} N_i) - \sum_{i=1}^n M_i \sum_{i=1}^n \log_{10} N_i}{(\sum_{i=1}^n M_i)^2 - n \sum_{i=1}^n M_i^2} \quad (3.5)$$

$$a_{lsq} = \frac{\sum_{i=1}^n (\log_{10} N_i + b M_i)}{n} \quad (3.6)$$

Although commonly used, research shows this method is sensible to occasional large magnitude events. A more resilient approach suggested is the application of maximum likelihood method instead, presented next (Shi & Bolt, 1982).

$$b_{mlk} = \frac{\log_{10} e}{\frac{\sum_{i=0}^{n-1} M_i}{n} - M_c} \quad (3.7)$$

$$a_{mlk} = \log_{10} N + b_{mlk} M_c \quad (3.8)$$

With M_c , cut-off magnitude.

Standard Deviation of Gutenberg-Ritcher, η

This parameter offers a quantitative method to evaluate how suitable the Gutenberg-Ritcher distribution is for the observed data. Low values indicate a proximity between these two and, as such, a strong conformation with the inverse power law (Asim et al., 2017; Panakkat & Adeli, 2007).

$$\eta = \frac{\sum_{i=1}^n (\log_{10} N_i - a_j - b_j M_i)^2}{n - 1} \text{ with } j = [lsq, mlk] \quad (3.9)$$

Magnitude deficit, ΔM

The deficit is defined as the difference between the highest magnitude registered and its expected value over the pre-defined number of prior events (n).

$$\Delta M = M_{max,observed} - M_{max,expected} \quad (3.10)$$

Following Gutenberg-Ritcher law and the approximation that a high-magnitude event is likely to occur once in a window of n events (Panakkat & Adeli, 2007), then the expected magnitude is:

$$M_{max,expected} = \frac{a_j}{b_j} \text{ with } j = [lsq, mlk] \quad (3.11)$$

Mean time between characteristic events, μ_θ

This indicator is defined as the elapsed time between consecutive characteristic events occurred over the last n events considered. The term characteristic events, in this study, refers to the grouping of magnitudes in certain ranges, Θ – in this case, due to the reduced dataset, the intervals consider will be the rounding of the magnitude to the closest integral number as such: $\Theta = \{1,2,3,4,5,6,7,8,9\}$. It's expected especially for large magnitudes that the mean through time doesn't deviate much.

$$\mu_\theta = \frac{\sum_{i=1}^{n'} (t_i - t_{i-1})}{n'} \quad (3.12)$$

With n' symbolizing the number of events found over the n-sized window studied. This calculation will result in 9 new columns and when $n' = 0$ the results is set as 0.

Coefficient of variation of the mean time between μ_θ , c_θ

For each of the μ_θ calculated its standard deviation from the actual data is taken as a seismic indicator. Representing the conformance between the data and the characteristic distribution with high values indicating the seismic zone under study is not well represented by this distribution.

$$c_\theta = \frac{\sqrt{\frac{1}{n'} \sum_{i=1}^{n'} (t_i - t_{i-1} - \mu_\theta)^2}}{\mu_\theta} \quad (3.13)$$

Increments of b-value, x_i

Based on Reyes' proposed methodology, the initial five increments of the b-value with a step size, s, are considered seismic indicators. Following Reyes' study, the step is set to 4.

$$\begin{aligned} x_{1i} &= b_i - b_{i-s} \\ x_{2i} &= b_{i-s} - b_{i-2s} \\ x_{3i} &= b_{i-2s} - b_{i-3s} \end{aligned} \quad (3.14)$$

$$x_{4i} = b_{i-3s} - b_{i-4s}$$

$$x_{5i} = b_{i-4s} - b_{i-5s}$$

Maximum magnitude felt in Δt , x_{6i}

Adapted from Reyes, the parameter x_{6i} in this research will represent the maximum magnitude felt in the time interval defined for the study. This alteration prevents the loss of information in the scenario of the $\Delta t > 5$ days, which was the used interval by the researcher.

$$x_{6i} = \{\text{Max}(M_i)\} \text{ when } t \in [0, \Delta t] \quad (3.15)$$

In addition to the seismic indicators presented three additional parameters representing the time of the occurrence are added. The reasoning behind this decision, as mentioned before, is to differentiate the input vectors in the scenario of no new occurrence.

The temporal indicators computed when following the second method are:

Year, Month and Week

These indicators, each a position on the input vector, represent, correspondingly, the year, month and week number of the last day on the current vector.

3.3.2. Labelling

Alongside the creation of new features, the computation of labels is also done. In each presented method, the target variable associated with each input vector is determined by the following Δt days. The binary value goes according to the following questions:

- (1) Are there any events in the next Δt days?
- (2) Is at least one having a magnitude higher than 4.0?

If both conditions verify the target is set to one, else it's a zero.

This process is simultaneously executed with the calculation of the seismic indicators on the function '*seismic_indicators_label_df*'.

Summarizing, eight different studies are executed per region as shown in Figure 3.4, with each of the sliding window methods generating a different number of samples and imbalance which may impact their performance, the general characteristics of each method on the different datasets are shown in

Table 3.2. Since the maximum likelihood and least squares methods only affect the calculation of the seismic indicators the number of samples and total positive events is the same between them.

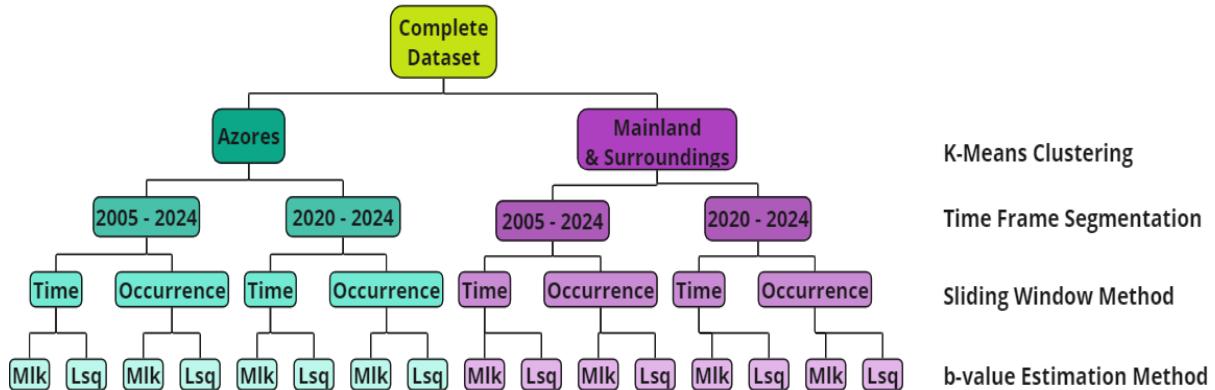


Figure 3.4 Overview of the datasets constructed on the earthquake data for the two regions: Azores and Mainland Portugal. Eight studies were conducted by region, each a combination of time-frame segmentation, sliding window technique and method for b-value estimation. This results in a total of 16 datasets for research on this study.

Table 3.2 General information on the effect each sliding window methods, occurrence and time, have on each dataset in terms of size and percentage of positive events.

Dataset	Occurrence Method		Time Method	
	Size	Total Positive Events (%)	Size	Total Positive Events (%)
Azores	501	19.76	673	20.21
Azores [2020,2024]	116	14.66	117	14.53
Mainland	555	25.95	670	25.52
Mainland [2020,2024]	100	24.00	118	25.42

3.4. EXPERIMENTAL DESIGN

This research can be segmented into two main sections. In the first stage, a comprehensive exploration is conducted using various machine learning models on the constructed datasets. The second stage focuses on neural network architectures. A generalized model is designed and then further optimized (tuned) for each of the available sets within the respective regions.

This section is divided accordingly with an introduction to each approach followed by an overview of the experimental design.

For both cases, the data was split respecting the time order with the first 70% samples for training and the remaining for testing.

3.4.1. Machine Learning Exploration

Algorithms Choice

All the algorithms presented in this section were imported from the *Scikit-learn* library, known for its user-friendly interface.

The reasoning behind each model chosen is to introduce a new logic and reasoning to this study, from simpler approaches to more complex ones. In the end, allowing to conclude from multiple machine learning algorithms families if some are more appropriated for the data.

As such, K-nearest neighbors; Naïve Bayes, and Support Vector Machine are implemented, falling into simpler approaches. Introducing, respectively, the logic of instance-based; probabilistic, and support vector machine approaches.

Perceived as more complex options are the ones using ensemble methods, which combine multiple simple models to improve performance. The combination of such models can follow a bagging or boosting approach. In this study, Random Forest (bagging) is explored, an algorithm that leverages tree-based classifiers.

A brief explanation of each of the presented model's rationale can be found in the next chapter.

Hyperparameter Tuning: Grid Search with a Pre-Processing Pipeline

Each of the mentioned algorithms is deployed by implementing a grid search, making use of the *GridSearchCV* made available by *Scikit-learn*. This step performs an exhaustive search of all possible parameter combination given a search space, a dictionary structure where the estimator's name and possible values are specified. The combination that results in the best scoring validation metric (in this study, the $\beta - f1$) is selected and deployed on the test set.

Given that the chosen algorithms do not account for the order in which the input vectors are presented, it is possible to apply cross-validation during the grid search process. The fragmentation of the data into K sets with each one being used once as the validation, ensures a more robust solution and can have a high impact on the learning process, especially when using small datasets as in this study. Considering the advantages associated with this technique, particularly when using a large K value, this research fixed that parameter to 10 folds.

To this process, a pre-processing pipeline is added, as well, with two objectives, the scaling and feature selection of the data based on each training set. The scaler chosen was *StandardScaler()* and the feature selection is executed over Recursive Feature Elimination (RFE) with a *DecisionTreeClassifier()*. Since the number of selected features yields great importance in the training process the parameter was added as an estimator to the grids' search space, making it possible to optimize the number of chosen features.

The datasets used in this study contain an uneven target distribution (Table 3.2), which can lead to models that prioritize the frequent class and perform poorly on the rare ones. To handle this automatically across the different datasets, sample weights are used during model training (exception of KNN due to library-imposed limitations). Since the data splitting during the grid search preserves the target distributions, the weights are calculated based on the proportion of each target class within the full train data. Specifically, each sample's weight is inversely proportional to the number of examples in its class. This approach ensures models pay more attention to the under-represented class.

3.4.2. Neural Network Exploration

Architectural Choice

In contrast to the previously discussed models, Recurrent Neural Networks (RNNs) offer the advantage of handling sequential data inputs. This capability is particularly relevant for this research, as it may significantly improve prediction accuracy by accounting for temporal dependencies within the data. To exploit this benefit, this study incorporates such algorithms to assess if considering these sequential relationships enhances the quality of results. This task is deployed utilizing *Keras* and *TensorFlow* libraries.

To provide a foundation for understanding this choice, the next chapter offers a brief introduction to various neural network structures and their overall functioning.

The chosen architecture results from a simplification of the revised literature (Banna et al., 2021), given that the volume of datasets and hyperparameter tuning phase are associated with a high computation expense, one architecture that showcased promising results in a study with the same data type and similar input vectors is used. With not much optimization being tried in this regard, the details of this architecture alongside some experimental settings can be found in the next chapter.

Hyperparameter Tuning and Pre-Processing

In order to preserve the time order, relevant to this algorithm, the validation set consists of the last 25% rows of the training set. Additionally, in terms of pre-processing, a scaling phase is deployed using the *StandardScaler* (). To capture the time dependencies the input data is reshaped into a matrix where each column is a feature and each row part of a temporal sequence with the last row being the vector for prediction. In this research, the sequence size chosen was 5.

A grid search, a systematic search technique, is employed to optimize the general model architecture for each study. Following the grid search across all possible hyperparameter combinations, the best model is determined through the validation set performance on the $\beta - f1$ metric. Subsequently, the hyperparameters which result in the highest score on the chosen evaluation metric are deployed on the test data for final performance evaluation.

The presence of a different degree of imbalance on each dataset is dealt with in a similar way as previously, with the weight of each label corresponding to the proportion of the opposite one on the training set.

3.5. QUALITY METRICS

The correct results analysis is ensured by the use of adequate quality metrics to measure the model's performances, in this research, as mentioned before, some of the datasets used present a certain level of target imbalance. Furthermore, in the problem context a high number of false alarms decreases the confidence from authorities in the warnings produced by these models. All of these aspects must be taken into consideration when choosing the metrics, since an unadjusted metric may give deceiving results which don't consider the issues presented.

The simplest metrics, elements of a confusion matrix, computed to gather a general notion of the model performance are the following:

1. True Negatives (TN): Correctly predicts a non-earthquake event.
2. False Negatives (FN): Misses an actual earthquake.
3. True Positives (TP): Correctly predicts an earthquake event.
4. False Positives (FP): Issues a false alarm for an earthquake.

With the ideal scenario translating into 0 false positives and false negatives.

While confusion matrices provide a real-world perspective on model outcomes for a single dataset, their insensitivity to imbalanced data and inflation of values in larger datasets make them unsuitable for direct comparison between studies with varying data sizes. For these reasons, the following metrics are introduced.

Sensitivity, S_n (%)

Measurement of the ability to correctly predict the positive outcomes.

$$S_n = \frac{TP * 100}{(TP + FN)} \quad (3.16)$$

Specificity, S_p (%)

Measurement of the ability to correctly predict the negative outcomes.

$$S_p = \frac{TN * 100}{(TN + FP)} \quad (3.17)$$

Usually, there exists a trade-off between this metric and sensitivity, meaning the increase of TP positive prediction may come at the cost of an increase of FP and vice-versa, especially in cases of imbalance, for this reason the two metrics must be plotted and the trade-off between them considered when measuring a model's performance.

For both metrics a higher value translates in a more robust solution. A balance between the metrics is desired since the goal is to gather true predictions but not at the cost of an increase of false alarms (FP).

P0(%)

Measurement of a negative prediction reliability, if came at the cost of high number high false negatives.

$$P0 = \frac{TN * 100}{TN + FN} \quad (3.18)$$

P1(%)

Measurement of a positive prediction reliability, if came at the cost of high number of false positives.

$$P1 = \frac{TP * 100}{TP + FP} \quad (3.19)$$

Ideally, both metrics would have a maximum score, but similar to the previous pair there is usually a trade-off between the two. As mentioned before, a high false alarm rate deems a model has untrustworthy as such a high value of P1 is prioritized overall.

Summarizing, sensitivity and specificity measure the proportion of well classified targets from one class. While P0 and P1 overlook the proportion and focus on if the correct classification came at the cost of a high misclassification of the target.

$\beta-f_1$ Score, $\beta-f_1$ (%)

The metric used to decide the better performing and tune the models is $\beta - f_1$ score, which isn't sensitive to imbalance and its single value represents the balance desired between metrics. As shown in Equation (3.20) the β parameter adds a weight factor which allows to tune the metric to favor the reduction of false positives through a β lower than 1, in this research the value used was of 0.5, this decision derived from β values experimentation.

This metric allows for a simpler tuning process and also an easier comparison between the multiple models to select the best ones.

$$\beta - f_1 = \frac{(\beta^2 + 1) * TP}{\beta^2(FN + FP) + (1 - \beta^2)TP} * 100 \quad (3.20)$$

Additionally, Friedman's Chi-square statistical test is deployed on the $\beta-f_1$ scores regarding datasets and models' performances to assist in identifying relevant differences. Upon

achieving a significant p-value, the post hoc Nemenyi test compares the parameters in a pairwise manner and ranks how much they differ on the given metric.

4. THEORETICAL AND EXPERIMENTAL SETTINGS

In this chapter a brief introduction to every model studied and its experimental settings can be found, along with their experimental settings, expectations and justification for its usage. The deep explanation of each model is out of the scope of this research, as such the referenced literature is indicated.

4.1. MACHINE LEARNING METHODS

4.1.1. K-Nearest Neighbours

K-Nearest Neighbors is based on the intuitive idea that data points with similar feature values are likely to belong to the same class. This notion applied to both, classification and regression problems.

Consider a set of training points represented in a n-dimensional space, where each axis corresponds to one of the n features. Given a distance metric the similarity between a new sample and every training data is computed, the k closest ones (neighbors) are selected and used to determine the class label. In a classification task such is done through majority voting, the most frequent label on the k neighbors is the predicted, on the other hand, for regression an average is used. This final step can also take into consideration weights so that more similar, closer neighbors, have a higher impact on the prediction (Vanneschi & Silva, 2023).

In general, this is a simple algorithm that doesn't attempt to generalize from the training data, it stores it and compares it with the new vector, for this reason, it is denominated a lazy learner. In addition, it is highly dependent on the hyperparameters. Despite these disadvantages, the algorithm is chosen since it doesn't demand too much computational power, may perform better than expected, and more importantly, allows the analysis of instance-based algorithms performance.

Experimental Settings

This classifier has multiple parameters which can be tuned on. The most obvious and previously presented parameters are the k number of neighbors to be considered, the *similarity metric* and the *weights* which determine the strength of each neighbors on the final decision.

Additionally, the *algorithm* is also considered on the search space, which offer a tree-based approach as an alternative to the previously described n-dimensional space (known as *brute*).

Table 4.1 Estimators subject to tuning on the KNN algorithm and the corresponding search space. Resulting in a total of 1024 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20.

Parameters	Search Space
Number of Neighbors, k	[1, 3, 5, 7, 9, 11, 13, 15]
Similarity Metric	[euclidean, manhattan, chebyshev, minkowski]
Weights	[uniform, distance]
Algorithm	[auto, ball_tree, kd_tree, brute]

4.1.2. Gaussian Naïve Bayes

The classifier as suggested by the name follows the rational introduced by Bayesian's Theorem. This theorem, based on conditional probability, offers a probabilistic approach to determine the probability of an event given prior knowledge of variables correlated to the outcome. For application to machine learning tasks, consider an event E which is represented by the input vector $E = \{x_1, x_2, \dots, x_n\}$, with x_i symbolizing each feature known and let's assume it can have two possible results represented has $C = \{+, -\}$, Bayes Rule is written as follows for classification:

$$P(C|E) = \frac{P(E|C)P(C)}{P(E)} \quad (4.1)$$

With $P(C|E)$ representing the probability of a class given the occurrence of E . In this example, would be necessary to calculate the likelihood of each possible outcome, $P(+|E)$ and $P(-|E)$, since the denominator is equal in both scenarios, it is irrelevant to the determination of the most probable target. The resulted prediction would be positive if $f_b(E)$ is higher or equal to one and negative otherwise (Zhang, 2004).

$$f_b(E) = \frac{P(+|E)}{P(-|E)} \quad (4.2)$$

This formula simple stating the higher probability determinates the most likely outcome. This reasoning is generalizable to multi-class classifications.

Unfortunately, although simple this theorem often comes with high computation expenses due to the calculation of each $P(E|C)$. Given this reason the variables are assumed to be independent which results in $P(E|C) = \prod_{i=1}^n P(x_i|C)$, this is a strong assumption which rarely happens in real-life problems hence the naïve (Berrar, 2018; Rish, 2001). Moreover, to calculate $P(x_i|C)$, assumptions on the distribution of feature values given a label are necessary, the most common one being the Gaussian which applies:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right) \quad (4.3)$$

With parameters σ_C and μ_C estimated using the maximum likelihood method during training, given a new data sample its class is determined, in part, by how well its features fit the distribution calculated.

Summarizing, this classifier offers a time efficient method of prediction and despite being a naïve learner this approach it's known for common good performances, in times comparable to complex and more computationally expensive algorithms. Even though, in the context of time-series, the variables independence approximation in addition to the lack of a time awareness mechanism may lead to less quality of results, the low time effort supports the inclusion of this technique.

For these reasons, this model is computed alongside the goal to introduce a classifier belonging to the family of probabilistic approaches.

Experimental Settings

For this classifier, only the *var smoothing* was tuned. This parameter is a numerical value which multiplied by the highest variance, is then added to the variance of each feature.

The var smoothing helps to stabilize the calculations, by mitigating the scenarios of small variance features given a certain target gaining more weight than desirable and disturbing the final outcomes.

Table 4.2 Estimators subject to tuning on the NB algorithm and the corresponding search space. Resulting in a total of 400 combinations considering the number of features is also tuned with the possible values being 5, 10, 15.and 20

Parameters	Search Space
Var smoothing	100 logarithmically spaced values in range $[10^{-1}, 10^{-13}]$

4.1.3. Support Vector Machine

Support Vector Machines consist of determining the best hyperplane to separate data samples into their corresponding targets. For the sake of an easier understanding and visualization, let's assume a target $C = \{+, -\}$ where each data point is described by two features, x_1 and x_2 , so represented in a two-dimensional plane, Figure 4.1 (a). The classification task is executed by defining a hyperplane that separates the data points according to their binary labels. To reduce sensitivity to noise and improve generalization to

unseen data, the plane chosen is the one that maximizes the margin, this is the separation between the closest data points of each class and the hyperplane itself. The points that lay on the margin are denominated support vectors (Vanneschi & Silva, 2023).

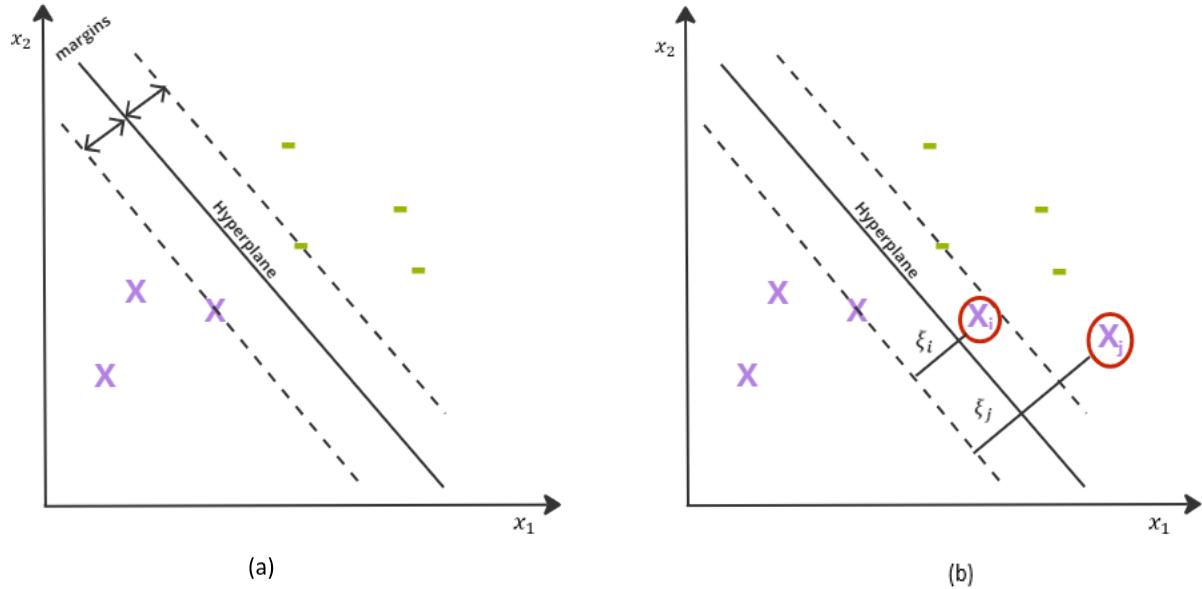


Figure 4.1 (a) Representation of a simple linear binary problem in a 2-dimensional space. (b) Introduction of a soft-margin representation to the problem.

The example given showcases an easily linear separable set of data, in reality, most cases found don't have that behavior. As such, a soft margin concept is introduced, Figure 4.1 (b), which allows for data points inside the margins or misclassifications, the distance between these to the correct margin is computed as a measure of control, ξ_i . A regularization term, C , is taken as input which controls the tradeoff between margin distance and the number of misclassifications and how off they are, the final objective being the maximization of the margins while minimizing the misclassified scenarios.

A high value of C leads to a decrease in misclassification cases but usually at the trade-off of a less generalizable model to unseen data points.

This reasoning allows for the application of multiple algebraic knowledge and constraints to deduce a function for classification, as shown next. The proof behind this can be found in (Vanneschi & Silva, 2023).

$$f(\mathbf{x}) = \text{sign} \sum_i \alpha_i C_i \mathbf{x}_i^T \cdot \mathbf{x} + b \quad (4.4)$$

Where x symbolizes the unseen point and x_i the data points. The b factor represents the penalty term, with the Lagrange factor α_i being zero for non-support vector machines or misclassified samples.

$$b = C_i(1 - \xi_i) - \mathbf{w}^T \cdot \mathbf{x}_i \quad (4.5)$$

The soft margin introduction is a helpful logic, but only if the data is near-linearly separable as in Figure 4.1 (b). In real life, cases of non-linearly separable problems are common, a simple example is given in Figure 4.2 (a), where a solution through SVM seems impossible. Nevertheless, a non-obvious solution is the mapping of the points in a higher dimensional space, Figure 4.2 (b). SVMs allow this transformation through a denominated *kernel trick*, which offers a solution that preserves the relationships among data points. Some of the most commonly used kernels are: polynomial and radial basis functions.

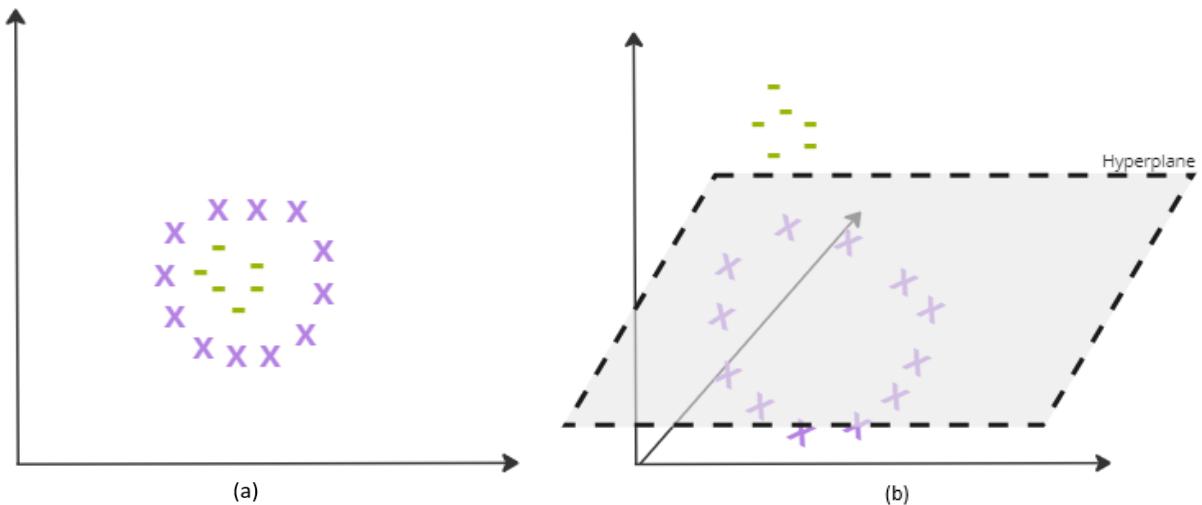


Figure 4.2 (a) Representation of a non-linear binary problem in a 2-dimensional space. (b) Representation of the same problem after transformation of the points into a 3-dimensional space.

Experimental Settings

The search space is composed of the parameters previously introduced, the regularization term, C , and the type of kernel transformation. In addition, the parameter gamma is also tuned, this value determines the weight of the influence an individual data point has on the hyperplane location, with higher values associated with a stronger local influence and a more complex decision boundary.

Table 4.3 Estimators subject to tuning on the SVM algorithm and the corresponding search space. Resulting in a total of 320 combinations considering the number of features is also tuned with the possible values being 5, 10, 15 and 20.

Parameters	Search Space
Regulation term, C	[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]
Kernel	[<i>rbf, linear, poly, sigmoid</i>]
Gamma	[<i>auto, scale</i>]

4.1.4. Random Forest

This ensemble predictor follows the bagging method, which generates multiple versions of a single simple classifier making use of training data subsets, with replacement. In parallel each model independently learns on its subsample; the conjunction of all versions is done through majority voting. This is, given a sample the label is the value a majority of the classifier predicts (Breiman, 1996).

In the Random Forest case, the single classifier is a decision tree. This base model as the name indicates follows a tree structure, which is represented in Figure 4.3.

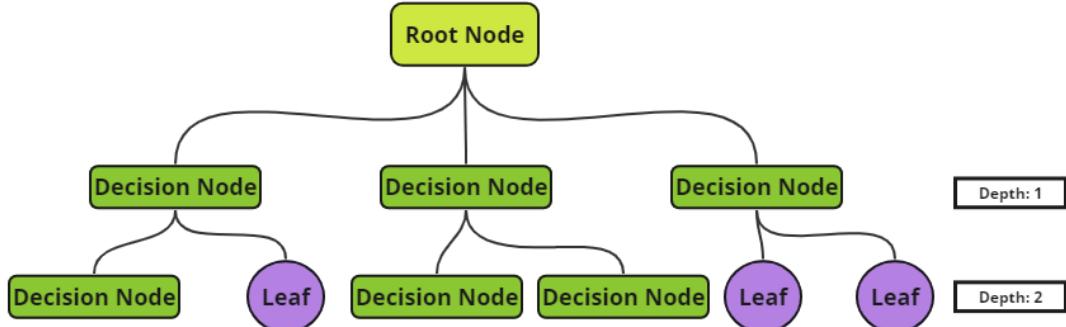


Figure 4.3 – Representation of the general structure of a decision tree – with depth 2.

The root node englobes all known data points present on the training set, in the following layer there is the fraction into sub-trees given decision nodes which impose conditions on at least one variable. At each one of these nodes, the algorithm evaluates the condition to decide which branch to follow afterward. This process finds its end either at a leaf node, when a prediction associated with the path is reached, or when a stop criterion is met.

During the training phase, a crucial step in the design of the described structure is the selection of the features and conditions at each decision node. This decision is led through an impurity measurement since at each step the goal is a homogeneous separation of the targets. This

measure is given as an input parameter, with the most common options being the Gini or the information gain.

The Random Forest, introduces in this study an exploration of ensemble bagging methods. It is considered more complex than the previously presented predictors, but its performance is highly dependent on the quality of the decision trees constructed which reinforces the importance of an efficient hyperparameter tuning which has a high effect on the quality of this classifier.

Experimental Settings

There are a broad set of parameters possible to tune on the random forest but since it comes with a high time consumption this study was forced to limit to the ones believed to be more relevant.

Prune techniques to prevent overfitting are considered essential as such the number of trees is tuned, with a focus on high values since by the Strong Law of Large Numbers this parameter leads to a decrease in the overfitting although at the cost of computational expense (Breiman, 2001). Additionally, the maximum depth is also considered to prevent the full growth of the trees until maximum purity is reached and the maximum number of features examined at each split, this step not only introduces more diversity and randomness between the different trees, and as such more robust solutions, but also reduced the time effort of the classifier.

Lastly, as mentioned before, the split criteria are one of the most important aspects of building decision trees and Random Forests, with a significant impact on the overall performance, as such it is also tuned.

Table 4.4 Estimators subject to tuning on the RF algorithm and the corresponding search space. Resulting in a total of 720 combinations considering the number of features is also tuned with the possible values being 5, 10, 15, and 20.

Parameters	Search Space
Number of Estimators	[90, 100, 150, 200, 250]
Maximum Features	[<i>sqrt</i> , <i>log2</i>]
Maximum Depth	[2, 3, 4, 5, 6, 7]
Criterion	[<i>gini</i> , <i>entropy</i> , <i>log loss</i>]

4.2. NEURAL NETWORKS

Artificial neural networks (ANNs) are computational models loosely inspired by the brain structure and function. Recurrent neural networks denote a type of architecture designed particularly to handle sequential data, making them the preferred architecture for this study.

To fully understand RNN architecture the learning process associated with artificial neural networks must be introduced. This process is easier explained with the use of a feedforward neural network, a family that englobes some of the simplest forms of architectures (Figure 4.4 (a)). The connections between each layer are done through weight vectors, a set of numbers, symbolized by line on the figure, which determine the influence of each node on the final prediction. Activation functions, applied after each layer, transform linear combinations into non-linear outputs, enabling the network to learn complex relationships in the data. Reaching the output layer, a loss function quantifies the discrepancy between the predicted and actual values, the learning process is then an optimization method with the goal of minimization of the loss function. The learning occurs, typically, through backpropagation which guides the weights and biases terms updates to minimize the error function. This process can be visualized by considering the loss function as a surface projected in a high-dimensional space, where each axis represents a single weight or bias term. Each point in this surface corresponds to a specific vector of weights, to find the point that corresponds to a minimum the surface is navigated through gradient descent which calculates partial derivatives concerning each axis-value, since it is known the gradient in case of a minimum is zero. Until an acceptable value is reached, a step of size defined by a learning rate is taken and the gradient of the encountered

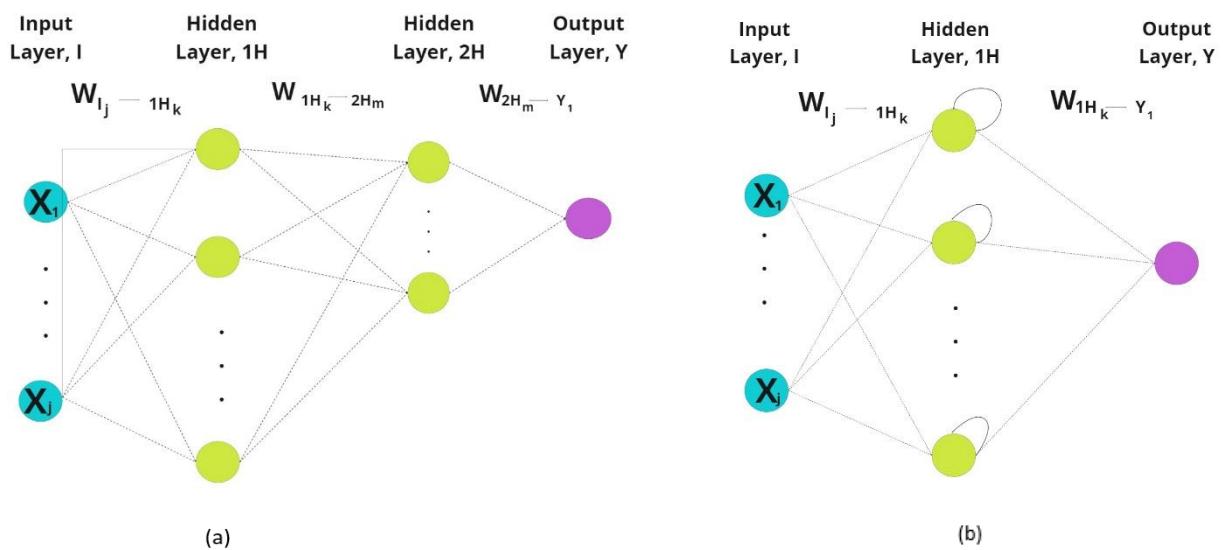


Figure 4.4 (a) Representation of an FNN receiving an input vector of size j with 2 hidden layers with k and m neurons, respectively. Each circle represents one neuron and the lines the weight connections between them, each representing one value which at the end of each layer is scaled through an activation function. (b) Representation of an RNN with 1 hidden layer having k neurons. The curved line representing the inner loops.

point is calculated. A deeper understanding of this learning process and its mathematics can be found in Vanneschi & Silva, 2023.

A Recurrent Neural Network (RNN) layer can be thought of as having an inner loop of interconnected feedforward networks, represented in Figure 4.4 (b) by the curved arrows and the unfolded RNN layer in Figure 4.5. These networks share the same weights (w_x , w_h and w_y) and biases (b_h and b_y) but differ from the simple FNN, that each layer receives not only the current input but also a hidden state, h_k , from the previous layer. This hidden state, captured by Equation 4.6, allows the network to learn temporal dependencies within a sequence. This value is subsequently used to compute the prediction at time t, as shown in equation (4.7).

$$h_t = f_1(w_x x_t + w_h h_{t-1} + b_h) \quad (4.6)$$

$$y_t = f_2(w_y h_t + b_y) \quad (4.7)$$

With f_1 and f_2 representing activation functions.

After processing the entire sequence, the weights are updated using Backpropagation Through Time (BPTT) which follows the same rationale presented before, but in this architecture type the weights to update are shared through the inner networks and layers are not independent as in the previous case, this alters slightly the backpropagation process. To capture the hidden layers dependency on the past and shared weights, the chain rule is applied which allows to encapsulate how the error at each time step is affected by these aspects. A demonstration of how dependencies affect the backpropagation calculation can be found in Goodfellow et al., 2016.

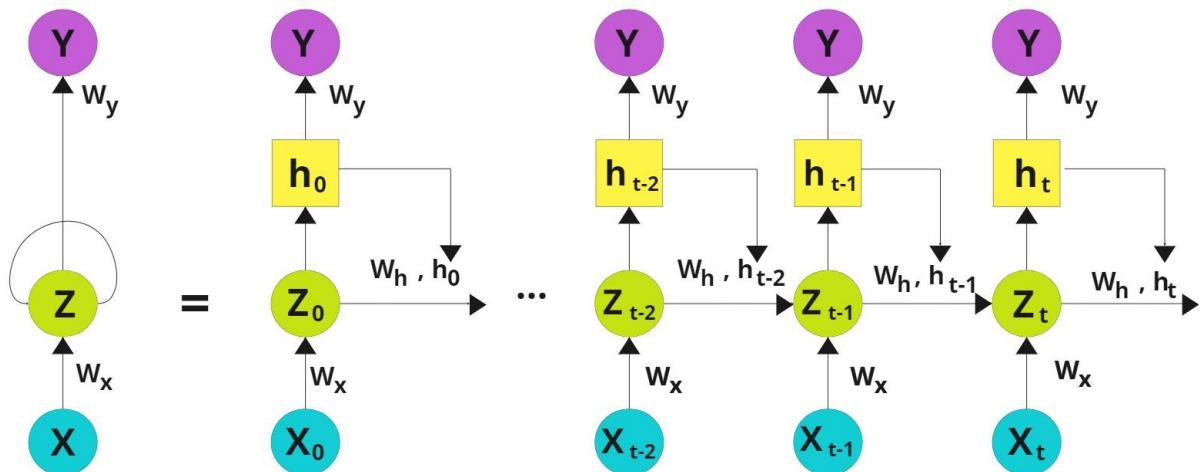


Figure 4.5 Representation of an unfolded RNN layer, representing the inner loop process that each RNN unit goes through. Processing a sequence of size t, where w_x , w_h and w_y represent, respectively, the input, hidden state and prediction weights. The Z symbolizes the hidden state, h_t , pre-activation function.

Long short-term memory refers to a specific type of RNN building blocks designed to capture long short-term dependencies and mitigate the vanishing gradient and issue arisen from the

inherent dependencies between hidden states. The gates constitute a fundamental element of the concept, it is through them that the flow of information is controlled, a general design of one cell is shown in Figure 4.6. The forget gate is fed the hidden state of the previous timestamp, h_{t-1} , and the input data of the current one, x_t . The combination of this information serves as input to a sigmoid activation function which maps each value to the range [0, 1], low values indicate that a certain past information does not add much value and high values the opposite. This process is described in Equation 4.8.

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \quad (4.8)$$

On the other hand, the input gate focuses on accessing which new information should be kept, such is done through a combination of a tanh activation function, which maps the values to the interval [-1,1], and again a sigmoid. The first output being a vector of candidates to be added to the cell state, Equation 4.10, and the last the selection of those who offer relevant information, Equation 4.9.

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \quad (4.9)$$

$$\check{C}_t = \tanh(W_c [h_{t-1}, x_t] + b_c) \quad (4.10)$$

The cell state, responsible for storing memory over time, is only fed the previous timestamps state. Its update results from the forget gate and the input gate, one responsible, respectively, for discarding non-useful information and the other to add the relevant information at the current timestamp (Equation 4.11).

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \quad (4.11)$$

Lastly the hidden state to be carried to the next timestamp is calculated using the cell state, C_t , which carries relevant information over time, h_{t-1} , the hidden state of the previous input and the current input, x_t . This information is combined as described by Equation 4.12. Following the previous RNN, this process continues until the last of the element of the sequence is reached, then the loss function is computed and backpropagation applied to the shared weights and biases, considering the gating system and different dependencies this math varies once again but the rational is preserved.

$$h_t = \sigma(W_o [h_{t-1}, x_t] + b_o) * \tanh(C_t) \quad (4.12)$$

Summarizing, these units allow for long short-term dependencies capture through the cell state and mitigates the vanishing gradient problem by utilizing gates to control information flow and enabling a direct information path through the cell state. These reasons, on top of the results in revised literature, make these units the main focus of this study.

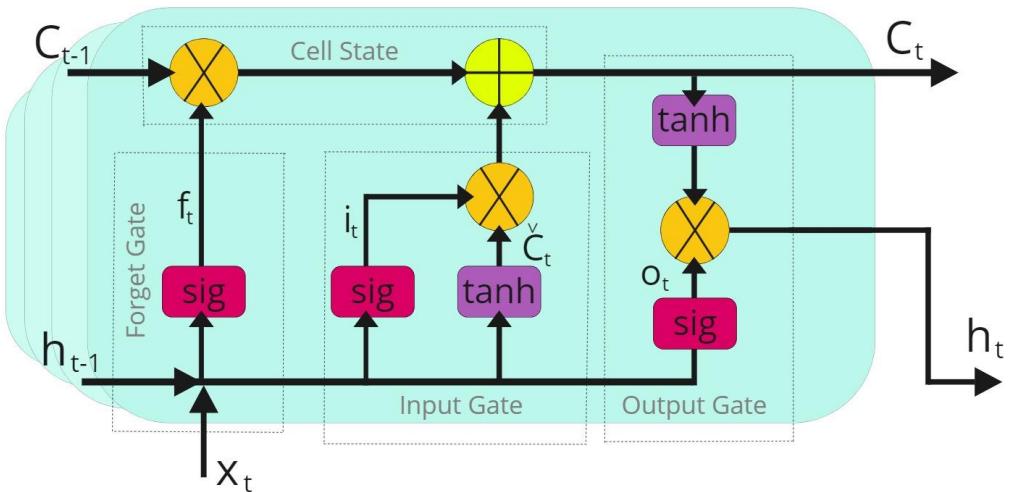


Figure 4.6 Representation of an LSTM unit, where the circles represent mathematical operations, sig and tanh, sigmoid and tanh activation functions respectively.

Experimental Settings

The architecture chosen, as mentioned before, follows the revised literature, a general description of the layers is shown in Figure 4.7. In this study, a maximum of 300 epochs for training is chosen due to the substantial computational cost associated with these algorithms and the comprehensive exploration required by the grid search and the number of datasets studied.

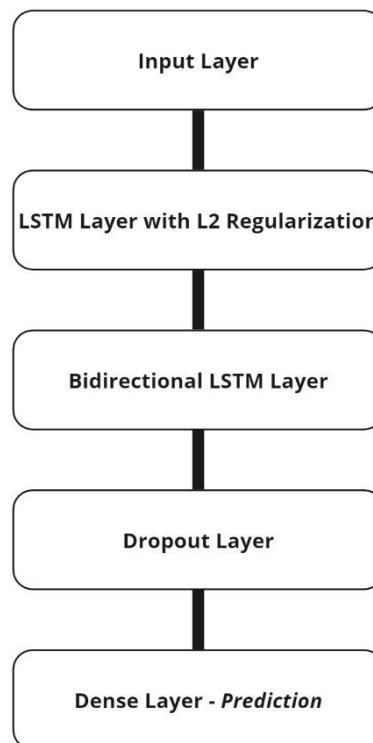


Figure 4.7 Description of the RNN general architecture applied to all datasets. With the Bidirectional layer being tuned on whether to be included or not.

The imbalanced nature of the problem demands an appropriate loss function, since the correct prediction of the majority class can overshadow the lack of predictions on the minority leading to an ambiguous update of the weights. Adding to this the binary nature of the target the chosen loss function was *BinaryFocalCrossEntropy*, which adds more weight to the wrongly predicted instances by means of a γ parameter, as shown in equation 4.13.

$$FocalLoss = -\alpha y(1-p)^\gamma \log(p) - (1-y)p^\gamma \log(1-p) \quad (4.13)$$

Where y represents the true label, p the predicted probability and α an hyperparameter that rules the trade-off between precision and recall. The default options are the ones implemented in this research, $\alpha=0.25$ and $\gamma=2.0$.

The hyperparameters subjected to tuning and the corresponding search space can be found in Table 4.5, given time constraints the search space is limited.

Table 4.5 Estimators subject to tuning on the LSTM algorithm per layer and the corresponding search space. With **0*** corresponding to the removal of the corresponding layer – Bidirectional. Resulting in a total of 288 combinations.

	Parameters	Search Space
First Layer	Number of Neurons	[150, 200, 250]
	L2 Regularization	[0.3, 0.4]
Second Layer	Number of Neurons	[0*, 100, 150]
Third Layer	Drop Out Rate	[0.2, 0.4]
General	Batch Size Rate	[10, 20]
	Learning Rate	[0.001, 0.0001]
	Optimizer	[<i>Adadelta, Adam</i>]

5. RESULTS AND DISCUSSION

In the following pages, the results achieved, alongside an analysis and hypothesizing, can be found. The results are presented by region, this is the Azores and Mainland results are analyzed separately since they represent two different geological areas.

Additionally, the results of the grid searches with the previously presented search space per model and dataset can be found in Appendix B, the features selected for each model per dataset are presented in Appendix C and the confusion matrixes and referred metrics in Appendix D.

5.1. AZORES

On the subject of methodologies, Friedman's Chi-square test on the $\beta-f_1$ metric indicates no significant differences, reaching a p-value of 0.95. It is noticed the lack of performance from KNN on the datasets which resulted from the combination of occurrence method and the mlk estimation method (Figure 5.1 and Figure 5.2), but since it is only observable in this model it is not considered a dataset limitation but rather an ill fit between model, hyperparameters and data.

The reason behind no overperforming methodology, regarding the time restriction, on the $\beta-f_1$ metric may be a result of more focused location associated with frequent events allowing for the computation of insightful input data even when only considering the last 4 years. In terms of the window methodology, time and occurrence, the prediction window of 10 days allied with a common events area doesn't translate in a high divergence between methods, with Table 3.2 showcasing the size and imbalance don't vary much. A smaller prediction window is expected to amplify the differences of both methodologies, in this case, the preferable method is unclear.

An analysis regarding the pair wise metrics (Figure 5.2) displays that overall the time reduction of the dataset appears to introduce more variability, wider boxes, on the model's performance, showcasing a stronger dependency on the algorithm. On the other hand, the full catalogue reduces that issue with more robust datasets where the models perform more similar in almost all metrics. On the complete catalogue it is also possible to identify the occurrence method as a steady approach with an average typically among the highest achieved on each metric.

In terms of model performance once again the Friedman's Chi-square test on the $\beta-f_1$ metric indicates no significant difference with a p-value of 0.72. In the pair wise metrics, it is possible to identify a strong tradeoff between sensitivity and specificity (Figure 5.2), introducing too much variability for a clear model comparison. On the other hand, in the P0 metric most models perform similar, this is the identification of true negative results does not come at a high cost of false negatives, which allows to then focus on the P1 results where it is identified

the Random Forest models achieves steady results, higher or equal than the average, across all datasets, this allied with a low sensitivity which translates into a low false alarm rate but not predicting much of the events, a scenario presented as reasonable given the task. Still Naïve Bayes and LSTM shouldn't be overlooked, even though they don't offer such a robust solution in certain datasets their performance is far superior on P1 and β - f_1 metrics (Figure 5.2).

On a side note, the LSTM model of the full catalogue with a time window and lsq methodology which achieves a perfect score on P1 and specificity (Figure 5.2) comes at a high cost of sensitivity, a deeper look into the confusion matrix (Figure D.9) shows it is only predicting 1 of 40 events.

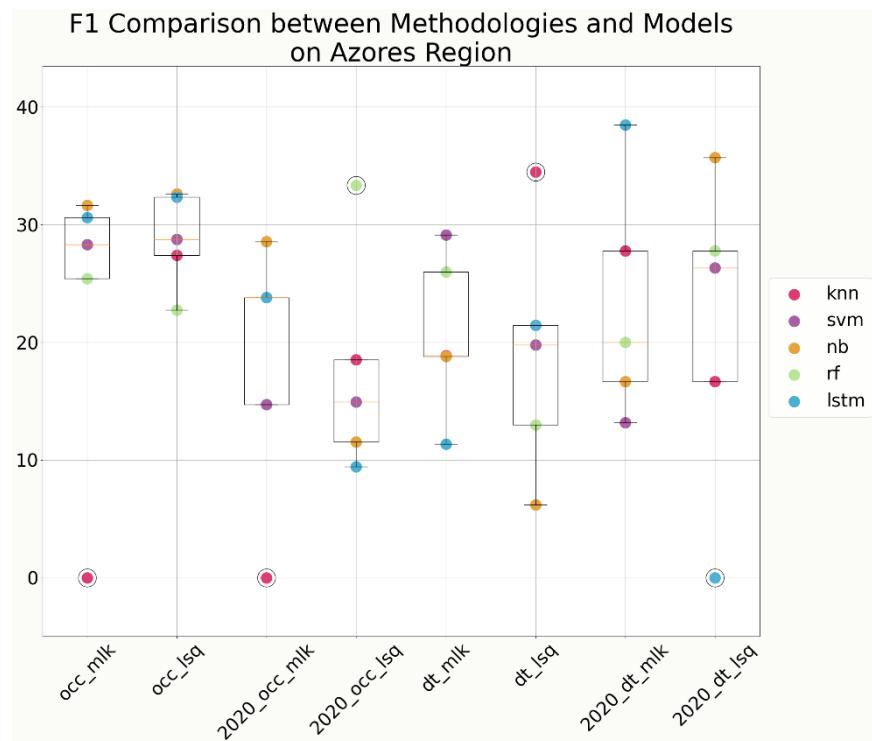


Figure 5.1 Boxplot of the β - f_1 metric on the Azores cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.

In the end, the best-found combination of model and dataset is the LSTM model receiving the last 4 years of data using the time window method and mlk estimation with a β - f_1 of 38.46% (Figure 5.1). As mentioned, this dataset introduces more variability but this may also result in better results with the right model as it is the case of this study hence the methodology approach preferable remaining unclear.

As mentioned before the Western Azores was previously studied in some of the revised literature, these results can be found in the Annex. The previous researchers set a lower

threshold magnitude and focus on a smaller region, but even with the higher magnitude threshold and broader area on this research the results obtained in the common machine learning algorithms, Naïve Bayes and SVM, showcased improvements in different metrics, especially on the P1. These results are found over the multiple methodologies studied.

Additionally, a M5P model posed some challenging results on previous studied, this algorithm belongs to tree-based family, but using a single classifier, the leveraging of ensemble methods allowed the achieving of better results than the ones obtained on this single classifier.

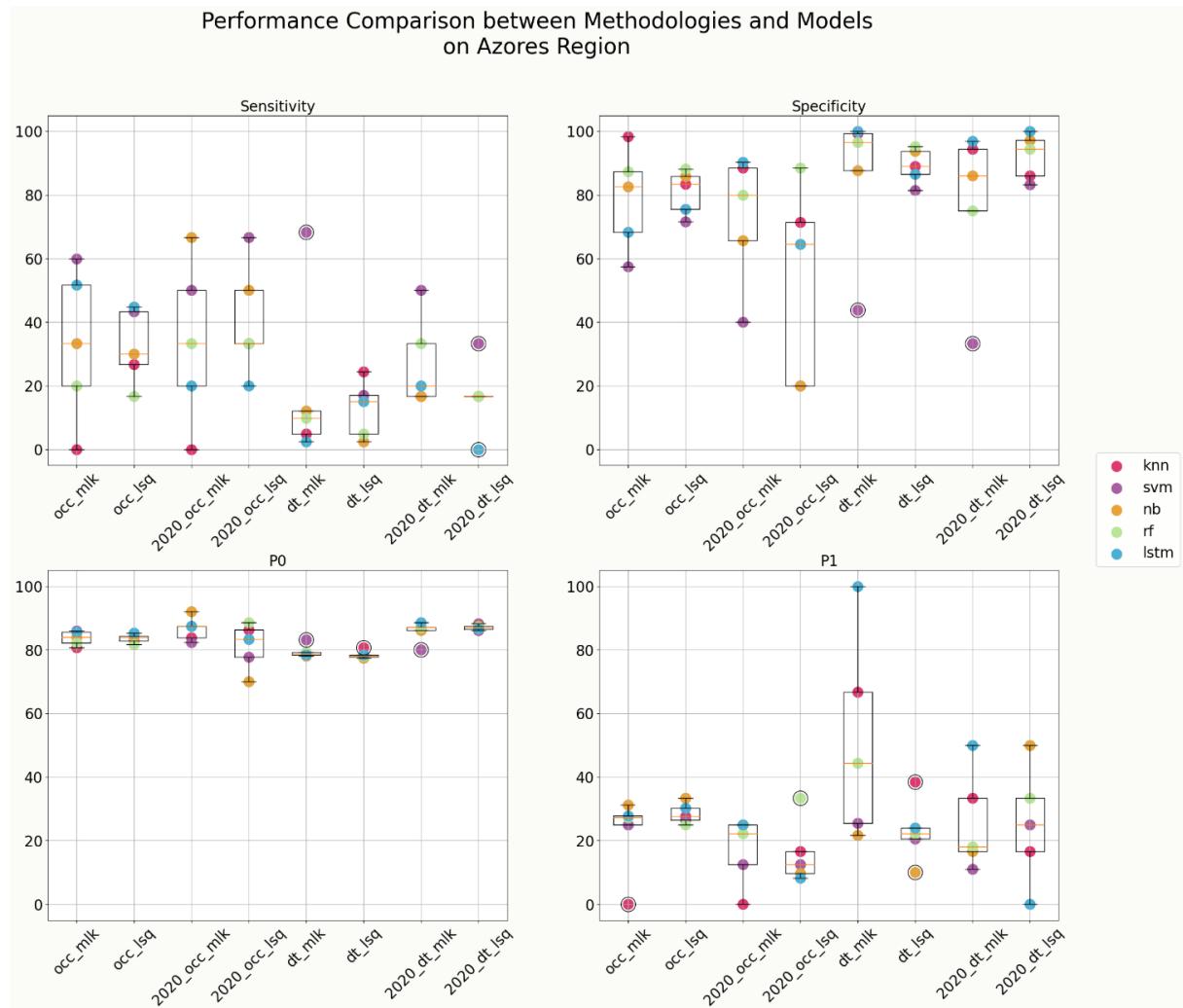


Figure 5.2 Boxplot of the pair wise results (P0 and P1, Sensitivity and Specificity) on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.

Despite this increase of performance in some models, the best performance for the region of Azores continues to be the artificial neural networks built on previous studies, reaching a better high-valued harmony between all metrics. Given the different methodologies, architectures and geographic area size it is unclear why ANN overperforms.

5.2. MAINLAND AND SURROUNDINGS

Regarding the mainland and surroundings cluster, the results in terms of methodology showcase that the time reduction approach when compared to the usage of a complete dataset on the P1 and β - f_1 metrics do not reach the same range of results (Figure 5.4 and Figure 5.5). With Support Vector Machine and K-Nearest Neighbors failing to make a true earthquake prediction in at least one of the reduced sets, the Random Forest and Naïve Bayes algorithms, on the other hand, do not achieve a single positive result in any methodology combination inside this reduction approach.

Given the goal of the study to forecast earthquakes and ensure a decrease of false alarms while doing so, it is concluded the datasets which only account for the last 4 years of data are unreliable. This result may arise from the lack of representative events in such a broad area deeming the computed indicators unable to capture the seismological activity of the different regions composing the mainland cluster.

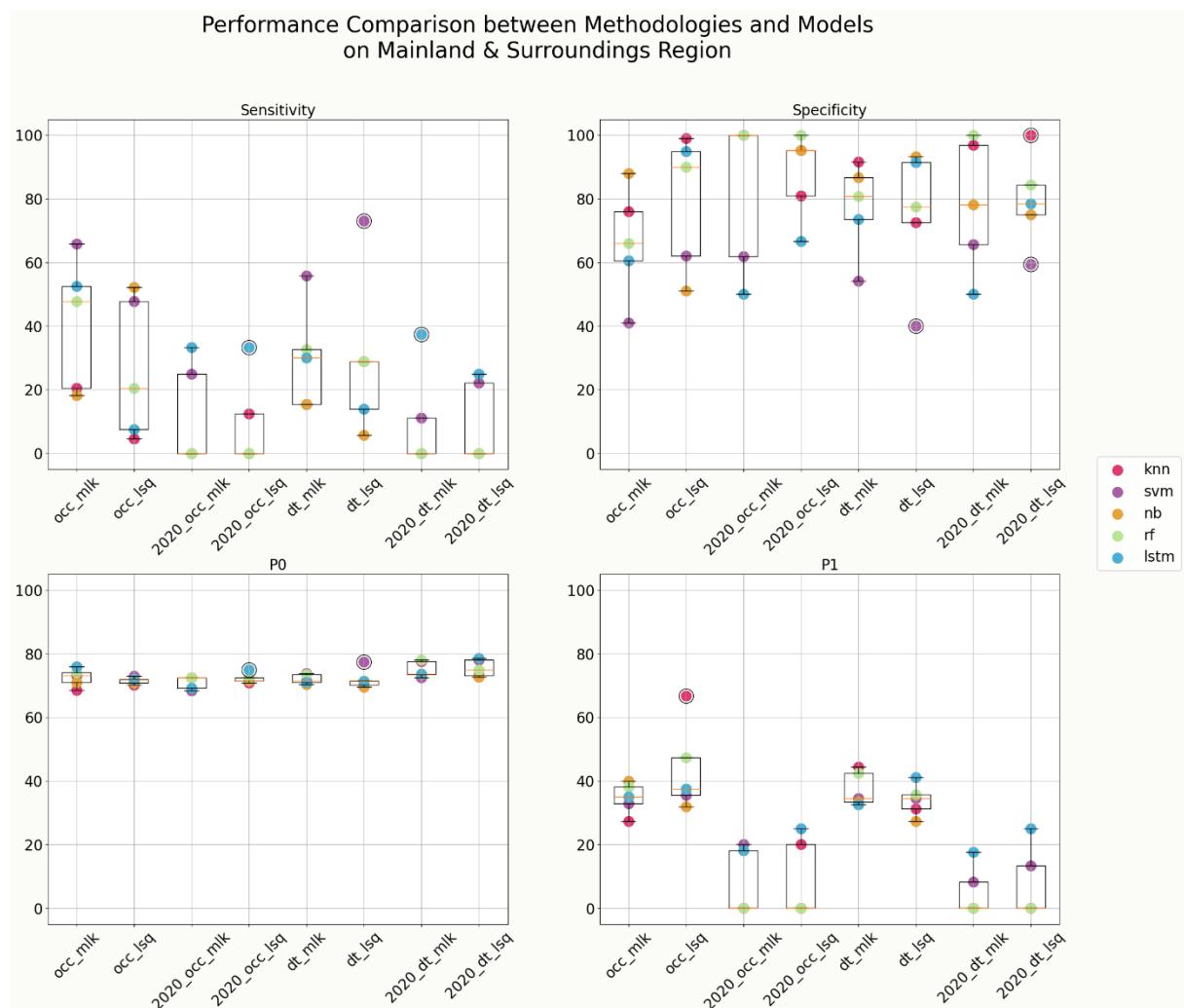


Figure 5.3 Figure 5.4 Boxplot of the pair wise results (P0 and P1, Sensitivity and Specificity) on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.

Between the applied methods, occurrence and time, there is some variability in results in terms of the pair wise metrics and $\beta-f_1$ (Figure 5.4 and Figure 5.5), with the average performance between them always close. On top, considering the statistical test on the $\beta-f_1$ metric the only significant divergence in performance is on the reduced datasets, it is then concluded that in the scenario of this study and area the only advantage noticed is the usage of a full catalogue with no time restriction.

Concerning the model's performance over the complete datasets, the SVM, RF and LSTM demonstrate steadier high $\beta-f_1$ results aligning with the statistical test (Table 5.1) where it is validated the similar performance among the models and divergence from the KNN and NB (Figure 5.4 and Table 5.2). The remaining models, oscillate with NB achieving comparable values on the occurrence method and KNN on the time method. Should be noted the LSTM is the only model able to achieve some correct predictions on all the reduced datasets, although lower than on the complete.

The comparable range of metrics to the prior study strongly validates the mainland area as a suitable location for further research.

Table 5.1 Post Hoc Neimenyi test performed relative to the dataset's performance on the $\beta-f_1$ metric. Test performed after a p-value of 0.02 being obtained on the Friedman Chi-Square test.

		Occurrence [2020,2024]		Time [2020, 2024]		Occurrence [2005, 2024]		Time [2005, 2024]					
		Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq				
Time [2005, 2024]	Occurrence [2020, 2024]	Occurrence [2020, 2024]		1.00	0.90	0.90	0.90	0.16	0.48	0.09	0.28		
		Mlk	Lsq	1.00	0.90	0.90	0.90	0.32	0.68	0.19	0.48		
Lsq	Mlk			1.00	0.90	0.90	0.90	0.06	0.25	0.03	0.12		
				1.00	0.90	0.90	0.90	0.22	0.56	0.12	0.36		
Mlk	Lsq			1.00	0.90	0.90	0.90	1.00	0.90	0.90	0.90		
				1.00	0.90	0.90	0.90	1.00	0.90	0.90	0.90		
Lsq	Mlk												

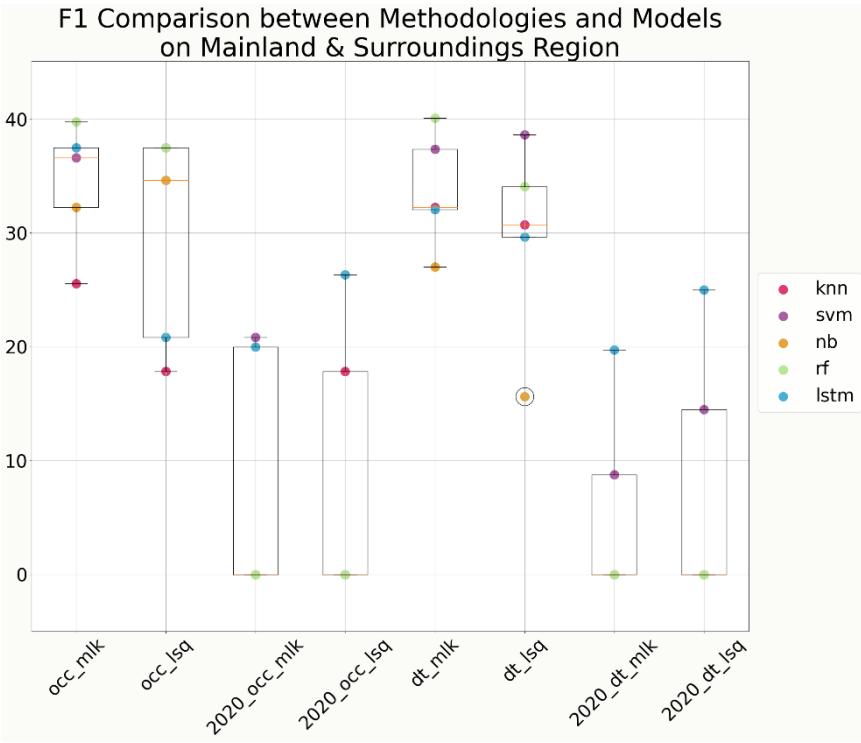


Figure 5.5 Boxplot of the β - f_1 metric on the Mainland and Surroundings cluster per dataset and model. With occ representing the occurrence method and dt the time. Lsq and Mlk refer to the estimation method implemented. Computed in Python with usage of the Matplotlib library.

Table 5.2 Post Hoc Neimenyi test performed relative to the models' performance on the β - f_1 metric. Test performed after a p-value of 0.00 being obtained on the Friedman Chi-Square test.

	KNN	SVM	NB	RF	LSTM
KNN	1.000	0.205	0.900	0.643	0.411
SVM		1.000	0.069	0.900	0.900
NB			1.000	0.364	0.175
RF				1.000	0.900
LSTM					1.000

6. CONCLUSION

The application of AI for earthquake prediction in Portugal has yielded mixed results. While advancements in specific algorithms have shown a performance increase when compared to the state of the art, regarding the Azores region, they haven't reached the level of quality initially anticipated. The introduction of Random Forest and LSTM demonstrated a noticeable ability of reducing false alarms, marking them as possible targets for deeper studies.

Furthermore, a crucial takeaway from this research is the lack of a single, universally successful methodology. No single method seems to outperform the others across the board, varying significantly with the algorithm choice. The only takeaway from the methodologies study being that the mainland region is not compatible with only using the most recent 4 years of data since it does not result in representative vectors for the area.

The findings regarding the first research question (*Do certain algorithms or methodologies exhibit superior performance in earthquake prediction when applied to data specific to Portugal?*), highlight the importance of continued exploration and development of diverse AI methods exploring the multiple methodologies in the Portugal region.

Addressing the second research question (*Considering the western Azores region has been subject to earthquake prediction studies; can those techniques also be applied to the mainland region of Portugal?*), interestingly, the mainland Portugal showcased results in the same range as the Azores, although as mentioned before, the results achieved are not considered of high-quality, as such it merely marks it as a potential area for further investigation alongside the Azores.

Summarizing, even though some algorithms showcased some improvement and the mainland proved to be a considerable target for future studies, the results on both regions support that a real-time application for earthquake prediction are likely not feasible in a near future, with further research needed both algorithm wise and datatype.

7. LIMITATION AND FUTURE WORK

One of the main limitations encountered during this research was data acquisition for Portugal. With the outcome of a research for open-source data resulting only in time-series data catalogues imposing a strong limitation to this study. From these results, two catalogs were explored, the first collected by the United States Geological Survey, USGS², which even though widely used on the literature review it proved to lack some quality on the Portugal region. The second and used catalogue available through EMSC showcased an improved quality, but still not the desirable high-quality that such studies demand. This highlights the need for Portuguese institutions to consider open-source catalog approaches to fuel this type of research.

Additionally, time constraints were encountered. Training multiple models with grid search across a wide range of datasets proved to be time-consuming with the available computational resources.

In a next step, a prediction window tuning would be tried to amplify the methodologies differences and understand how they affect the final predictions. Alongside, a deeper research into neural networks would be deployed, since it is a broad field showcasing promising results on other regions and on the Western Azores.

² <https://earthquake.usgs.gov/earthquakes/search/>

BIBLIOGRAPHICAL REFERENCES

- Alarifi, A. S. N., Alarifi, N. S. N., & Al-Humidan, S. (2012). Earthquakes magnitude predication using artificial neural network in northern Red Sea area. *Journal of King Saud University - Science*, 24(4), 301–313. <https://doi.org/10.1016/j.jksus.2011.05.002>
- Alexandridis, A., Chondrodima, E., Efthimiou, E., Papadakis, G., Vallianatos, F., & Triantis, D. (2014). Large earthquake occurrence estimation based on radial basis function neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 52(9), 5443–5453. <https://doi.org/10.1109/TGRS.2013.2288979>
- Alves, E. I. (2006). Earthquake forecasting using neural networks: Results and future work. *Nonlinear Dynamics*, 44(1–4), 341–349. <https://doi.org/10.1007/s11071-006-2018-1>
- Asencio-Cortés, G., Martínez-Álvarez, F., Morales-Esteban, A., & Reyes, J. (2016). A sensitivity study of seismicity indicators in supervised learning to improve earthquake prediction. *Knowledge-Based Systems*, 101, 15–30. <https://doi.org/10.1016/j.knosys.2016.02.014>
- Asencio-Cortés, G., Martínez-Álvarez, F., Troncoso, A., & Morales-Esteban, A. (2017). Medium-large earthquake magnitude prediction in Tokyo with artificial neural networks. *Neural Computing and Applications*, 28(5), 1043–1055. <https://doi.org/10.1007/s00521-015-2121-7>
- Asim, K. M., Idris, A., Iqbal, T., & Martínez-Álvarez, F. (2018). Seismic indicators based earthquake predictor system using Genetic Programming and AdaBoost classification. *Soil Dynamics and Earthquake Engineering*, 111, 1–7. <https://doi.org/10.1016/j.soildyn.2018.04.020>
- Asim, K. M., Martínez-Álvarez, F., Basit, A., & Iqbal, T. (2017). Earthquake magnitude prediction in Hindu Kush region using machine learning techniques. *Natural Hazards*, 85(1), 471–486. <https://doi.org/10.1007/s11069-016-2579-3>
- Banna, M. H. Al, Ghosh, T., Nahian, M. J. Al, Taher, K. A., Kaiser, M. S., Mahmud, M., Hossain, M. S., & Andersson, K. (2021). Attention-Based Bi-Directional Long-Short Term Memory Network for Earthquake Prediction. *IEEE Access*, 9, 56589–56603. <https://doi.org/10.1109/ACCESS.2021.3071400>
- Bbeiman, L. (1996). Bagging Predictors. *Machine Learning*, 24, 123–140.
- Berhich, A., Belouadha, F. Z., & Kabbaj, M. I. (2020, March 31). LSTM-based Models for Earthquake Prediction. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3386723.3387865>

- Berrar, D. (2018). Bayes' theorem and naive bayes classifier. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3, pp. 403–412). Elsevier. <https://doi.org/10.1016/B978-0-12-809633-8.20473-1>
- Bezzeghoud, M., Adam, C., Buforn, E., Borges, J. F., & Caldeira, B. (2014). Seismicity along the Azores-Gibraltar region and global plate kinematics. *Journal of Seismology*, 18(2), 205–220. <https://doi.org/10.1007/s10950-013-9416-x>
- Bhargava, B., & Pasari, S. (2022). Earthquake Prediction Using Deep Neural Networks. *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, 476–479. <https://doi.org/10.1109/ICACCS54159.2022.9785011>
- Borges, J. Â. F., Fitas, A. J. S., Bezzeghoud, M., & Teves-Costa, P. (n.d.). *Seismotectonics of Portugal and its adjacent Atlantic area*. www.elsevier.com/locate/tecto
- Botzen, W. J. W., Deschenes, O., & Sanders, M. (2019). The economic impacts of natural disasters: A review of models and empirical studies. In *Review of Environmental Economics and Policy* (Vol. 13, Issue 2, pp. 167–188). Oxford University Press. <https://doi.org/10.1093/reep/rez004>
- Breiman, L. (2001). *Random Forests*.
- Carvalho, A., & Malfeito, N. (n.d.). *MAPAS DE PERIGOSIDADE SÍSMICA PARA PORTUGAL CONTINENTAL: UMA ANÁLISE CRÍTICA Parte I – Períodos de recorrência de sismos*. www.lnec.pt
- Carvalho, A., & Malfeito, N. (2018). *MAPAS DE PERIGOSIDADE SÍSMICA PARA PORTUGAL CONTINENTAL: UMA ANÁLISE CRÍTICA Parte I – Períodos de recorrência de sismos*. www.lnec.pt
- Custódio, S., Dias, N. A., Carrilho, F., Góngora, E., Rio, I., Marreiros, C., Morais, I., Alves, P., & Matias, L. (2015). Earthquakes in western Iberia: Improving the understanding of lithospheric deformation in a slowly deforming region. *Geophysical Journal International*, 203(1), 127–145. <https://doi.org/10.1093/gji/ggv285>
- Daniell, J. E., Khazai, B., Wenzel, F., & Vervaet, A. (2012). *The worldwide economic impact of historic earthquakes*.
- Dasgupta, S., Bhattacharya, A., & Jana, K. (1998). Quantitative assessment of seismic hazard in eastern-northeastern India through Poisson probability density function analysis. *Journal of Geological Society of India*, 52, 181–194.
- Fontiela, J., Sousa Oliveira, C., & Rosset, P. (2018). Characterisation of seismicity of the azores archipelago: an overview of historical events and a detailed analysis for the period 2000–2012. In *Active Volcanoes of the World* (pp. 127–153). Springer. https://doi.org/10.1007/978-3-642-32226-6_8

- Geller, R. J. (1997). *SPECIAL SECTION-ASSESSMENT OF SCHEMES FOR EARTHQUAKE PREDICTION* *Earthquake prediction: a critical review* (Vol. 131). <https://academic.oup.com/gji/article/131/3/425/2138719>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gutenberg, B., & Richter, C. F. (1944). Frequency of Earthquakes in California. *Bulletin of the Seismological Society of America*, 34(4), 185–188.
- Last, M., Rabinowitz, N., & Leonard, G. (2016). Predicting the maximum earthquake magnitude from seismic data in Israel and its neighboring countries. *PLoS ONE*, 11(1). <https://doi.org/10.1371/journal.pone.0146101>
- Martínez-Álvarez, F., Reyes, J., Morales-Esteban, A., & Rubio-Escudero, C. (2013). Determining the best set of seismicity indicators to predict earthquakes. Two case studies: Chile and the Iberian Peninsula. *Knowledge-Based Systems*, 50, 198–210. <https://doi.org/10.1016/j.knosys.2013.06.011>
- Mavrouli, M., Mavroulis, S., Lekkas, E., & Tsakris, A. (2023). The Impact of Earthquakes on Public Health: A Narrative Review of Infectious Diseases in the Post-Disaster Period Aiming to Disaster Risk Reduction. In *Microorganisms* (Vol. 11, Issue 2). MDPI. <https://doi.org/10.3390/microorganisms11020419>
- Morales-Esteban, A., Martínez-Álvarez, F., & Reyes, J. (2013). Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence. *Tectonophysics*, 593, 121–134. <https://doi.org/10.1016/j.tecto.2013.02.036>
- Morales-Esteban, A., Martínez-Álvarez, F., Troncoso, A., Justo, J. L., & Rubio-Escudero, C. (2010). Pattern recognition to forecast seismic time series. *Expert Systems with Applications*, 37(12), 8333–8342. <https://doi.org/10.1016/j.eswa.2010.05.050>
- Moustra, M., Avraamides, M., & Christodoulou, C. (2011). Artificial neural networks for earthquake prediction using time series magnitude data or Seismic Electric Signals. *Expert Systems with Applications*, 38(12), 15032–15039. <https://doi.org/10.1016/j.eswa.2011.05.043>
- Narayananakumar, S., & Raja, K. (2016). A BP Artificial Neural Network Model for Earthquake Magnitude Prediction in Himalayas, India. *Circuits and Systems*, 07(11), 3456–3468. <https://doi.org/10.4236/cs.2016.711294>
- Niksarlioglu, S., & Kulahci. F. (2013). An Artificial Neural Network Model for Earthquake Prediction and Relations between Environmental Parameters and Earthquake. *World Academy of Science, Engineering and Technology International Journal of Geological and Environmental Engineering*, 7(2).

- Panakkat, A., & Adeli, H. (2007). NEURAL NETWORK MODELS FOR EARTHQUAKE MAGNITUDE PREDICTION USING MULTIPLE SEISMICITY INDICATORS. In *International Journal of Neural Systems* (Vol. 17, Issue 1). www.worldscientific.com
- Panakkat, A., & Adeli, H. (2009). Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Computer-Aided Civil and Infrastructure Engineering*, 24(4), 280–292. <https://doi.org/10.1111/j.1467-8667.2009.00595.x>
- Perol, T., Gharbi, M., & Denolle, M. (2018). *SEISMOLOGY Convolutional neural network for earthquake detection and location*. <http://advances.sciencemag.org/>
- Reyes, J., Morales-Esteban, A., & Martínez-Álvarez, F. (2013). Neural networks to predict earthquakes in Chile. *Applied Soft Computing Journal*, 13(2), 1314–1328. <https://doi.org/10.1016/j.asoc.2012.10.014>
- Rish, I. (2001). *An empirical study of the naive Bayes classifier*.
- Roy, S. (2014). Probabilistic Prediction for Earthquake in Bangladesh: Just How Big Does the Earthquake Have to Be Next Years? *Open Journal of Earthquake Research*, 03(02), 108–114. <https://doi.org/10.4236/ojer.2014.32011>
- Shi, Y., & Bolt, B. A. (1982). The standard error of the magnitude-frequency b value . *Bulletin of the Seismological Society of America*, 72(5), 1677–1687. <https://doi.org/10.1785/bssa0720051677>
- Sri Lakshmi, S., & Tiwari, R. K. (2009). Model dissection from earthquake time series: A comparative analysis using modern non-linear forecasting and artificial neural network approaches. *Computers and Geosciences*, 35(2), 191–204. <https://doi.org/10.1016/j.cageo.2007.11.011>
- Vanneschi, L., & Silva, S. (2023). *Lectures on Intelligent Systems*.
- Wang, Q., Guo, Y., Yu, L., & Li, P. (2020). Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 148–158. <https://doi.org/10.1109/TETC.2017.2699169>
- Zhang, H. (2004). *The Optimality of Naive Bayes*. www.aaai.org

APPENDIX A

A simple example of the Time Method with $\Delta t = 1$ day and $n= 1$, for simplicity purposes, is offered on the next figures. Alongside an explanation on the caption, it should be noted this scenario gets slightly more complex with the increase of n since in the study its value differ from Δt .

Iteration 1

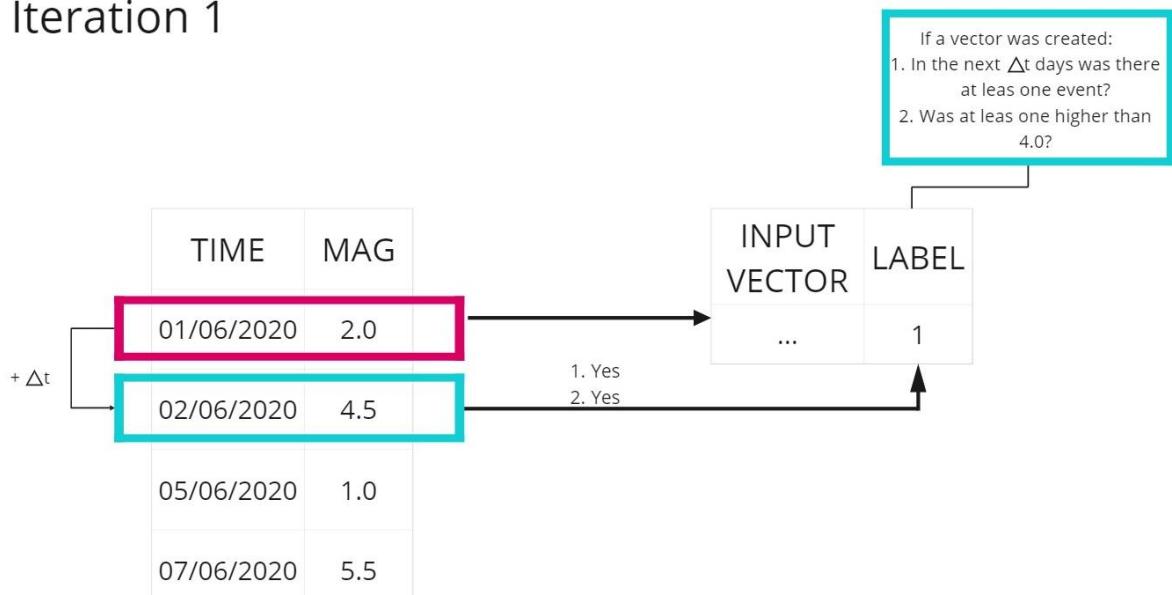


Figure A.1 Iteration 1 with a sliding window of size $\Delta t = 1$ using the time method. The pink rectangle corresponding to the current window under analysis and the blue the prediction window. In this step, a vector is created using the past n events – in this case the one in the current window (pink). The label is set to 1 since both conditions are met on the prediction window (blue).

Iteration 2

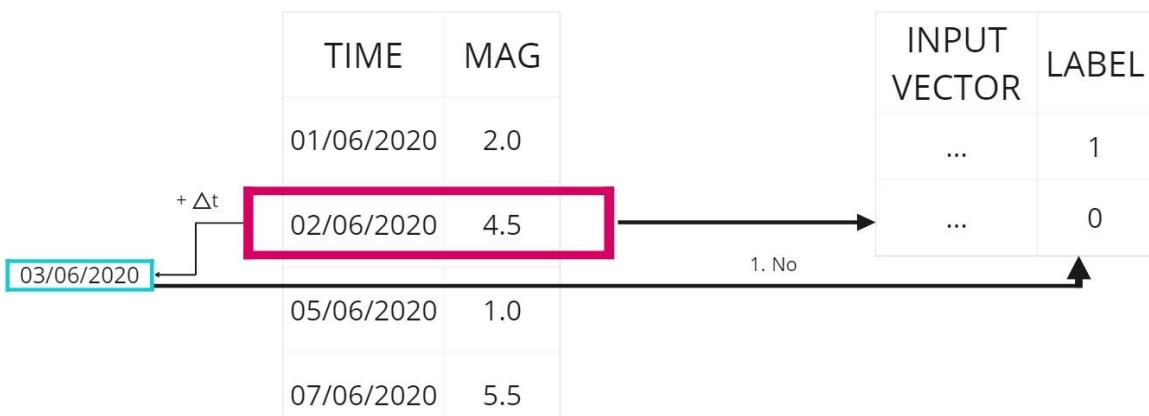


Figure A.2 The second iteration has an input vector computed with the information of the current window and a label of zero since one of the conditions is not met - the existence of an event in the prediction window.

Iteration 3

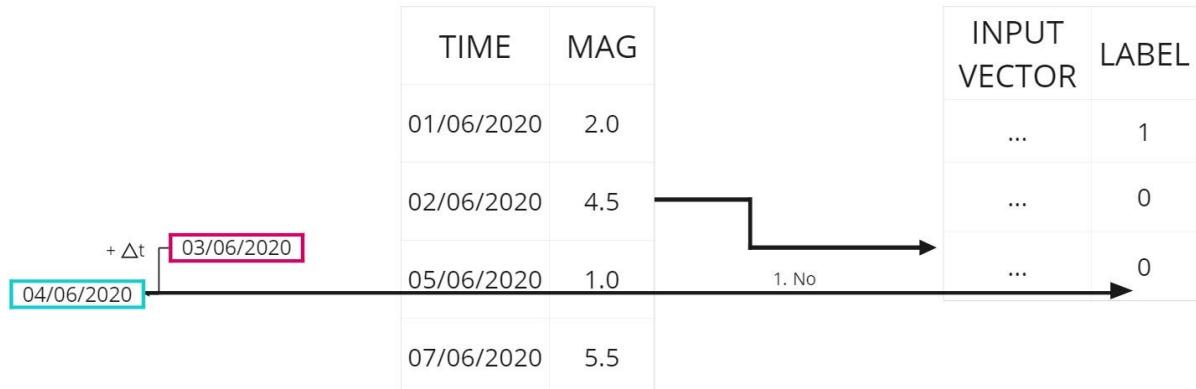


Figure A.3 Third iteration has an input vector equal to the previous one, since the n past events are the same (the one on the 02/06/2020). The label is set to zero since no events occurred in the prediction window.

Iteration 4



Figure A.4 Fourth iteration has an input vector equal to the last two previous ones, since the n past events are the same. The label is set to zero, even though events occurred in the prediction window the threshold magnitude is not reached.

Iteration 5



Figure A.5 Fifth iteration the input vector based on the last 1 event corresponds to the one on the 05/06/2020, the current window. The label is set to zero since no events occurred in the prediction window.

Iteration 6



Figure A.6 Sixth iteration has an input vector equal to the last one, since the n past events are the same. The label is set to one since at least one event in the prediction window is higher than 4.0.

To facilitate the understanding of how the different methods, affect the input vector, the same data presented previously is now subject to the Occurrence method also using $\Delta t = 1$ day and $n=1$, for simplicity and consistency purpose. The result and explanation can be found in the following pages.

Iteration 1

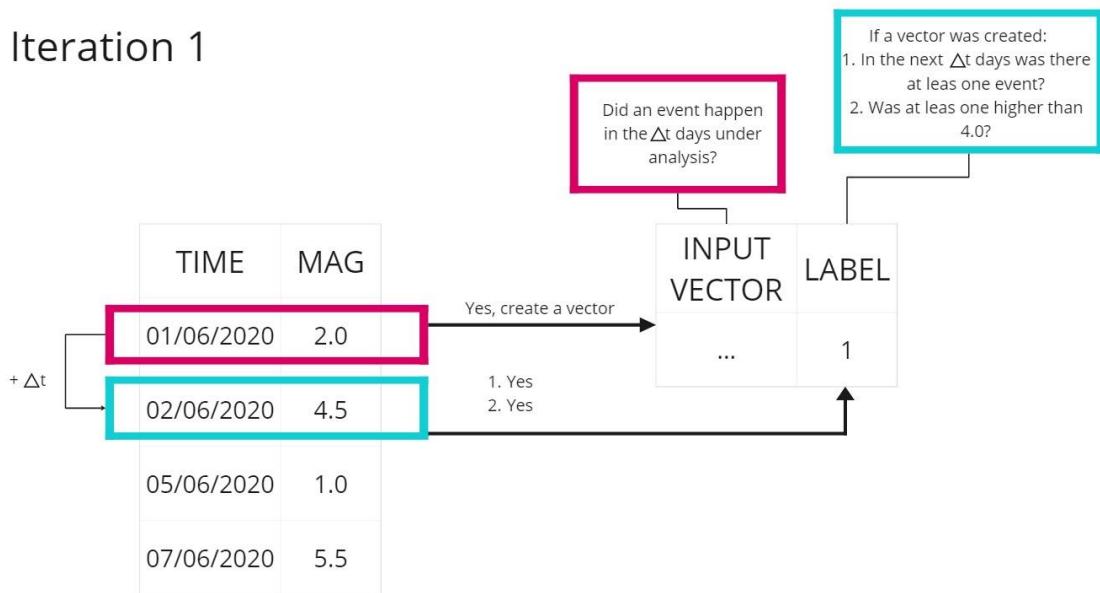


Figure A.7 The second iteration has an input vector computed with the information of the current window and a label of zero since one of the conditions is not met - the existence of an event in the prediction window.

Iteration 2

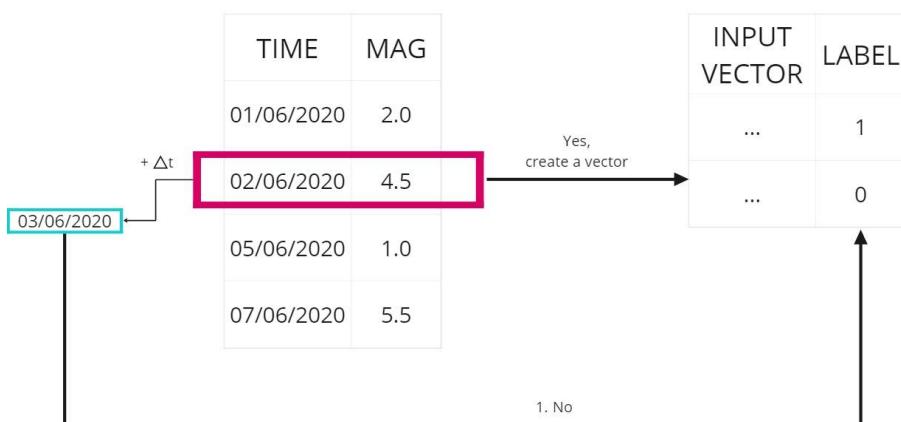


Figure A.8 Second iteration since there is at least one event in the current window an input vector is computed using the last n events ($n=1$). The label is set to zero since there is no event in the prediction window.

Iteration 3

The diagram illustrates the data for Iteration 3. On the left, a table shows a sequence of events over time:

TIME	MAG
01/06/2020	2.0
02/06/2020	4.5
05/06/2020	1.0
07/06/2020	5.5

An arrow points from the table to a second table on the right, labeled "INPUT VECTOR" and "LABEL". This second table has two rows:

INPUT VECTOR	LABEL
...	1
...	0

A red box highlights the date "03/06/2020" on the timeline. A green box highlights the date "04/06/2020". A bracket labeled "+ Δt" indicates the prediction window starting from "04/06/2020". A note states: "No, then no vector is created".

Figure A.9 Third iteration there is no event in the current window, as such no input vector creation. No input vector then no label.

Iteration 4

The diagram illustrates the data for Iteration 4. On the left, a table shows a sequence of events over time:

TIME	MAG
01/06/2020	2.0
02/06/2020	4.5
04/06/2020	
05/06/2020	1.0
07/06/2020	5.5

An arrow points from the table to a second table on the right, labeled "INPUT VECTOR" and "LABEL". This second table has two rows:

INPUT VECTOR	LABEL
...	1
...	0

A red box highlights the date "04/06/2020" on the timeline. A green box highlights the date "05/06/2020". A bracket labeled "+ Δt" indicates the prediction window starting from "04/06/2020". A note states: "No, then no vector is created".

Figure A.10 Fourth iteration since there is no event then no input vector calculation is triggered. Then even though there is an event in the prediction window the input vector is not created.

Iteration 5



Figure A.12 Fifth iteration input vector created given there is at least one event in the current window. Label set to zero due to no event present in the prediction window.

Iteration 6



Figure A.11 Sixth iteration does not trigger an input vector creation since there is no event in the current window as such no label is created either.

APPENDIX B

Grid Search Results on KNN

Table B.1 Grid Search results per Mainland cluster dataset on a search space of 1024 combinations on the KNN algorithm.

	Mainland [2005, 2024] Dataset				Mainland [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
Number of Neighbors, k	3	9	3	1	13	1	9	1
Similarity Metric	Chebyshev	Euclidean	Chebyshev	Chebyshev	Manhattan	Chebyshev	Chebyshev	Manhattan
Weights	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform
Algorithm	Ball Tree	Auto	Kd Tree	Ball Tree	Auto	Brute	Auto	Auto

Table B.2 Grid Search results per Azores cluster dataset on a search space of 1024 combinations on the KNN algorithm.

	Azores [2005, 2024] Dataset				Azores [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
Number of Neighbors, k	7	1	5	1	1	1	1	1
Similarity Metric	Chebyshev	Chebyshev	Chebyshev	Chebyshev	Chebyshev	Euclidean	Chebyshev	Euclidean
Weights	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform
Algorithm	Auto	Auto	Brute	Brute	Auto	Auto	Auto	Auto

Grid Search Results on NB

Table B.3 Grid Search results per Mainland cluster dataset on a search space of 400 combinations on the Gaussian NB algorithm.

	Mainland [2005, 2024] Dataset				Mainland [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
Var smoothing	0.100	0.100	0.081	0.010	0.035	0.035	0.035	0.100

Table B.4 Grid Search results per Azores cluster dataset on a search space of 400 combinations on the Gaussian NB algorithm.

	Azores [2005, 2024] Dataset				Azores [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq
Var smoothing	0.066	0.100	0.081	0.043	0.100	0.081	0.100	0.100

Grid Search Results on SVM

Table B.5 Grid Search results per Mainland cluster dataset on a search space of 320 combinations on the SVM algorithm.

	Mainland [2005, 2024] Dataset				Mainland [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq
Regulation term, C	3.5	3.5	3.5	4.5	0.5	1.0	0.5	5.0
Kernel	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Poly	Rbf	Sigmoid	Sigmoid
Gamma	Auto	Auto	Auto	Auto	Auto	Auto	Auto	Auto

Table B.6 Grid Search results per Azores cluster dataset on a search space of 320 combinations on the SVM algorithm.

	Azores [2005, 2024] Dataset				Azores [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq
Regulation term, C	4	2	4.5	5	1	4.5	3	0.5
Kernel	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Linear	Sigmoid	Linear	Linear
Gamma	Auto	Auto	Auto	Scale	Auto	Auto	Auto	Auto

Grid Search Results on RF

Table B.7 Grid Search results per Mainland cluster dataset on a search space of 720 combinations on the RF algorithm.

	Mainland [2005, 2024] Dataset				Mainland [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
Number of Estimators	90	250	150	250	90	150	100	200
Maximum Features	Log2	Sqrt	Log2	Sqrt	Log2	Log2	Log2	Log2
Maximum Depth	2	6	2	2	2	2	5	2
Criterion	Log Loss	Entropy	Entropy	Gini	Log Loss	Entropy	Gini	Log Loss

Table B.8 Grid Search results per Azores cluster dataset on a search space of 720 combinations on the RF algorithm.

	Azores [2005, 2024] Dataset				Azores [2020, 2024] Dataset			
	Occurrence		Time		Occurrence		Time	
	mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
Number of Estimators	250	90	90	250	250	90	90	200
Maximum Features	Log2	Sqrt	Log2	Sqrt	Sqrt	Log2	Log2	Sqrt
Maximum Depth	2	2	2	2	2	2	2	2
Criterion	Entropy	Gini	Gini	Entropy	Gini	Entropy	Entropy	Gini

Grid Search Results on LSTM

Table B.9 Grid Search results per Mainland cluster dataset on a search space of 288 combinations on the LSTM algorithm.

		Mainland [2005, 2024] Dataset				Mainland [2020, 2024] Dataset			
		Occurrence		Time		Occurrence		Time	
		mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
First Layer	Number of Neurons	200	200	200	250	250	150	200	200
	L2 Regularization	0.4	0.4	0.3	0.4	0.4	0.4	0.4	0.3
Second Layer	Number of Neurons	150	150	150	150	150	100	100	0
Third Layer	Drop Out Rate	0.2	0.4	0.2	0.2	0.4	0.2	0.4	0.2
General	Batch Size	20	10	20	10	10	10	20	10
	Learning Rate	0.001	0.001	0.0001	0.001	0.001	0.001	0.001	0.0001
	Optimizer	Adadelta	Adadelta	Adadelta	Adadelta	Adadelta	Adadelta	Adadelta	Adadelta

Table B.10 Grid Search results per Azores cluster dataset on a search space of 288 combinations on the LSTM algorithm

		Azores [2005, 2024] Dataset				Azores [2020, 2024] Dataset			
		Occurrence		Time		Occurrence		Time	
		mlk	Lsq	mlk	Lsq	mlk	Lsq	mlk	Lsq
First Layer	Number of Neurons	200	250	250	250	150	250	200	200
	L2 Regularization	0.4	0.4	0.3	0.3	0.3	0.3	0.3	0.4
Second Layer	Number of Neurons	0	0	100	100	150	100	150	0
Third Layer	Drop Out Rate	0.2	0.2	0.2	0.2	0.4	0.4	0.2	0.4
General	Batch Size	20	20	20	10	10	20	20	10
	Learning Rate	0.001	0.001	0.0001	0.0001	0.0001	0.0001	0.001	0.0001
	Optimizer	Adadelta	Adadelta	Adadelta	Adadelta	Adadelta	Adam	Adadelta	Adadelta

APPENDIX C

In this section the features selected per model and dataset are presented, the specific θ values selected by each set should be checked in the given notebooks. Since only the machine learning model are subjected of a feature selection part only them where included.

Table C.1 Features selected per model and dataset on the Azores cluster for the KNN and NB algorithms.

		Year	Month	Week	T	M_{mean}	$dE^{1/2}$	b value	η	ΔM	μ_θ	c_θ	x_1	x_2	x_3	x_4	x_5	x_6
KNN	Occurrence [2020,2024]	mlk	lsq				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Time [2020,2024]	mlk	lsq					✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2005,2024]	mlk	lsq				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Time [2005,2024]	mlk	lsq					✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2020,2024]	mlk	lsq				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Time [2020,2024]	mlk	lsq					✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2005,2024]	mlk	lsq				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Time [2005,2024]	mlk	lsq					✓		✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2020,2024]	mlk	lsq				✓	✓		✓				✓	✓	✓	✓	
	Time [2020,2024]	mlk	lsq							✓	✓	✓	✓	✓	✓	✓	✓	
NB	Occurrence [2020,2024]	mlk	lsq				✓	✓		✓				✓	✓	✓	✓	
	Time [2020,2024]	mlk	lsq							✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2005,2024]	mlk	lsq							✓	✓	✓	✓	✓	✓	✓	✓	
	Time [2005,2024]	mlk	lsq							✓	✓	✓	✓	✓	✓	✓	✓	
	Occurrence [2020,2024]	mlk	lsq							✓	✓	✓	✓	✓	✓	✓	✓	

Table C.2 Features selected per model and dataset on the Azores cluster for the SVM and RF algorithms.

		Year	Month	Week	T	M_{mean}	$dE^{1/2}$	b value	η	ΔM	μ_θ	c_θ	x_1	x_2	x_3	x_4	x_5	x_6		
						Occurrence [2020,2024]														
SVM																				
RF	Occurrence [2005,2024]	Time [2020,2024]	Occurrence [2020,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		Time [2005,2024]	Occurrence [2005,2024]	mlk	lsq					✓	✓	✓	✓	✓	✓	✓	✓	✓		
		mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		

Table C.3 Features selected per model and dataset on the Mainland and Surroundings cluster for the KNN and NB algorithms.

		Year	Month	Week	T	M_{mean}	$dE^{1/2}$	b value	η	ΔM	μ_θ	c_θ	x_1	x_2	x_3	x_4	x_5	x_6	
		Occurrence [2020,2024]						Occurrence [2020,2024]						Occurrence [2020,2024]					
		Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq		
KNN	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
NB	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	Time [2005,2024]	Occurrence [2020,2024]	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
	mlk	lsq	Occurrence [2020,2024]						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		

Table C.4 Features selected per model and dataset on the Mainland and Surroundings cluster for the SVM and RF algorithms.

		Year		Month		Week		T		M_{mean}		$dE^{1/2}$		b value		η		ΔM		μ_θ		c_θ		x_1	x_2	x_3	x_4	x_5	x_6	
		Occurrence [2005,2024]	Time [2005,2024]	Occurrence [2020,2024]	Time [2020,2024]	Occurrence [2005,2024]	Time [2005,2024]	Occurrence [2020,2024]	Time [2020,2024]	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	x_1	x_2	x_3	x_4	x_5
SVM	RF	Time [2005,2024]	Occurrence [2005,2024]	Time [2020,2024]	Occurrence [2020,2024]	Time [2005,2024]	Occurrence [2005,2024]	Time [2020,2024]	Occurrence [2020,2024]	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		mlk	lsq	mlk	lsq	mlk	lsq	mlk	lsq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

APPENDIX D

Azores Results

KNN Algorithm

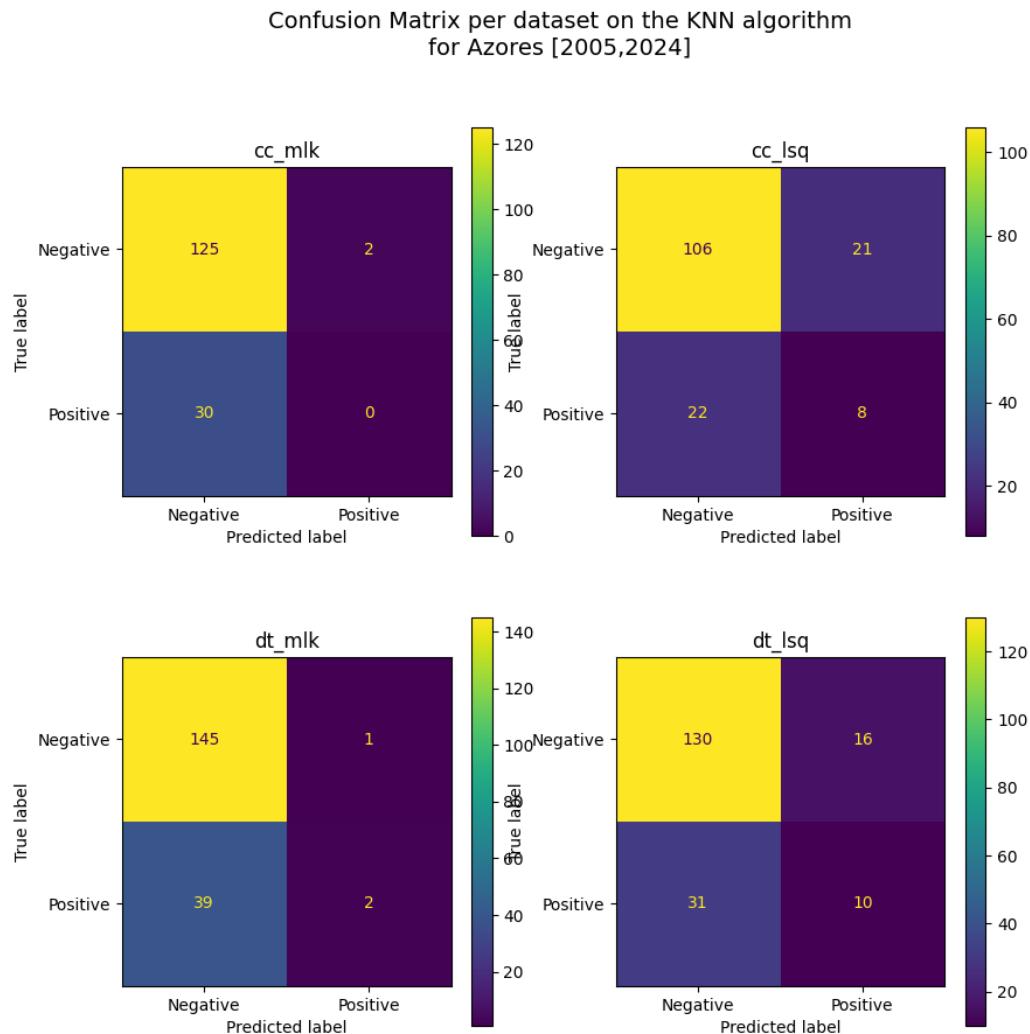


Figure D.1 Confusion Matrix results for the Azores [2005, 2024] cluster on the KNN algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

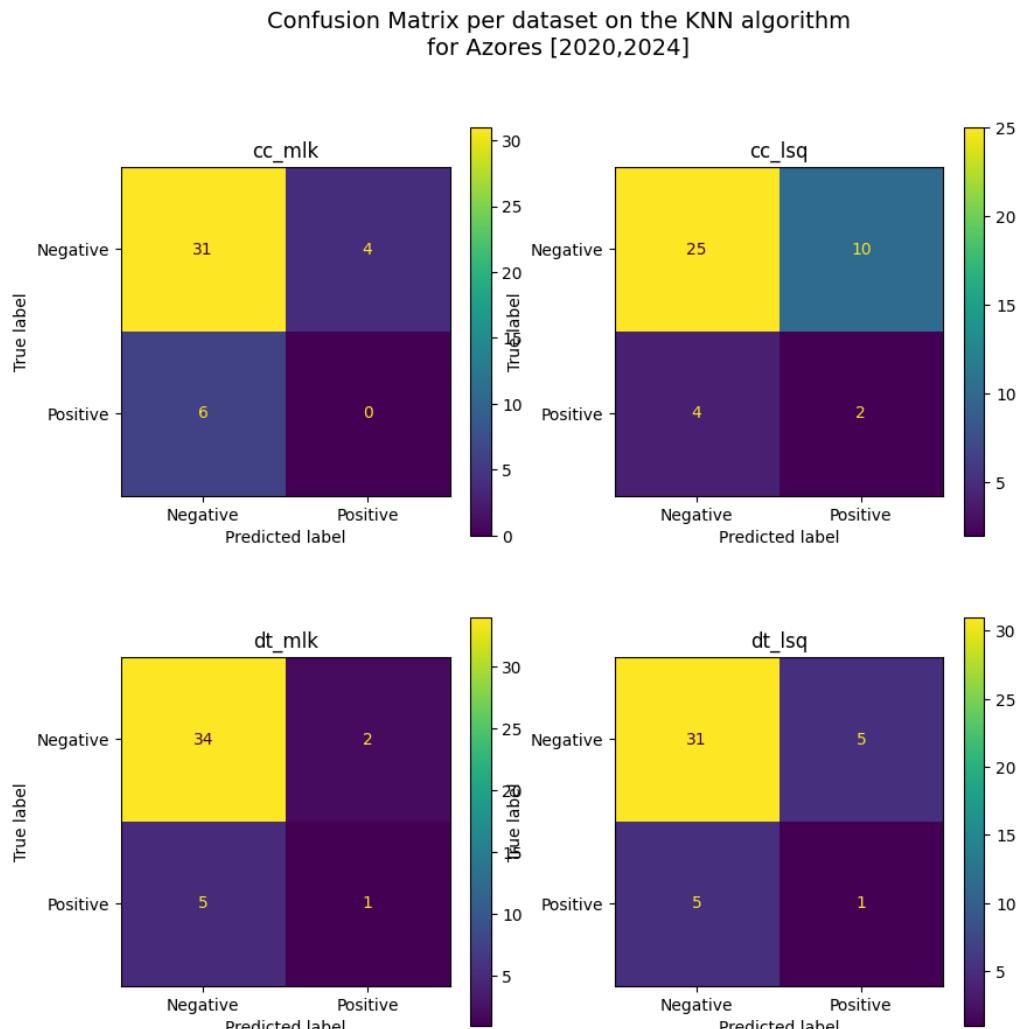


Figure D.2 Confusion Matrix results for the Azores [2020, 2024] cluster on the KNN algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.1 Metrics values for the Azores cluster on the KNN algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	0.00	26.67	4.88	24.39	0.00	33.33	16.67	16.67
Specificity	98.43	83.46	99.32	89.04	88.57	71.43	94.44	86.11
P0	80.65	8281	78.80	80.75	83.78	86.21	87.18	86.11
P1	0.00	27.59	66.67	38.46	0.00	16.67	33.33	16.67
F1	0.00	27.40	18.87	34.48	0.00	18.52	27.78	16.67

SVM Algorithm

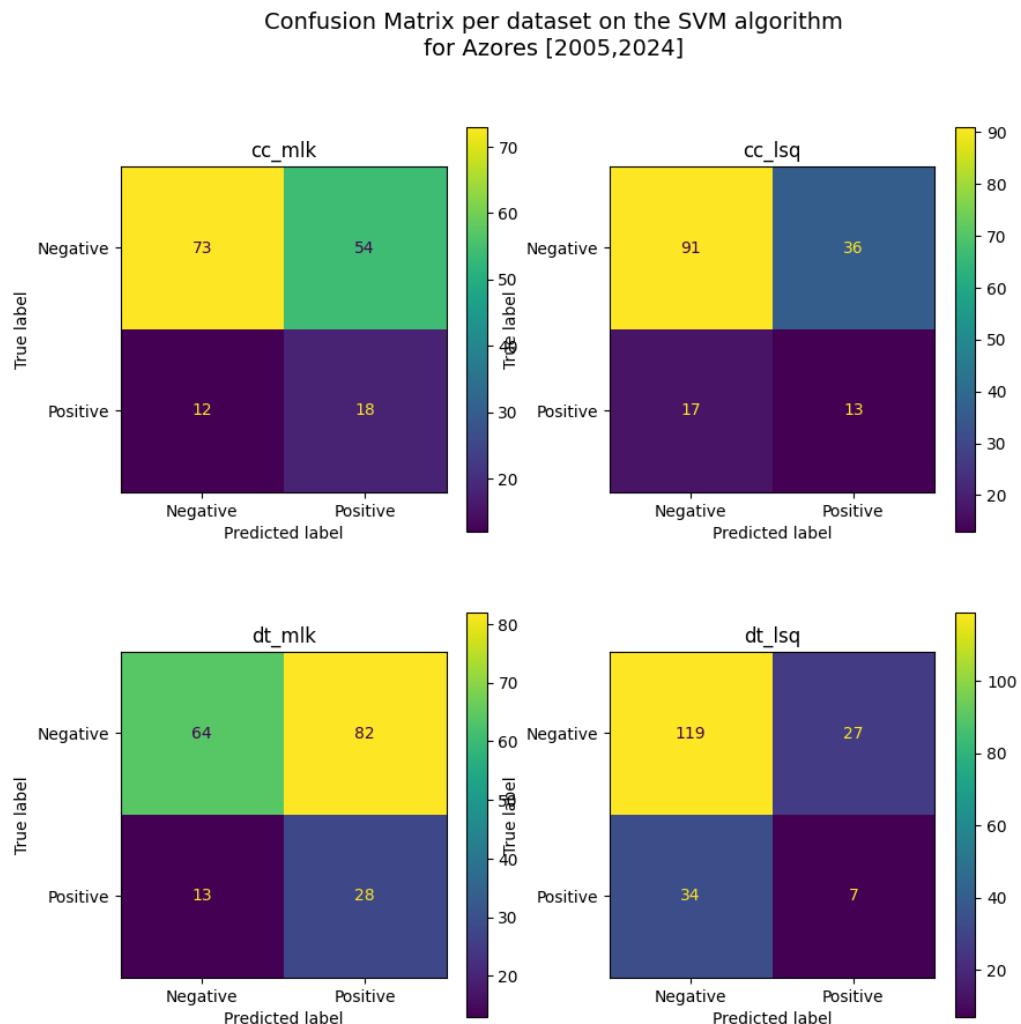


Figure D.3 Confusion Matrix results for the Azores [2005, 2024] cluster on the SVM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

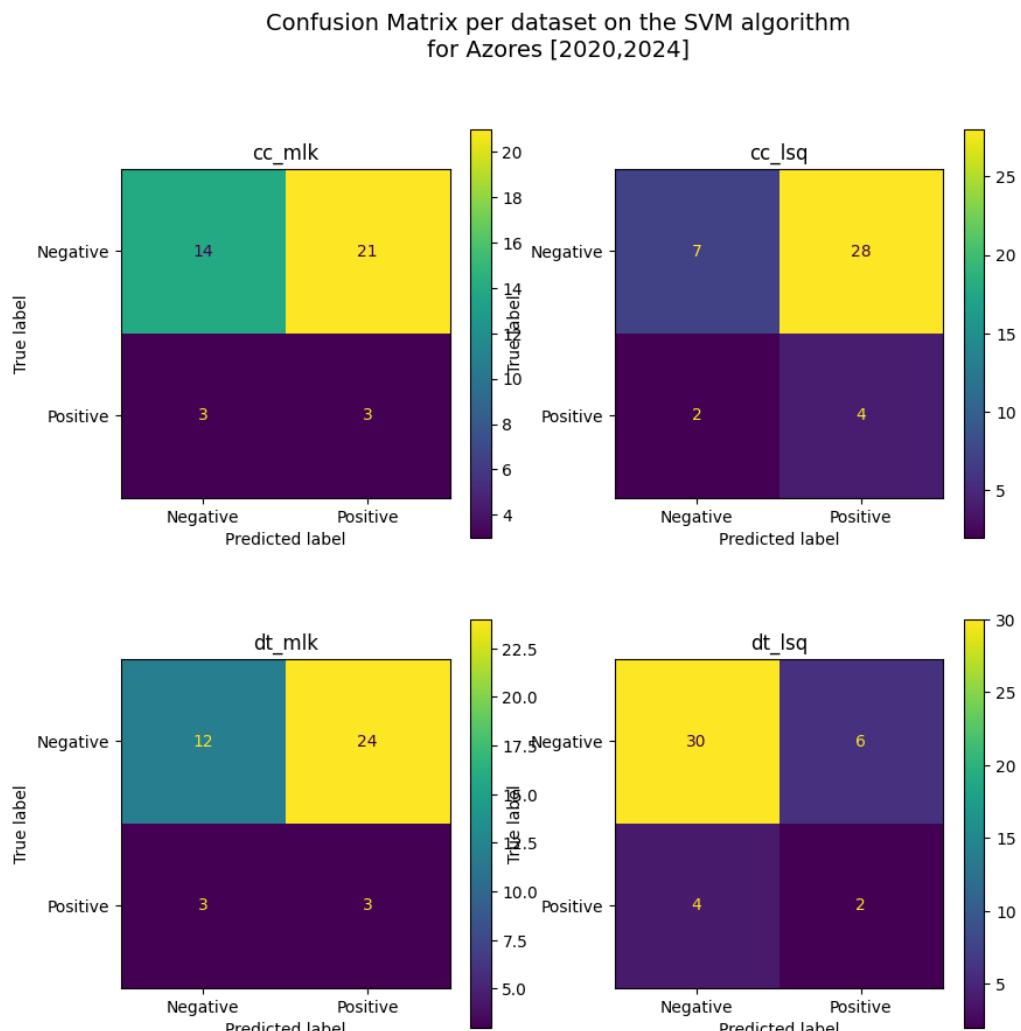


Figure D.4 Confusion Matrix results for the Azores [2020, 2024] cluster on the SVM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.2 Metrics values for the Azores cluster on the SVM algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	60.00	43.33	68.29	17.07	50.00	66.67	50.00	33.33
Specificity	57.48	71.65	43.84	81.51	40.00	20.00	33.33	83.33
P0	85.88	84.26	83.12	77.78	82.35	77.78	80.00	88.24
P1	25.00	26.53	25.45	20.59	12.50	12.50	11.11	25.00
F1	28.30	28.76	29.11	19.77	14.71	14.93	13.16	26.32

NB Algorithm

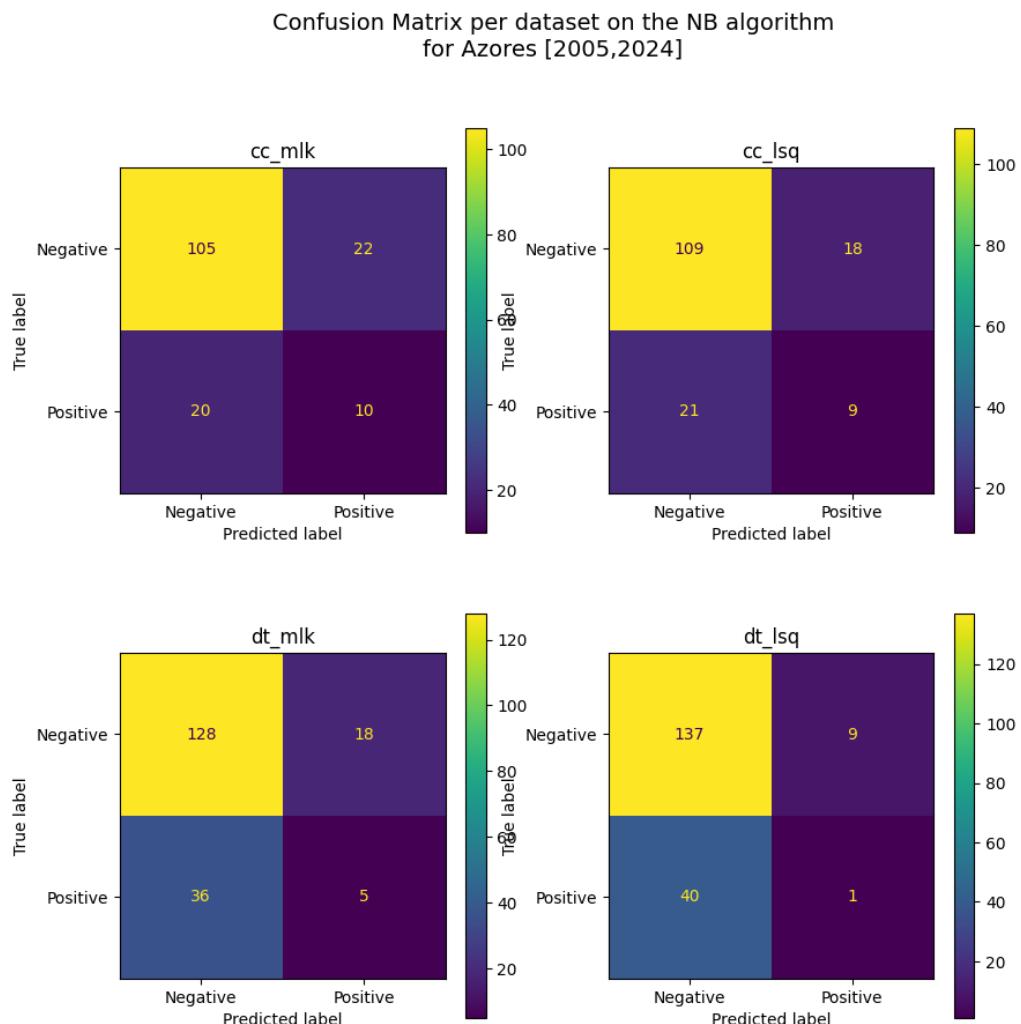


Figure D.5 Confusion Matrix results for the Azores [2005, 2024] cluster on the NB algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

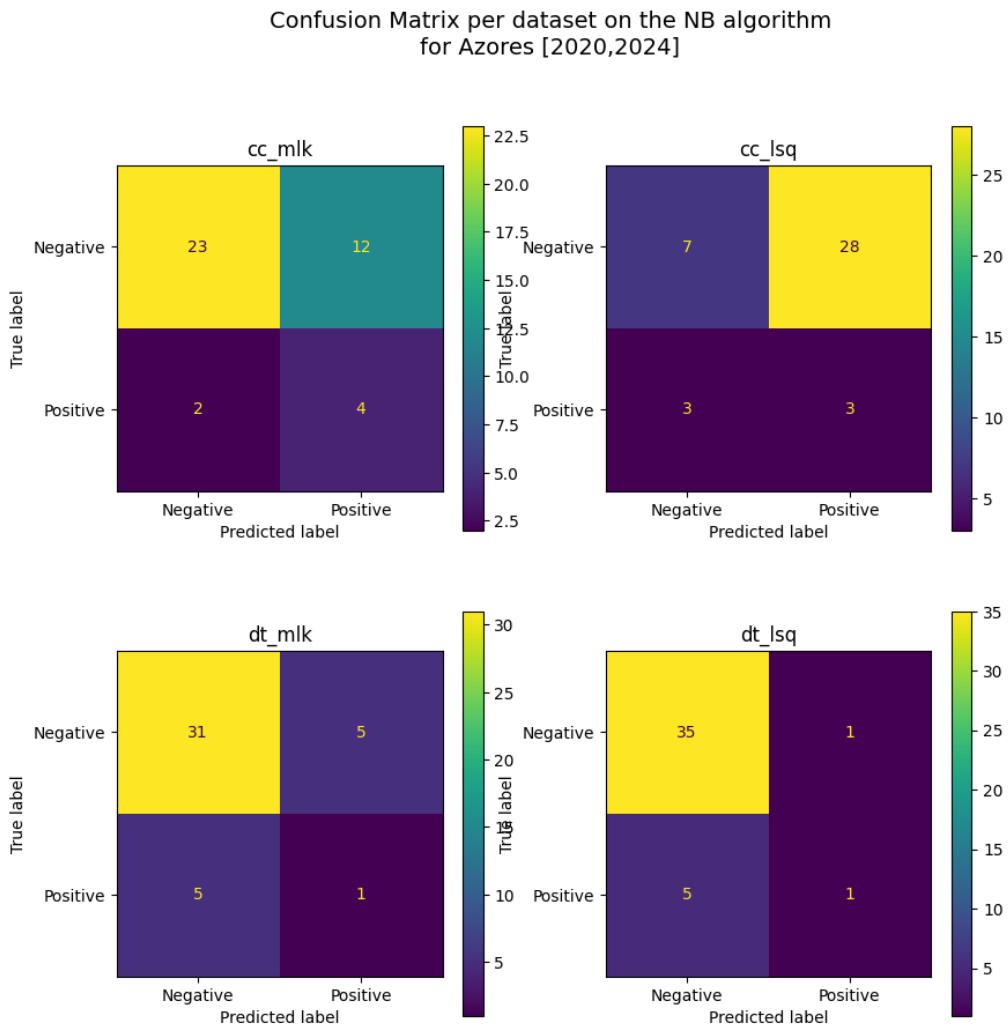


Figure D.6 Confusion Matrix results for the Azores [2020, 2024] cluster on the NB algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.3 Metrics values for the Azores cluster on the NB algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	33.33	30.00	12.20	2.44	66.67	50.00	16.67	16.67
Specificity	82.68	85.83	87.67	93.84	65.71	20.00	86.11	97.22
P0	84.00	83.85	78.05	77.40	92.00	70.00	86.11	87.50
P1	31.25	33.33	21.74	10.00	25.00	9.68	16.67	50.00
F1	31.65	32.61	18.80	6.17	28.57	11.54	16.67	35.71

RF Algorithm

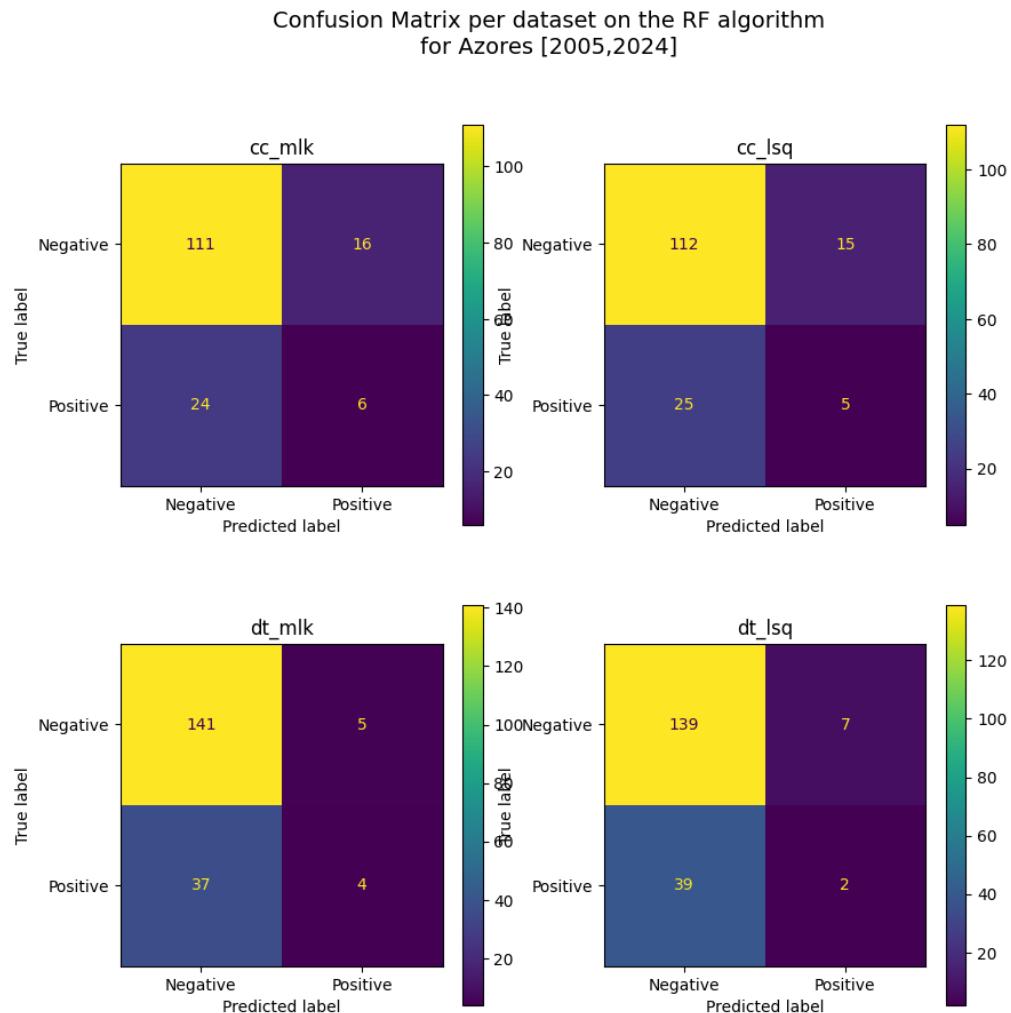


Figure D.7 Confusion Matrix results for the Azores [2005, 2024] cluster on the RF algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

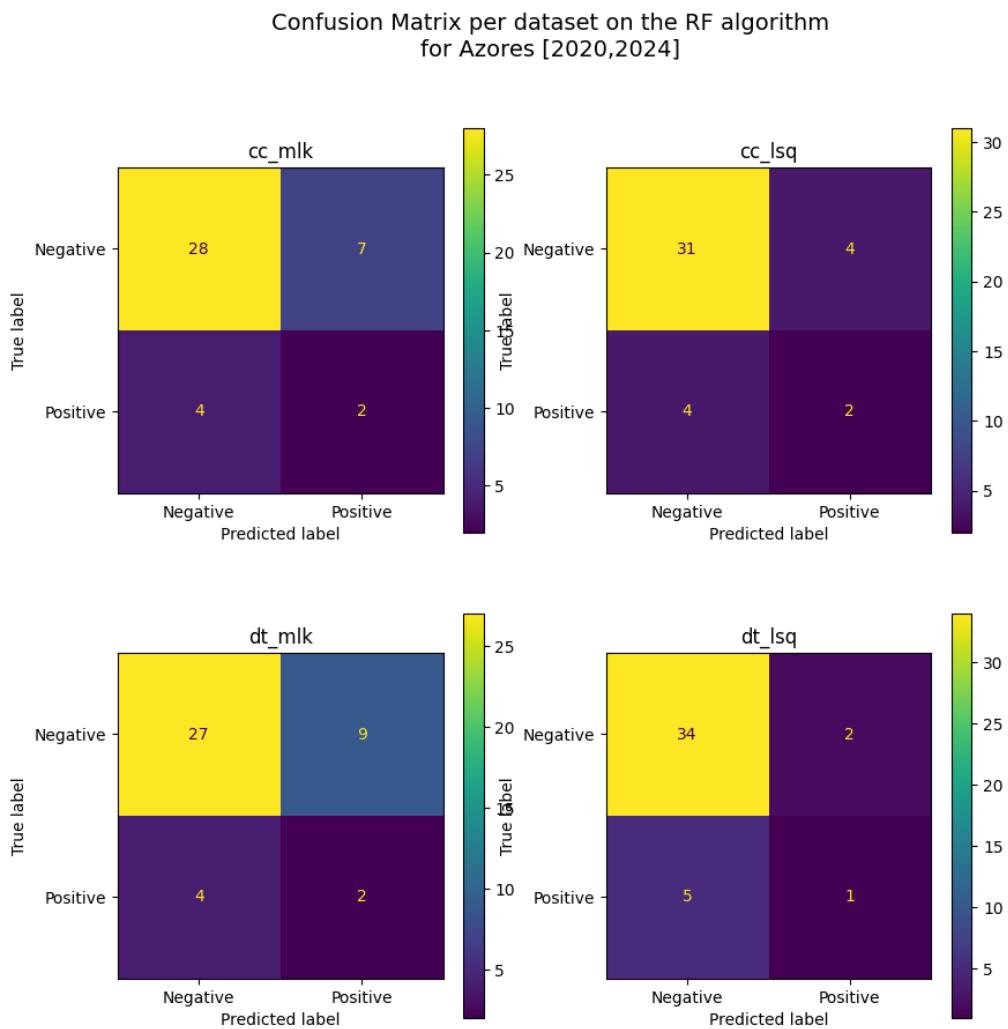


Figure D.8 Confusion Matrix results for the Azores [2020, 2024] cluster on the RF algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.4 Metrics values for the Azores cluster on the RF algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	20.00	16.67	9.76	4.88	33.33	33.33	33.33	16.67
Specificity	87.40	88.19	96.58	95.21	80.00	88.57	75.00	94.44
P0	82.22	81.75	79.21	78.09	87.50	88.57	87.10	87.18
P1	27.27	25.00	44.44	22.22	22.22	33.33	18.18	33.33
F1	25.42	22.73	25.97	12.99	23.81	33.33	20.00	27.78

LSTM Algorithm

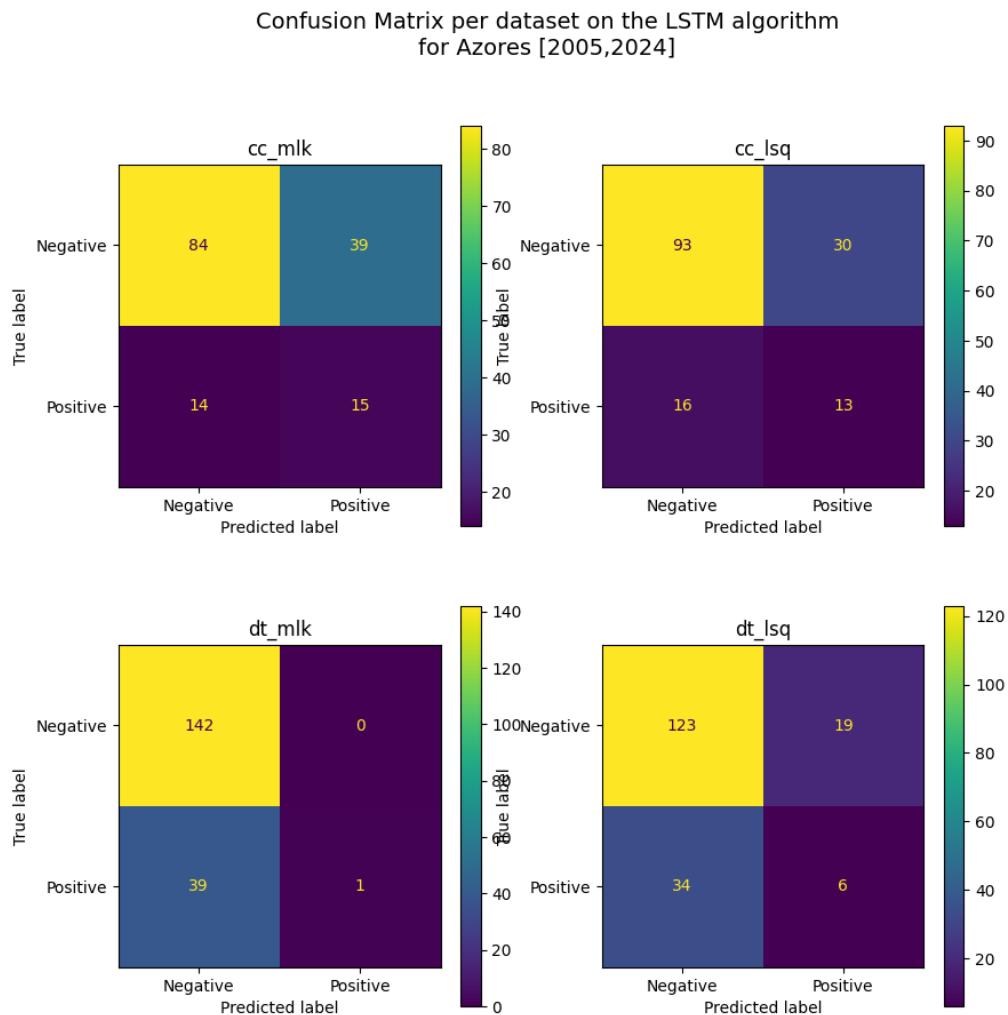


Figure D.9 Confusion Matrix results for the Azores [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Confusion Matrix per dataset on the LSTM algorithm
for Azores [2020,2024]

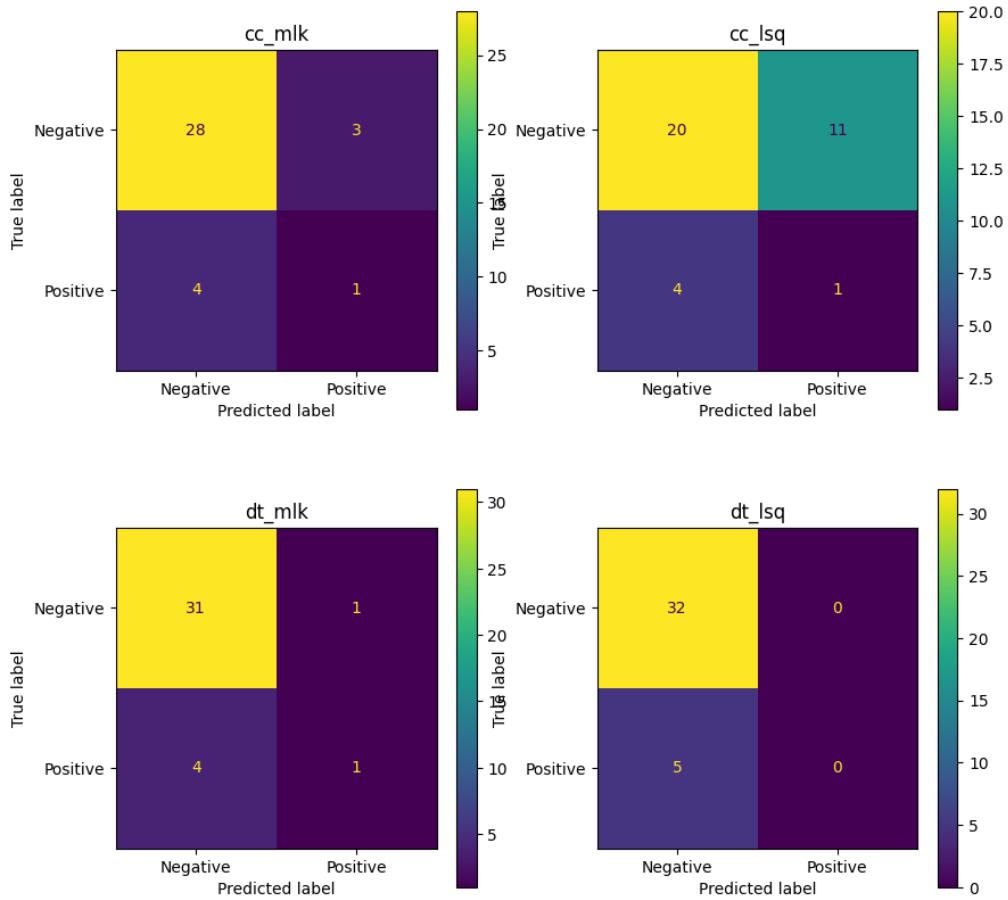


Figure D.10 Confusion Matrix results for the Azores [2020, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.5 Metrics values for the Azores cluster on the LSTM algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	51.72	44.83	2.50	15.00	20.00	20.00	20.00	0.00
Specificity	68.29	75.61	100.00	86.68	90.32	64.52	96.88	100.00
P0	85.71	85.32	78.45	78.34	87.50	83.33	88.57	86.49
P1	27.78	30.23	100.00	24.00	25.00	8.33	50.00	0.00
F1	30.61	32.34	11.36	21.43	23.81	9.43	38.46	0.00

Mainland and Surrounding Results

KNN Algorithm

Confusion Matrix per dataset on the KNN algorithm
for Mainland [2005,2024]

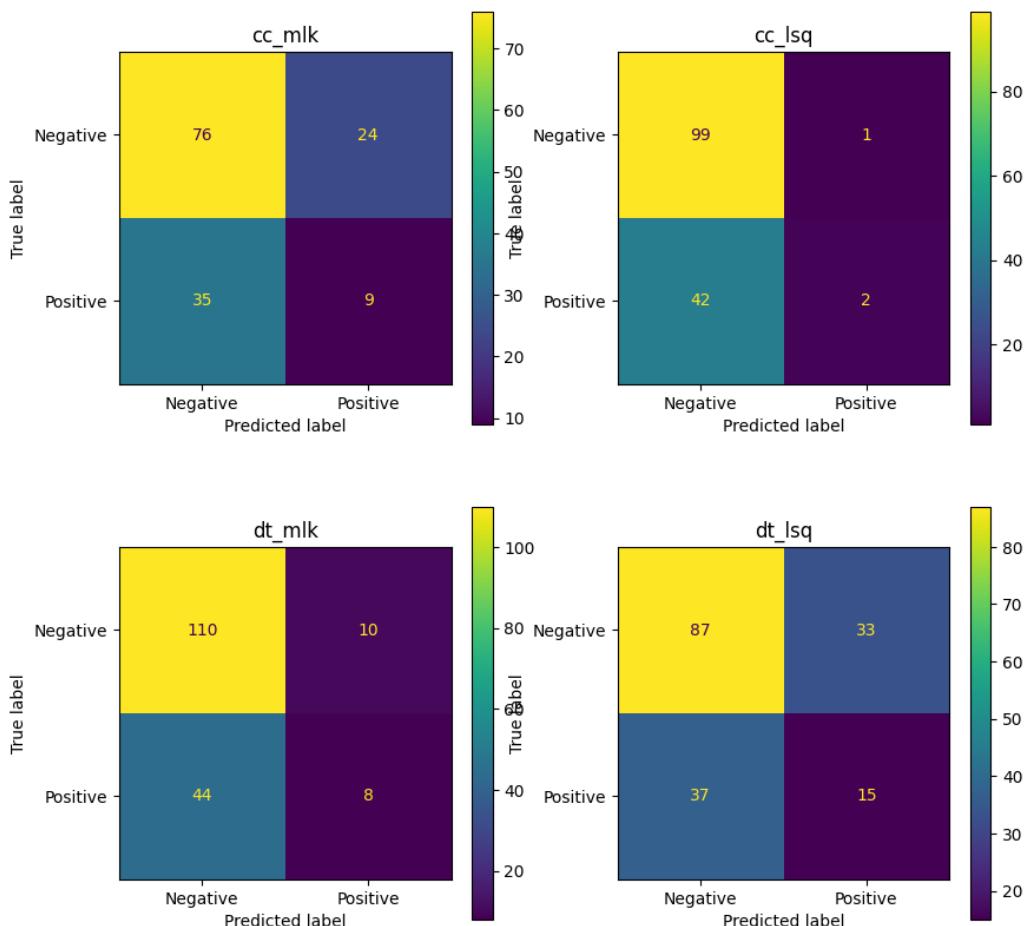


Figure D. 11 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the KNN algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Confusion Matrix per dataset on the KNN algorithm
for Mainland [2020,2024]

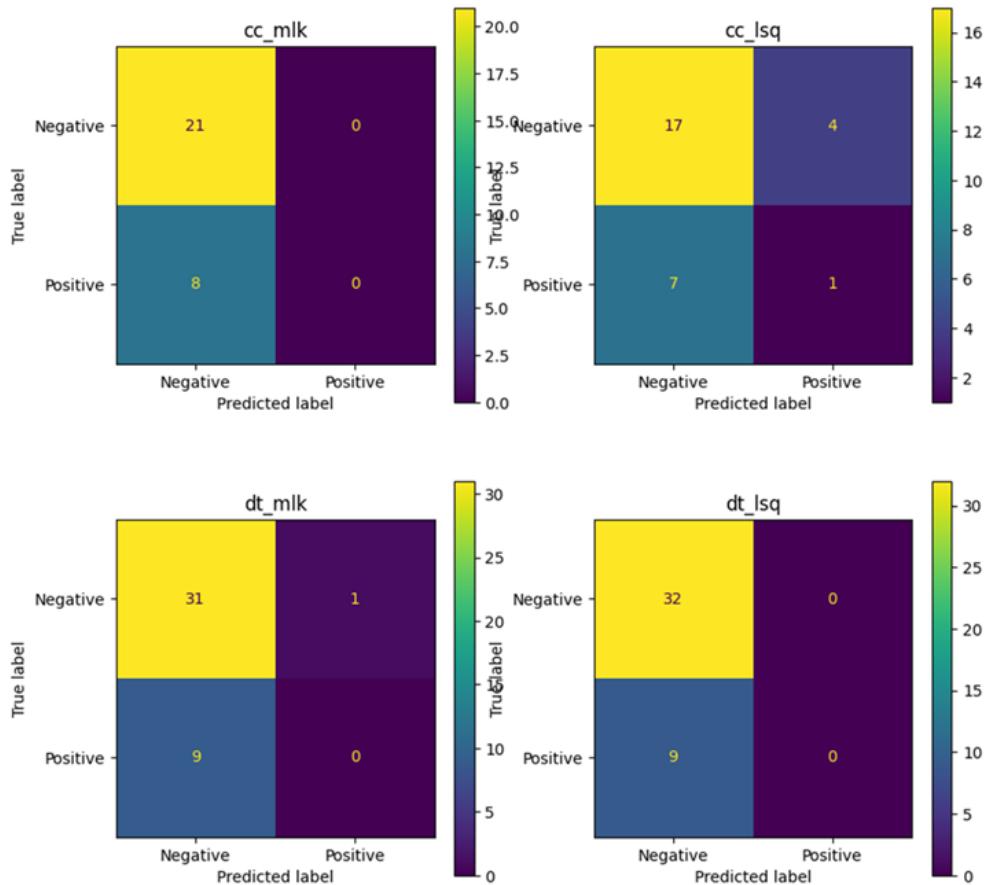


Figure D.12 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the KNN algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.6 Metrics values for the Mainland and Surroundings cluster on the KNN algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	20.45	4.55	15.38	28.85	0.00	12.50	0.00	0.00
Specificity	76.00	99.00	91.67	72.50	100.00	80.95	96.88	100.00
P0	68.47	70.21	71.43	70.16	72.41	70.83	77.50	78.05
P1	27.27	66.67	44.44	31.25	0.00	20.00	0.00	0.00
F1	25.57	17.86	32.26	30.74	0.00	17.86	0.00	0.00

SVM Algorithm

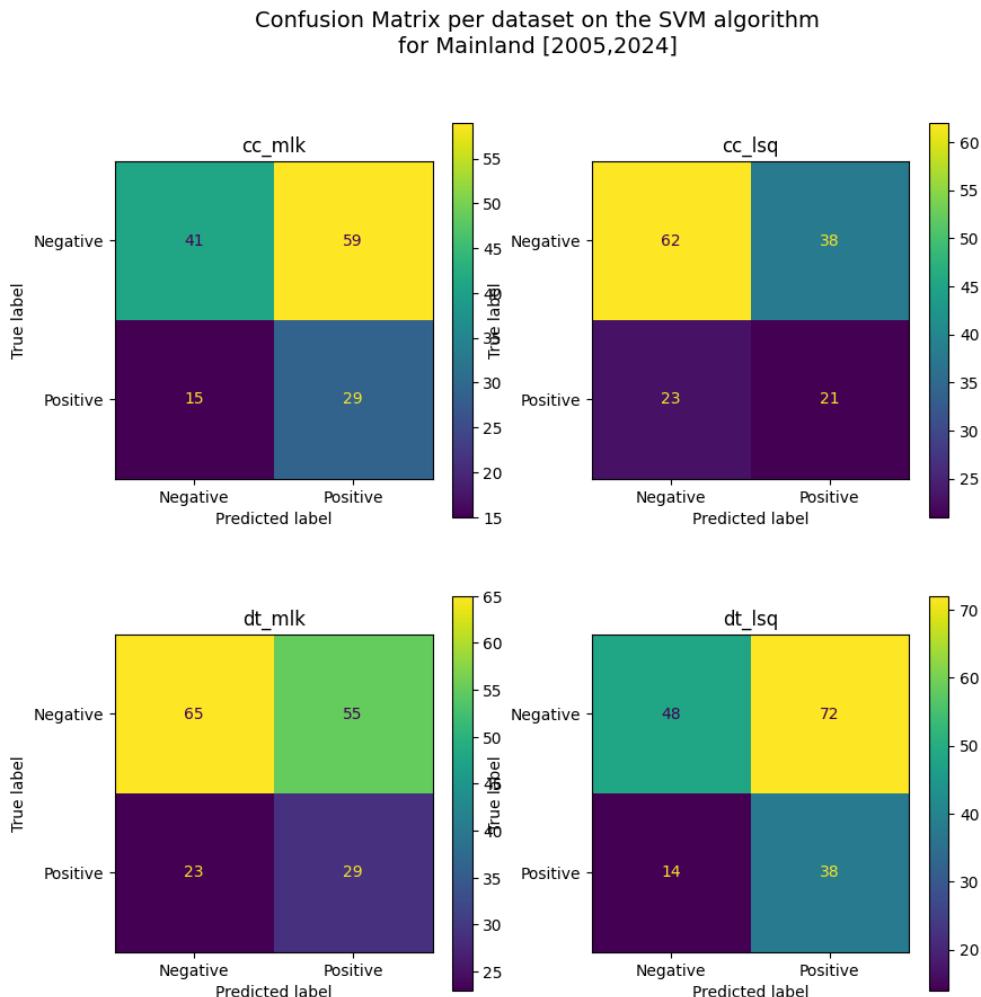


Figure D.13 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the SVM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

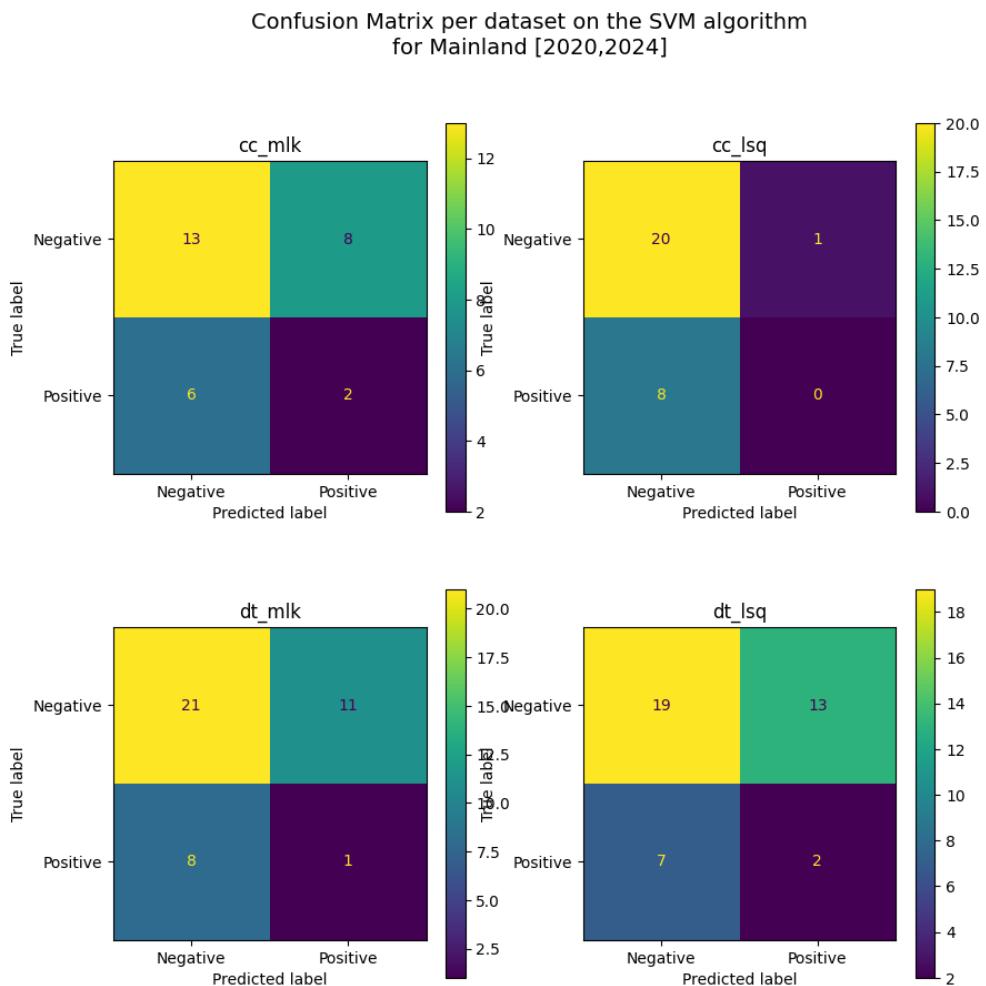


Figure D.14 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the SVM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.7 Metrics values for the Mainland and Surroundings cluster on the SVM algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	65.91	47.43	55.77	73.08	25.00	0.00	11.1	22.22
Specificity	41.00	62.00	54.17	40.00	61.90	95.24	65.62	59.38
P0	73.21	72.94	73.86	77.42	68.42	71.43	72.41	73.08
P1	32.95	35.59	34.52	34.55	20.00	0.00	8.33	13.33
F1	36.62	37.50	37.37	38.62	20.83	0.00	8.77	14.49

NB Algorithm

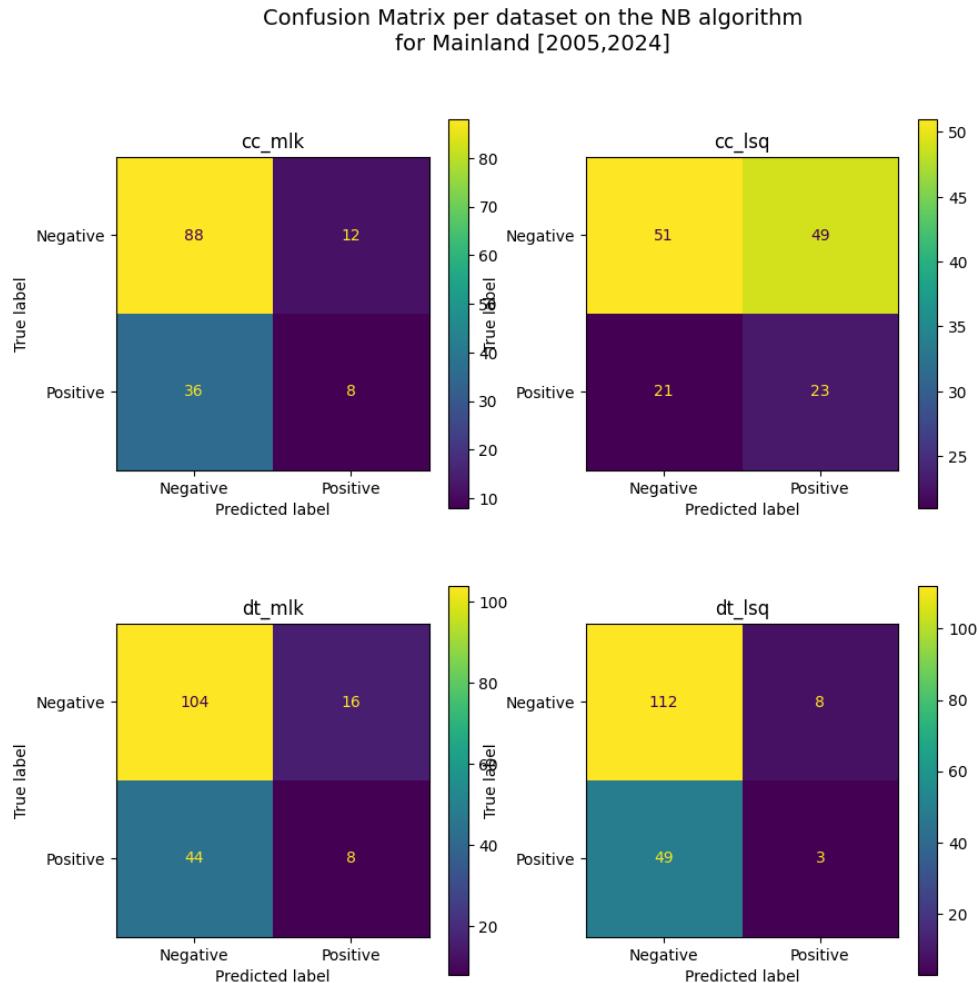


Figure D.15 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the NB algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Confusion Matrix per dataset on the NB algorithm
for Mainland [2020,2024]

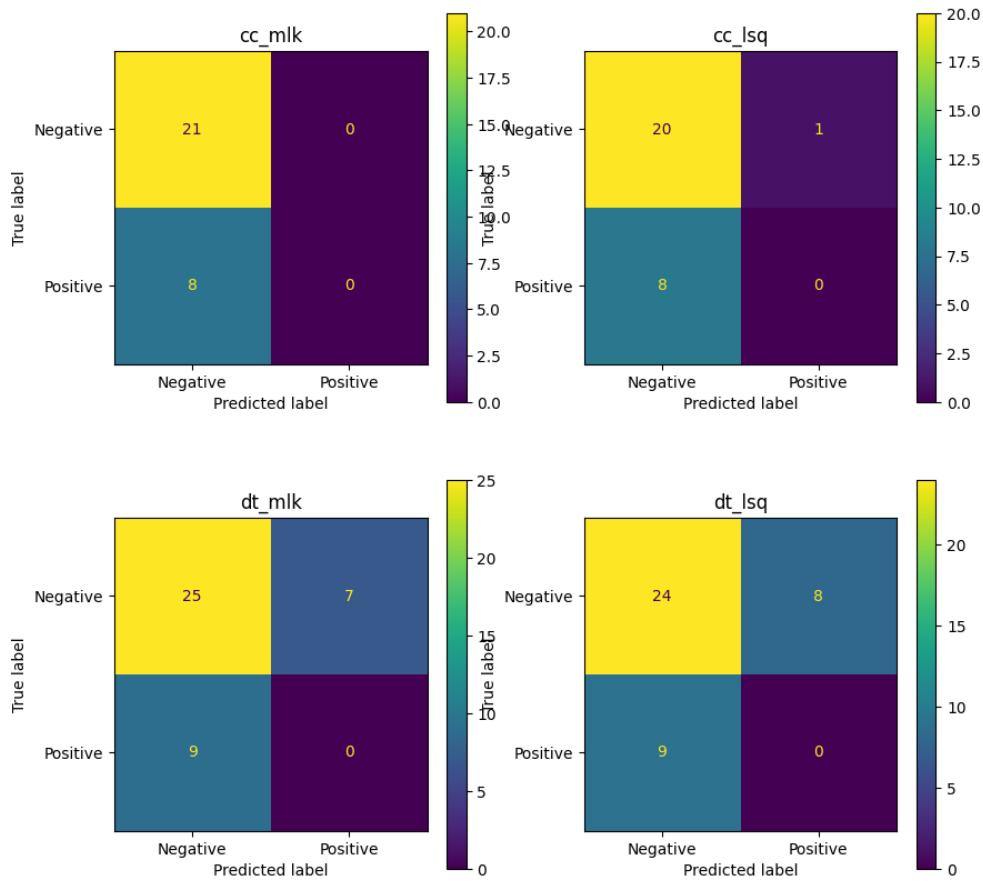


Figure D.16 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the NB algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.8 Metrics values for the Mainland and Surroundings cluster on the NB algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	18.18	52.27	15.38	5.77	0.00	0.00	0.00	0.00
Specificity	88.00	51.00	86.67	93.33	100.00	95.24	78.12	75.00
P0	70.97	70.83	70.27	69.57	72.41	71.43	73.53	72.73
P1	40.00	31.94	33.33	27.27	0.00	0.00	0.00	0.00
F1	32.26	34.64	27.03	15.62	0.00	0.00	0.00	0.00

RF Algorithm

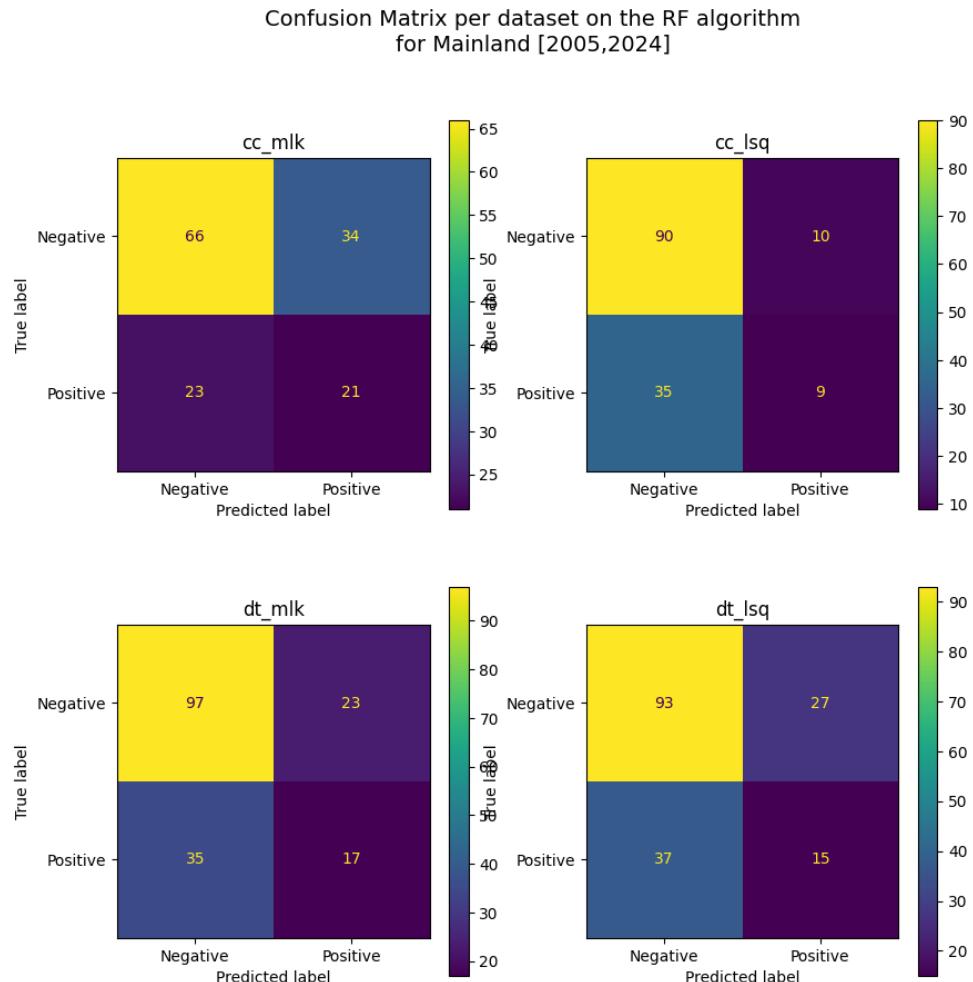


Figure D.17 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the RF algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Confusion Matrix per dataset on the RF algorithm
for Mainland [2020,2024]

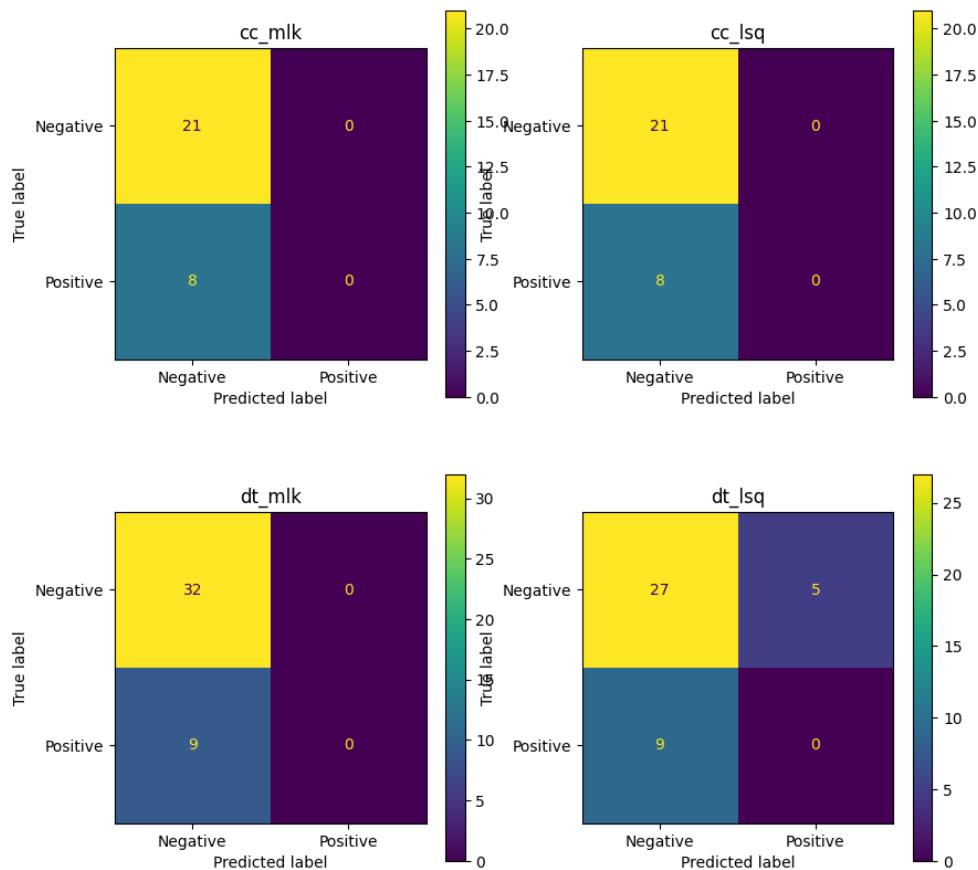


Figure D.18 Confusion Matrix results for the Mainland and Surroundings [2020, 2024] cluster on the RF algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.9 Metrics values for the Mainland and Surroundings cluster on the RF algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	47.73	20.45	32.69	28.85	0.00	0.00	0.00	0.00
Specificity	66.00	90.00	80.83	77.50	100.00	100.00	100.00	84.38
P0	74.16	72.00	73.48	71.54	72.41	72.41	78.05	75.00
P1	38.18	47.37	42.50	35.71	0.00	0.00	0.00	0.00
F1	39.77	37.50	40.09	34.09	0.00	0.00	0.00	0.00

LSTM Algorithm

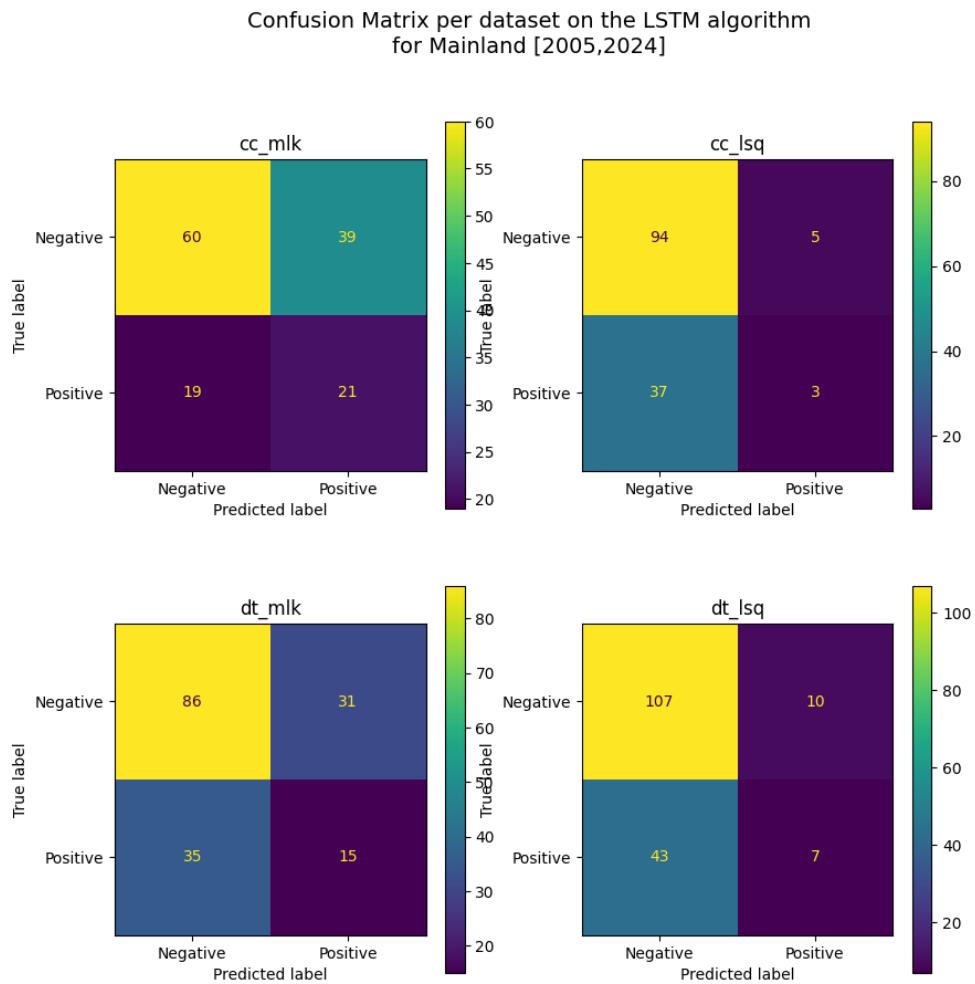


Figure D.19 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Confusion Matrix per dataset on the LSTM algorithm
for Mainland [2020,2024]

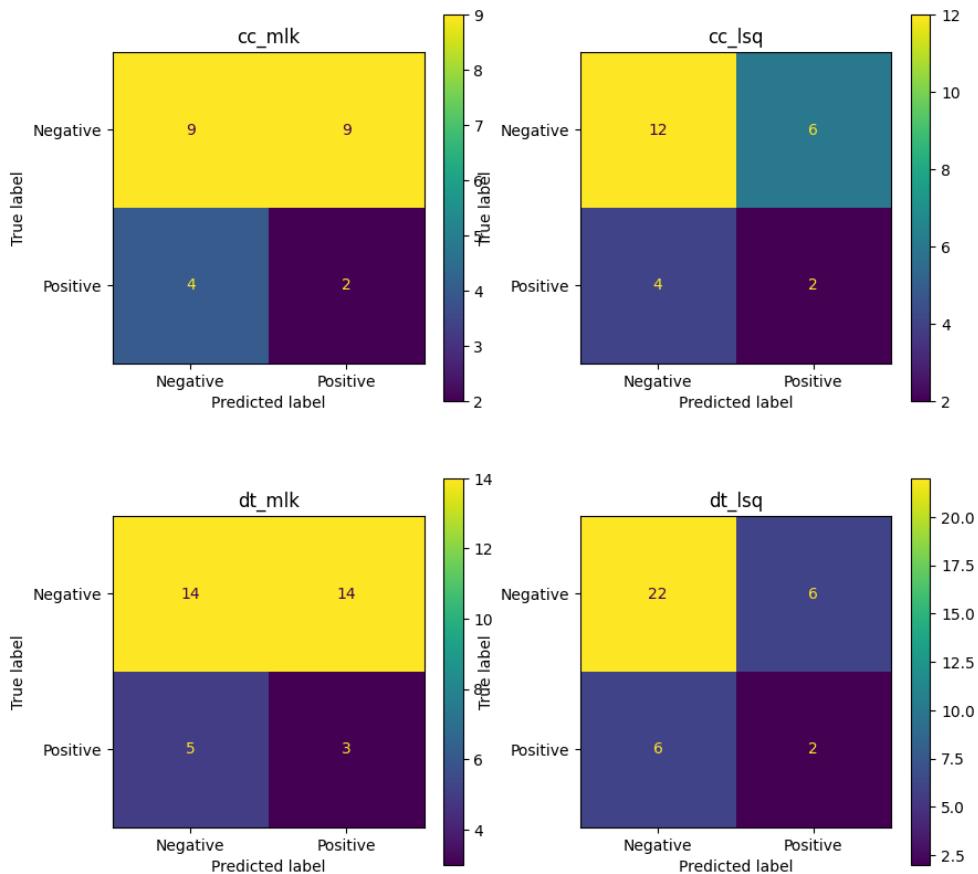


Figure D.20 Confusion Matrix results for the Mainland and Surroundings [2005, 2024] cluster on the LSTM algorithm, with dt referring to the time window method and cc the occurrence. The lsq and mlk identify the estimation method.

Table D.10 Metrics values for the Mainland and Surroundings cluster on the LSTM algorithm.

	Occurrence [2005,2024]		Time [2005,2024]		Occurrence [2020,2024]		Time [2020, 2024]	
	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq	Mlk	Lsq
Sensitivity	52.50	7.50	30.00	14.00	33.33	33.33	37.50	25.00
Specificity	60.61	94.95	73.50	91.45	50.00	66.67	50.00	78.57
P0	75.95	71.76	71.07	71.33	69.23	75.00	73.68	78.57
P1	35.00	37.50	32.61	41.18	18.18	25.00	17.65	25.00
F1	37.50	20.83	32.05	29.66	20.00	26.32	19.74	25.00

ANNEXES

Results obtained on previous studies on the Western Azores-Gibraltar region. Retrieved from the last published study (Martínez-Álvarez et al., 2013).

Parameter	ANN [30]	ANN [35]	New ANN
TP	5	5	8
TN	64	52	63
FP	5	17	6
FN	5	5	2
Sensitivity (%)	50.00	50.00	80.00
Specificity (%)	92.75	75.36	91.30
P_0 (%)	92.75	91.23	96.92
P_1 (%)	50.00	22.73	57.14
Mean (%)	71.38	59.83	81.34

Parameter	NB [30]	NB [35]	New NB
TP	4	10	4
TN	51	0	53
FP	18	69	16
FN	6	0	6
Sensitivity (%)	40.00	100	40.00
Specificity (%)	73.91	0.00	76.81
P_0 (%)	89.47	0.00	89.83
P_1 (%)	18.18	12.66	20.00
Mean (%)	55.39	28.16	56.66

Parameter	M5P [30]	M5P [35]	New M5P
TP	10	0	5
TN	0	69	48
FP	69	0	21
FN	0	10	5
Sensitivity (%)	100	0.00	50.00
Specificity (%)	0.00	100	69.57
P_0 (%)	0.00	87.34	90.57
P_1 (%)	12.66	0.00	19.23
Mean (%)	28.16	46.84	57.34

Parameter	SVM [30]	SVM [35]	New SVM
TP	5	6	5
TN	22	36	47
FP	47	33	22
FN	5	4	5
Sensitivity (%)	50.00	60.00	50.00
Specificity (%)	31.88	52.17	68.12
P_0 (%)	81.48	90.00	90.38
P_1 (%)	9.62	15.38	18.52
Mean (%)	43.25	54.39	56.75



NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa