Distributed Data Analysis and Mining

University of Pisa

Project by

Giuseppe Muschetta

Master Degree: Data Science and Business Informatics

Enrollment Number: 564026

Grade for the Project: Awarded 30 cum laude by Prof. Roberto Trasarti

Contents

Introduction

In this report, we are going to analyze the World Earthquake dataset, which can be downloaded on the "kaggle" platform by clicking here. To accomplish the task, we will utilize some of the tools learned during the "Distributed Data Analysis and Mining" course. The main tasks we will tackle are:

- **Data Understanding**
- **Data Preparation & Regression**
- **Clustering**
- **Classification**

We will approach these tasks using the Python programming language with ***PySpark***. PySpark is an interface for Apache Spark in Python. It not only allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analyzing your data in a distributed environment. PySpark supports most of Spark's features such as Spark SQL, DataFrame, Streaming, MLlib (Machine Learning) and Spark Core.

We also used other libraries such as Pandas, Matplotlib, Seaborn, Folium, Graphviz, etc. always in the context of visualisation. All computations and machine learning algorithms were solely performed using Spark.

Short explanation of the attributes

c0 Automatically generated index.

time Date and time of the earthquake, often in UTC format.

latitude Geographical latitude of the earthquake, ranging from -90 to +90 degrees.

longitude Geographical longitude of the earthquake, ranging from -180 to +180 degrees.

depth Depth of the earthquake origin below the surface, typically in kilometers.

mag Magnitude of the earthquake, indicating its relative size or energy.

magType Scale used to calculate the earthquake's magnitude (e.g., ML, MD).

nst Number of seismic stations used to record the earthquake.

gap Azimuthal Gap, the largest angle between two stations from the epicenter, in degrees.

dmin Shortest distance from the epicenter to the nearest station.

rms Root Mean Square of time residuals of seismic waves at different stations.

net Network code that provided the earthquake data.

id Unique identifier for each earthquake event.

updated Timestamp of the last update to the earthquake record.

place Textual description of the earthquake's location.

type Type of seismic event, e.g., earthquake, quarry blast.

horizontalError Uncertainty in the earthquake's epicenter location, horizontally.

depthError Uncertainty in the earthquake's depth measurement.

magError Error associated with the earthquake's magnitude measurement.

magNst Number of stations used to calculate the earthquake's magnitude.

status Review status of the earthquake record (e.g., automatic, reviewed).

locationSource Source of the earthquake's location information.

magSource Source of the earthquake's magnitude information.

1 Data Understanding

In this first phase of data understanding, we will not touch or change the dataframe, we will simply understand the data, visualise the statistics and distributions of the most relevant variables, having established how they relate to each other.

The dataset consists of 23 columns and 3,272,774 data points and tracks major earthquakes from 1970 until 2020. The columns with **numeric** values are: *time, latitude, longitude, depth, mag, nst, gap, dmin, rms, updated, horizontalError, depthError, magError, magNst* while the **categorical** ones (they are all nominal) are: *magType, net, place, type, status, locationSource, magSource.*

The following table shows us, for each categorical attribute, how many unique values we have.

Feature	**Unique Values**
magType	31
net	20
id	3,256,955
place	205,437
type	25
status	4
locationSource	199
magSource	333

Table 1: Number of unique values per categorical feature

Among the categorical attributes, only *magType*, *Type* and *Place* are interesting to us. As we will see in the next phase of data preparation, the *place* column will be used in the post-analysis, the magType column, which has 31 unique values, will be transformed with only the three unique values of greatest interest, and the *type* column will be used exclusively to eliminate all the relatively few data points that do not represent earthquakes, so as to have only rows in the dataset that concern earthquakes.

Once this is done, this column will be deleted and this choice will be justified in the classification phase.

1.1 Missing values

The table 2 shows the missing or null values for each attribute in the dataframe. We have sorted the table in descending order by the percentage of missing values. Obviously, columns such as *magError* and *horizontalError* are excluded for two reasons: firstly, because of their large number of missing values and, as we will see later, secondly, because all the variables that tell us about the error in the measures, including *depthError*, are of little importance for our study.

Despite the very high quantity of null values present in columns such as *dmin, magNst, nst* and *gap*, these columns, which are of great importance for our analysis, will not be eliminated but rather will be imputed with various methods.

In detail, we will impute the *dmin* and *gap* variables using their mean, which is justifiable by examining their distributions in Figure 2. For more critical variables to our subsequent analysis, such as *nst* and *magNst*, we have imputed their values using a ***Linear Regression*** model and a ***Decision Tree Regressor***, respectively.

Feature	Total Nulls	Percentage Nulls
magError	1781012	54.419
horizontalError	1531963	46.809
dmin	1346742	41.149
magNst	988917	30.217
nst	881566	26.936
gap	838549	25.622
depthError	606685	18.537
rms	211653	6.467
magType	167407	5.115
mag	156449	4.780
place	11	0.00033
depth	9	0.000275
status	1	0.000031

Table 2: Missing values statistics for the earthquake dataset.

1.2 General statistics of numerical variables

We start by looking at the statistics of the numerical variables (Table 3) so that we can immediately understand any errors and inconsistencies that may be present in the data. In fact, statistics such as **mean, standard deviation, min** and **max** value provide us an excellent basis for understanding whether there are errors or inconsistencies in the data.

For example, attributes like *mag* and *depth* cannot have negative values, and in the table we can see that their minimum value is a negative value, such points should be eliminated. The same applies to other columns such as *rms* and *depthError*, where the maximum value is several orders of magnitude higher than the mean and standard deviation. Obviously these are outliers and we will eliminate them.

Since PySpark API does not have built-in algorithms such as LOF or DBSCAN, with which one could perform a multivariate outlier detection and removal, we will content ourselves with removing outliers locally, i.e. column by column: we are talking about the removal of **univariate outliers** using the IQR method. The IQR is defined as the difference between the 75th percentile (Q3) and the 25th percentile (Q1) of the data. Outliers are typically defined as values below Q1 - *threshold**IQR or above Q3 + *threshold**IQR. The default value for threshold is 1.5, but this value is known to be very restrictive, so we will try different values for the threshold.

Observation: we're writing these lines after conducting, in Chapter 3.1, further exploratory clustering using the nine most significant numerical columns for our study. Thanks to the k-means algorithm, we were able to identify 32 multivariate outliers that were making the statistics for some variables truly inconsistent.

Therefore, the outliers have been removed, but with the certainty that we only eliminated those that can genuinely be defined as such in a multivariate context (which is much preferable) rather than a univariate one.

1.3 Distributions of the most relevant attributes

Since standardization only alters the scale of variables (the purpose is to bring all the variables on the same scale subtracting the mean and dividing for the standard deviation), not the distribution shape, we deemed it prudent to standardize the columns of greatest interest for our study before plotting them. These columns include *mag, depth, nst* and *magNst*, as illustrated in Figure 1.

Summary	Count	Mean	Stddev	Min	Max
Latitude	3,272,774	35.721	20.257	-84.422	87.265
Longitude	3,272,774	-92.857	80.553	-179.999	180.000
Depth	3,272,765	22.335	56.320	-10.000	735.800
Mag	3,116,325	1.879	1.353	-9.990	9.100
Nst	2,391,208	15.602	26.607	0.000	934.000
Gap	2,434,225	130.488	69.711	0.000	360.000
Dmin	1,926,032	0.256	1.333	0.000	141.160
Rms	3,061,121	0.315	0.400	-1.000	104.330
HorizontalError	1,740,811	1.267	3.168	0.000	280.600
DepthError	2,666,089	5.640	1167.801	-1.000	1,773,552.500
MagError	1,491,762	0.167	0.147	0.000	6.110
MagNst	2,283,857	12.607	21.127	0.000	941.000

Table 3: Statistical Summary of Earthquake Dataset Numerical Variables

Additionally, though not crucial, two other attributes have been plotted: *rms* and *dmin*, shown in Figure 2.

As we can easily observe, the column we will base our classification on, ***mag***, exhibits a distribution that is fairly Gaussian, or rather, normal-like, whereas all the others plotted, starting with *depth*, display a Pareto distribution. Indeed, among the Pareto distributions that naturally occur in our world, earthquakes are a prime example, thus underscoring the consistency in our findings.

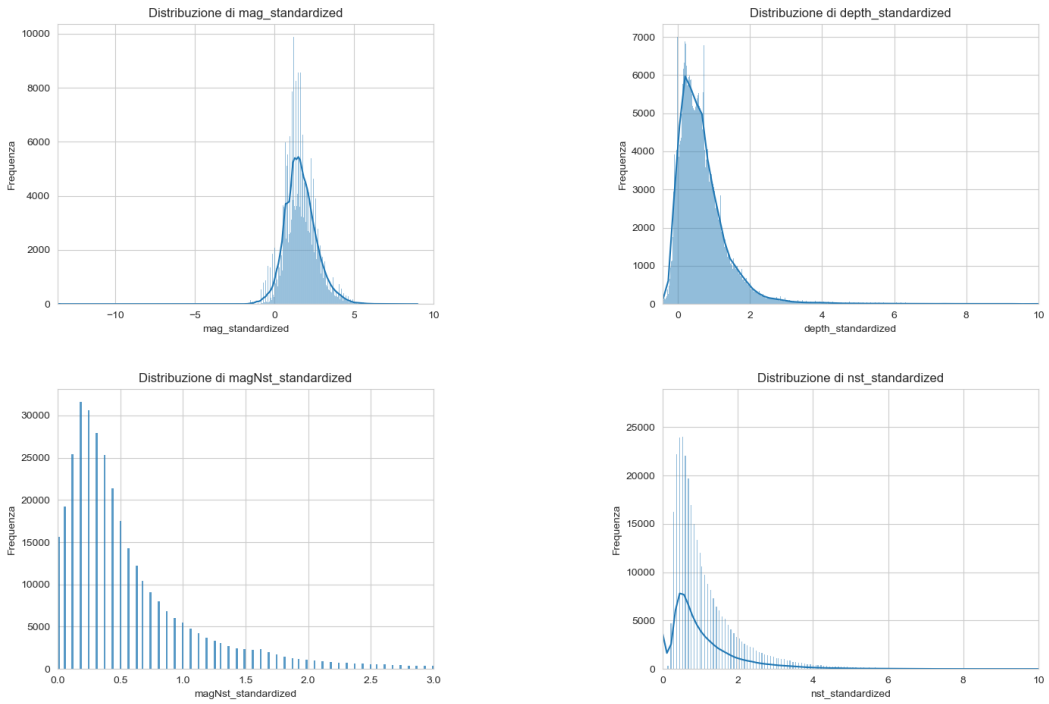

Figure 1: Standardised distributions of mag, depth, nst and magNst

Equally useful in our analysis are the distributions of the variables *dmin* and *rms*, which will help the classifier to correctly predict earthquakes.

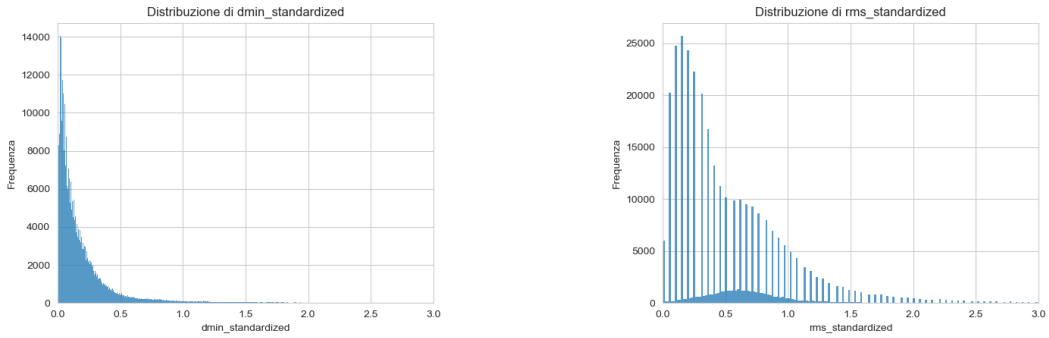

Figure 2: Standardised distributions for the variables dmin and rms.

1.4 Correlations between continuous attributes

As we can see in Figure 3, here we find justification for all the variables selected for our subsequent analysis, namely those that exhibit sufficient correlation with the variable *mag*. In Spark, correlations are calculated exclusively using Pearson's coefficient, which, unlike Spearman's rank coefficient, can only capture linear relationships between variables, and not nonlinear ones as Spearman's correlation coefficient can.

2 Data Preparation and Regression

In the initial stages of data preparation, our goal was to eliminate univariate outliers from specific columns. This decision stemmed from an analysis of the general statistics of the numerical variables, as indicated in Table 3, which revealed disproportionately high maximum values in comparison to the mean and variance. We opted to use the interquartile range (IQR) method with a preliminary threshold of 1.5. This threshold, usually considered restrictive, was subsequently adjusted to encompass two or three orders of magnitude greater. Nonetheless, after a threshold of 10, there were no further changes in the remaining rows of the dataframe. Post-clustering, we observed a significant reduction of earthquake occurrences on continents such as Europe, Asia, and Oceania. This led to the realization that what we had initially labeled as outliers were, in fact, relevant data, especially considering that our dataset encompasses earthquakes globally and is not restricted to any specific area.

For these reasons, we decided against removing these univariate outliers. Instead, as stated in the observation written in section 1.2, we ended up eliminating multivariate outliers using k-means clustering.

We have performed other critical data cleaning procedures, such as:

- Removal of duplicate rows
- Deletion of certain features that provided no significant information such as *status, id* and other features such as *magError, horizzontalError, depthError* have been deleted because they had a very large quantity of null values and also were poorly correlated with the target variable ***mag*** that we will use for our study in the next stages. Also columns such as *locationSource, magSource and net* have been deleted because they really provided poor information even in the post clustering or classification phase.
- Correction of data inconsistencies, such as negative values in the columns *mag* and *depth*

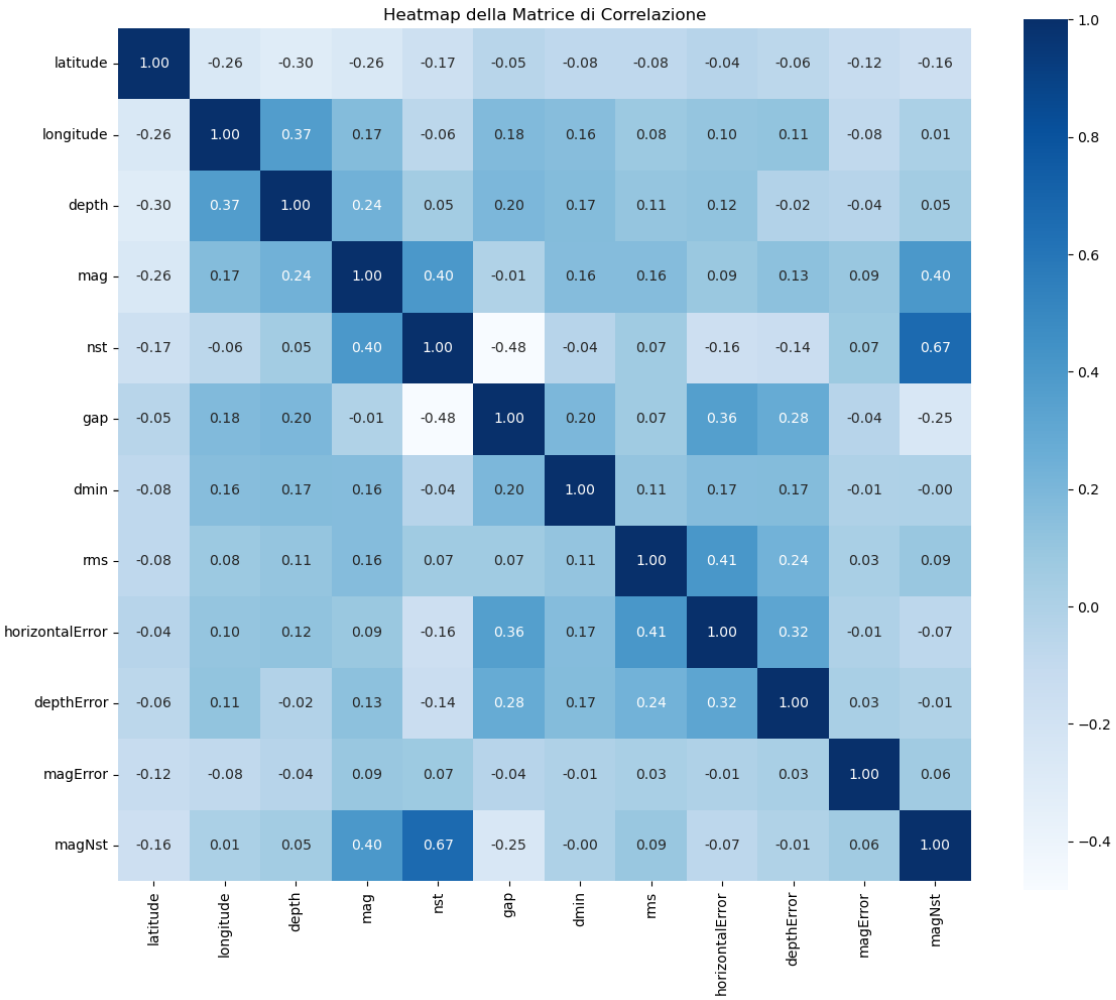

Figure 3: How all numerical features correlate

- In column *magType* all strings representing unique values are converted to lower case in order have all the unique categories written properly
- The attribute *magType* has more then 20 unique categories, so we decided to keep the 2 most frequent, *ml* and *md* and cluster the other in one category named *other*. This solely for the purpose of simplifying the classification model that we will train later.
- Removal of the few null values belonging to columns like *'mag', 'depth', 'rms', 'place', and 'magType'*
- Elimination of non-earthquake related rows from the *type* column, followed by the removal of the column itself, focusing the dataset solely on earthquakes for classification.
- Imputation of columns like *dmin* and *gap* using their mean values, given their distribution (Figure 2).
- For crucial columns like *nst* and *magNst*, imputation was performed using a **Decision Tree Regressor** (using predictors like *latitude, longitude, depth, mag, rms*) and a **Random Forest Regressor**, respectively.

After imputation, a new correlation matrix was calculated, revealing that correlations among continuous variables remained largely consistent with previous findings. Figure 4 illustrates the correlations among variables post-imputation.

Attribute	**Type**	**Attribute**	**Type**
time	timestamp	*gap*	double
latitude	double	*dmin*	double
longitude	double	*rms*	double
depth	double	*updated*	timestamp
mag	double	*place*	string
magType	string	*nst*	double
magNst	double		

Table 4: Columns in the final cleaned dataframe

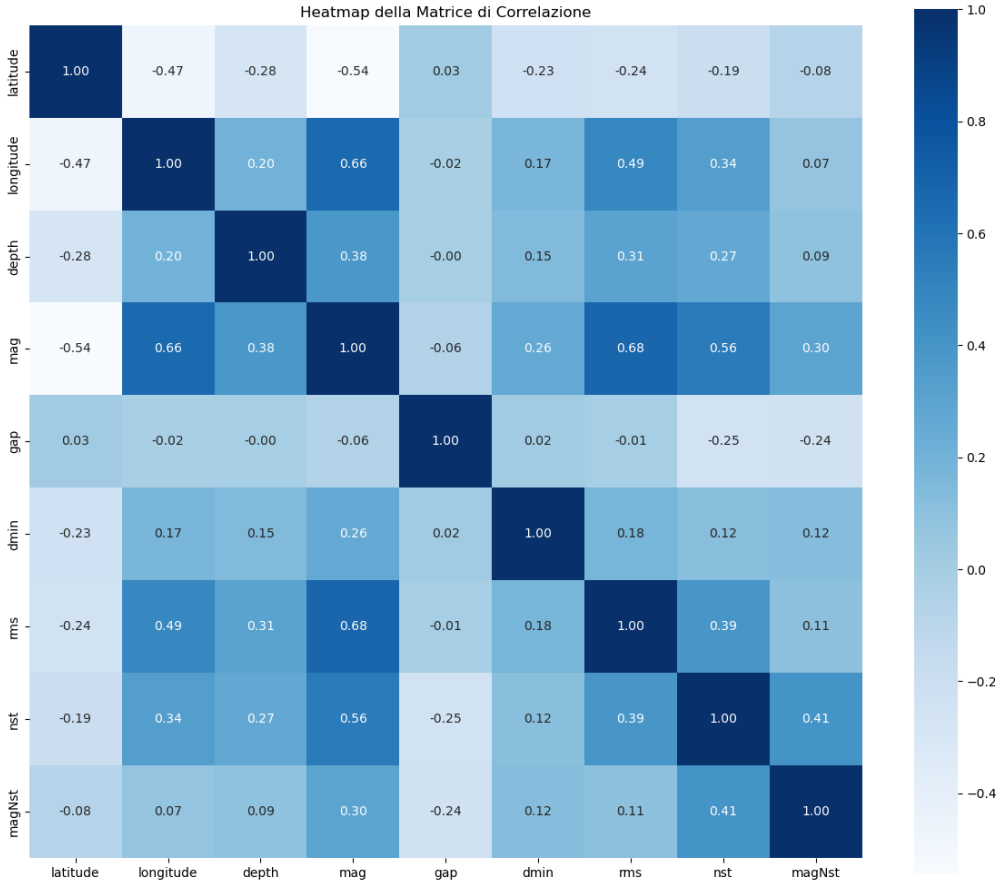

Figure 4: Correlation matrix after attribute imputations

After all these modifications we prepared a cleaned dataset consisting of 13 columns (Table 4) and 2,607,588 rows:

We're ready to save the cleaned .csv file and move on to the next step, ***Clustering***.

3 K-Means Clustering

The first and most crucial step in clustering will be carried out at an exploratory level, which is essentially the primary reason why clustering is often used. The second phase, on the other hand, will involve visualizing the clusters on a world map. This approach allows for an easy identification of individual earthquakes within each cluster and demonstrates how each cluster corresponds to different areas of the planet.

In both phases, the method of Inertia or SSE (Sum of Squares Error) and the method of Silhouette are employed to determine the optimal number of clusters for dividing global earthquakes.

3.1 Further Dataset Exploration

To perform clustering using the k-means algorithm, we utilized the following 9 columns:

latitude, longitude, depth, mag, nst, magNst, gap, dmin, rms.

Since k-means is a distance-based algorithm, it is crucial to standardize these columns before clustering. This ensures that all variables are on the same scale, preventing imbalances in the distances that could lead to errors in the results.

After standardizing the relevant variables, we calculated the most appropriate value for the hyper-parameter k using the Elbow method (which uses Inertia or SSE) and the Silhouette method as shown in Figure 5 and Figure 6.

Both methods suggested using k=8 as the parameter for k-means.

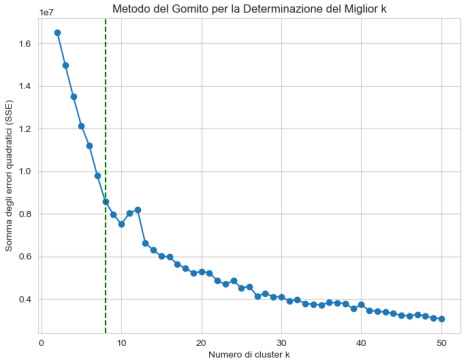

Figure 5: Elbow Method

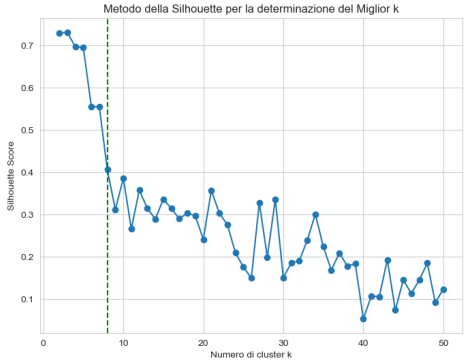

Figure 6: Silhouette Method

Post-clustering, it is common practice to visualize the clusters. However, as we used nine columns, placing us in a nine-dimensional space, we require dimensionality reduction techniques like t-SNE or PCA for two-dimensional plotting on a graph. We opted for PCA due to its simplicity and computational efficiency.

Thanks to PCA, we managed to obtain two principal components, pca_x and pca_y, which enabled us to plot the clusters in two dimensions and finally visualize their structure. The first thing we noticed was some points that were completely off-scale, revealing a few but significant multivariate outliers that we had initially sought but struggled to remove using methods like IQR without causing damage or loss of information.

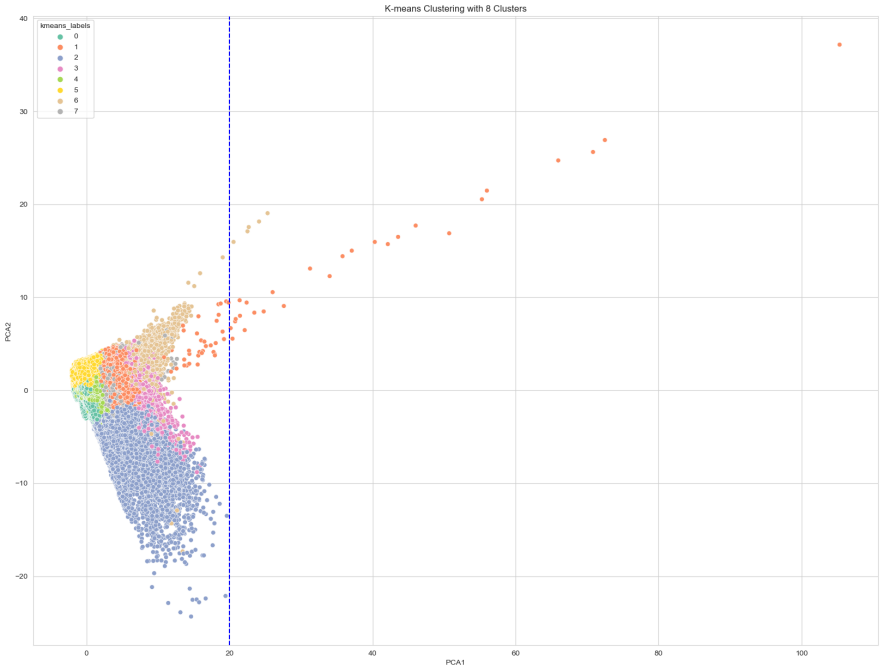

Figure 7: Clusters with outliers

After filtering and removing those 32 outliers, we re-plotted the clusters for a clearer visualization. Notice how the scale on the x-axis is now from 0 to 30, allowing us to better see the clusters shape.

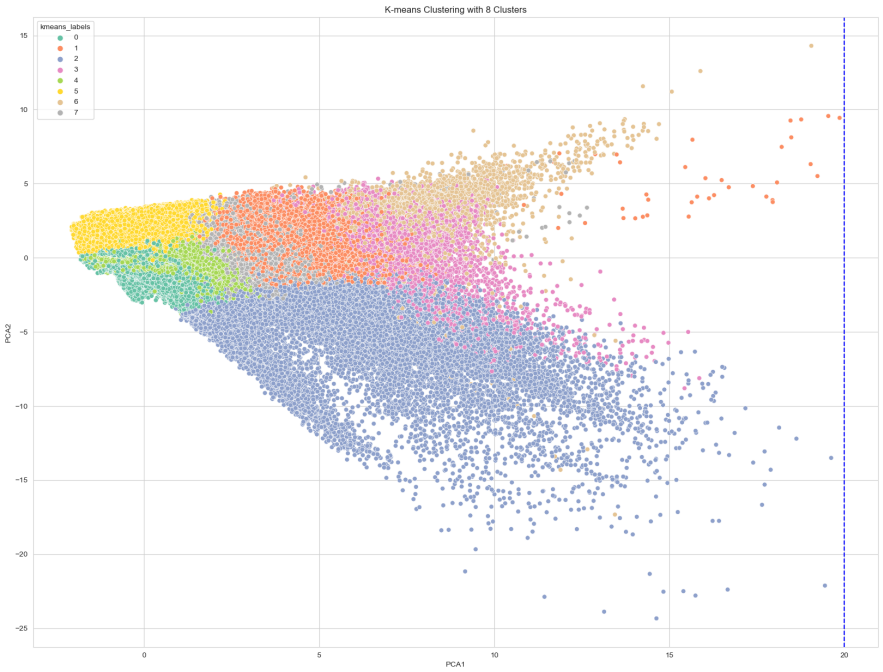

Figure 8: Cluster with no outliers

3.2 Cluster Statistics

Once we have identified our clusters, by calculating statistics such as ***means*** and ***standard deviations***, we can discern what makes each cluster unique and how they differ from one another. Thus, we aim to perform an analysis on each cluster individually, relying on the statistics obtained from an in-depth study of each separate cluster. Additionally, as is often the case, it is likely that in closely situated clusters, minor portions may intermingle with parts of another cluster. However, this is all part of the clustering process and is to be expected. Table 5 will show how clusters are populated.

Cluster	**count**	**percentage**
0	1,228,039	47.10
5	472,058	18.10
4	469,400	18.00
1	199,238	7.64
7	169,897	6.52
2	38,944	1.49
3	25,368	0.97
6	4,612	0.18

Table 5: Clusters and total data points belonging to each one of them

The following are four tables: The first table (Table 6) is obtained by calculating the average values of each attribute for every cluster. The second table (Table 7) accomplishes the same task as the first but includes the standard deviation for each attribute. The third and fourth tables (Tables 8 and 9), on the other hand, consider the maximum and minimum values, respectively, for each attribute within every cluster.

Cluster	**latitude**	**longitude**	**depth**	**mag**	**nst**	**magNst**	**gap**	**dmin**	**rms**
0	36.94	-118.00	6.08	1.31	15.60	12.61	90.95	0.09	0.11
1	-11.91	40.43	57.69	4.52	39.21	13.18	121.43	0.83	0.98
2	26.09	-25.76	39.88	4.21	131.69	112.52	68.64	0.65	0.63
3	-10.40	-41.21	498.41	4.44	56.31	21.51	118.63	1.41	0.86
4	60.24	-151.20	35.33	1.68	24.17	11.82	124.57	0.31	0.55
5	36.98	-117.61	8.51	1.30	7.71	6.57	217.02	0.18	0.12
6	-21.29	3.68	36.49	4.62	42.97	31.05	116.95	20.91	0.75
7	37.70	75.71	33.48	3.91	41.46	11.56	125.10	0.48	0.94

Table 6: The mean of the values belonging to each column is reported for each cluster

Cluster	**latitude**	**longitude**	**depth**	**mag**	**nst**	**magNst**	**gap**	**dmin**	**rms**
0	3.94	13.58	5.12	0.65	13.29	13.47	32.07	0.10	0.08
1	17.46	118.08	57.12	0.54	19.92	14.18	37.85	1.46	0.36
2	22.28	120.89	69.79	1.32	117.77	75.25	45.94	1.35	0.39
3	20.30	160.70	99.65	0.48	41.37	28.42	37.05	2.17	0.24
4	4.95	13.37	38.01	0.76	10.05	8.59	17.59	0.21	0.22
5	5.89	15.85	11.25	0.79	6.54	8.28	49.89	0.23	0.13
6	31.86	101.87	96.11	0.39	8.59	43.38	52.79	8.96	0.27
7	12.16	69.46	41.57	0.85	18.82	13.39	30.05	0.85	0.37

Table 7: Standard deviations of the values for each column reported for each cluster.

Cluster	latitude	longitude	depth	mag	nst	magNst	gap	dmin	rms
0	65.14	179.61	165.0	6.3	155.0	124.0	197.0	7.54	1.0
1	40.13	179.99	305.9	8.3	212.0	109.0	352.6	10.99	19.56
2	86.82	179.99	644.3	9.1	934.0	854.0	334.0	15.97	4.41
3	55.71	179.99	735.8	8.3	646.0	307.0	332.0	14.25	2.03
4	85.15	27.27	300.0	5.9	164.0	104.0	339.0	8.02	4.34
5	71.30	179.99	257.0	6.57	126.0	134.0	360.0	5.65	2.28
6	86.15	179.97	636.59	7.1	90.57	657.0	343.0	102.9	2.0
7	87.22	179.99	324.7	8.3	223.0	100.0	359.0	11.19	12.9

Table 8: Maximum values for each column reported for each cluster.

Cluster	latitude	longitude	depth	mag	nst	magNst	gap	dmin	rms
0	-0.2422	-179.67	0.001	0.01	-3.295	0.0	0.0	0.0	-1.0
1	-84.422	-179.997	0.01	0.16	0.0	0.0	8.0	0.0044	0.0
2	-77.08	-179.9963	0.001	1.29	0.0	0.0	6.5	0.0	0.02
3	-59.483	-179.999	242.41	2.1	5.0	1.0	9.0	0.066	0.07
4	19.5376	-179.9983	0.001	0.01	0.0	0.0	8.0	0.0	0.0
5	-34.105	-179.9955	0.001	0.01	-2.894	0.0	145.4	0.0	0.0
6	-73.462	-179.9479	1.22	1.83	3.0	2.0	16.0	10.735	0.03
7	6.47	-179.924	0.01	0.17	0.0	0.0	9.0	0.001	-1.0

Table 9: Minimum values for each column reported for each cluster.

3.3 Cluster Analysis

Cluster 0

- *Cluster Size*: 1,228,039 data points which correspond to 47% of the whole dataset. By far the biggest cluster of all.
- *Geographic Focus*: Predominantly in the western United States, especially California. Global distribution suggested by the wide longitudinal range.
- *Depth and Magnitude*: Mostly shallow earthquakes with low magnitude, indicating less severe seismic activity.
- *Key Observations*: Represents nearly half of the earthquake events in the dataset, signifying its substantial impact.
- *Additional Insights*: The cluster's spread across a significant longitudinal range indicates a presence of tectonic activity not only localized to California but also affecting other regions along the Pacific Rim.

Cluster 1

- *Cluster Size*: 199,238 data points corresponding to 7.64% of the dataset
- *Geographic Focus*: Wide global distribution.
- *Depth and Magnitude*: Deeper and more intense earthquakes, possibly pointing to more severe seismic events.

- *Additional Insights*: This cluster represent seismic activities along major fault lines worldwide, including subduction zones and rift valleys, where deeper and stronger earthquakes are more prevalent.

Cluster 2

- *Cluster Size*: 38,944 data points (1.49% of the total points in the dataset)
- *Geographic Focus*: Covers a diverse range of geographical locations, indicating a spread across multiple continents and regions.
- *Depth and Magnitude*: Exhibits significant variability, which might indicate a mixture of different types of seismic activities.
- *Additional Insights*: The diversity in depth and magnitude suggests this cluster includes both intraplate and interplate earthquakes, reflecting a complex interplay of geological factors.

Cluster 3

- *Cluster Size*: 25,368 rows for a percentage of 0.97%. We're talking about one of the smallest cluster, but the one which contain the most dangerous earthquakes.
- *Geographic Focus*: This cluster include key regions known for their seismic activity, in fact this cluster contains the most powerful earthquakes ever registered in high known seismic regions such as Japan, Indonesia, Assam, Tibet, Chile etc.
- *Depth and Magnitude*: As already said, this cluster is characterized by extremely deep earthquakes, suggesting significant geological events.

Cluster 4

- *Cluster Size*: 469,400 data points, corresponding to 18% of the dataset, this is the third biggest cluster of all.
- *Geographic Focus*: Northern hemisphere focus, possibly including areas with frequent seismic occurrences.
- *Depth and Magnitude*: Moderate depth and magnitude, indicating a specific type of seismic activity.
- *Additional Insights*: The cluster's location and earthquake characteristics may correlate with tectonic activity in regions like the Pacific Northwest or the northern parts of the Eurasian tectonic plate.

Cluster 5

- *Cluster Size*: 472,058 data points, which is 18.10% of the whole dataset and this is the second biggest cluster.
- *Depth and Magnitude*: Shallow earthquakes with a low to moderate magnitude, suggesting less severe seismic events.
- *Geographic Focus*: This cluster might represent areas with high frequencies of minor seismic activities, such as the Eastern Mediterranean region or the Intermountain West region of the United States.

Cluster 6

- *Cluster Size*: 4,162 points, the smallest cluster of all with a percentage of 0.18%
- *Geographic Focus*: Despite being smaller in size, it covers a wide range of geographical areas, suggesting a diverse set of seismic sources.
- *Depth and Magnitude*: Notable for its high magnitude earthquakes, indicating potentially more impactful seismic events.
- *Additional Insights*: The cluster could include significant seismic events in geologically active regions, such as mid-ocean ridges or areas with a history of large, destructive earthquakes.

Cluster 7

- *Cluster Size*: 169,897 points which is 6.52% of the total.
- *Depth and Magnitude*: Features a varied depth profile and relatively high magnitudes, pointing to significant seismic events.
- *Additional Insights*: This cluster might represent complex seismic activity along major plate boundaries, possibly including both interplate and intraplate earthquakes.

3.4 Geographical Clustering

For the geographical clustering phase, we used columns like ***latitude*** and ***longitude*** and chose the **k-means** algorithm for our model. Before proceeding with k-means clustering, it is standard practice to employ techniques to determine the optimal values for the model's sole hyperparameter, namely estimating the best values for **k**.

The first technique we implemented (Figure 9) is named ***Elbow Method*** and it is about varying k from a minimum of 2 to a maximum of 40 (in our case 40 is a good value because we have sufficient points on the x-axis to visualise a well-shaped curve), during which we calculate the inertia, or the Sum of Squared Errors (SSE).

This process generated a curve where the x-axis represents the values of k within the specified range and the y-axis shows the inertia values. The appropriate value of k is identified at the *elbow* of the curve, where the inertia values cease their steep decline and begin to stabilize at similar levels.

The goal of efficient clustering is to minimize the SSE between the points and centroids.

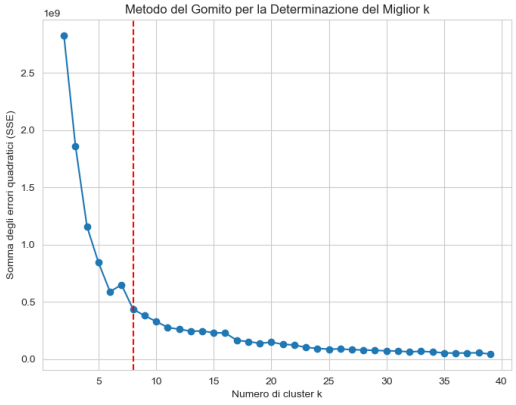

Figure 9: Inertia Curve

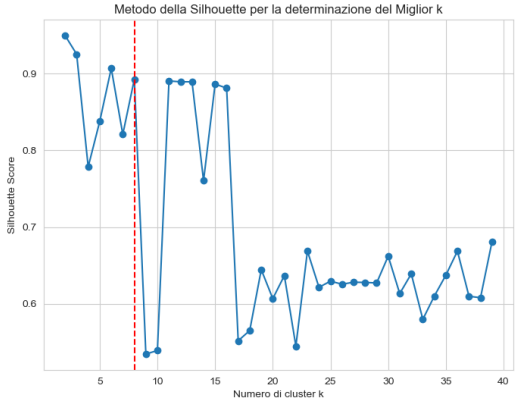

Figure 10: Silhouette Curve

Another method we employed was the ***Silhouette analysis*** shown in Figure 10. Here again, we varied k within the same range and selected values that struck a balance between the number of clusters and a higher silhouette score; this score reflects the quality of the clustering.

After conducting various tests using these methods, we determined that the optimal value of k in our case was k = 8. Thus, we identified 8 clusters to categorize our global earthquake data.

Since we only used two columns, there was no need for dimensionality reduction models like t-SNE or PCA, nor was it necessary to standardize the variables beforehand. Attributes like *latitude* and *longitude* are inherently on the same scale, and maintaining their real scale is beneficial for us. This allows for the visualization of earthquakes on a ***Folium map of the world***, where the clusters can be directly observed. See Figure 12.

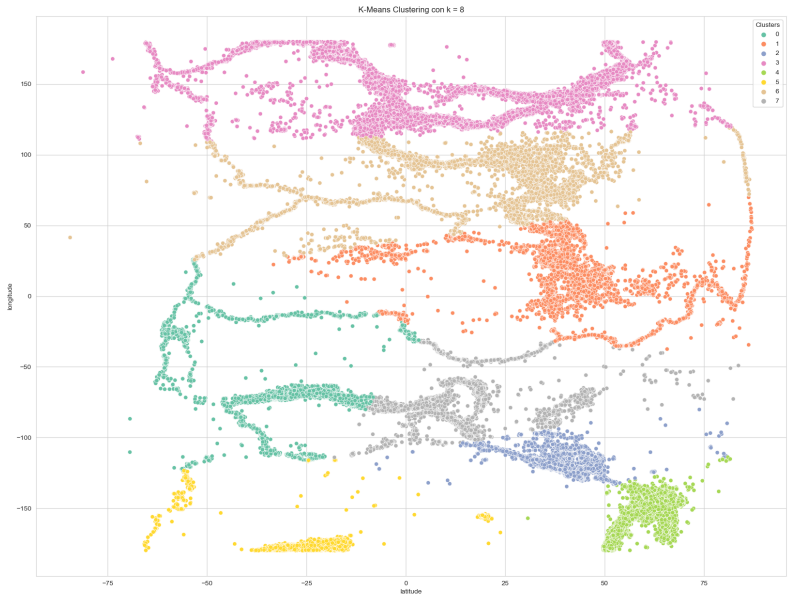

Figure 11: Scatter-Plot showing the clusters with different colors

After determining the most suitable value for the hyper-parameter K, we employed a scatter plot, as shown in Figure 11, in which each cluster is depicted with a distinct color. To achieve colors with good contrast and opacity, we utilized a specific Seaborn palette and continued to use it to maintain consistency in subsequent plots.

kmeans_Labels	**count**	**percentage**
2	1,655,188	63.475
4	469,751	18.015
3	185,497	7.114
1	81,175	3.113
7	59,945	2.299
5	59,550	2.284
6	57,051	2.189
0	39,431	1.512

Table 10: Clusters and total data points belonging to each one of them

In the Table 10 (which is different from Table 5 since we applied clusters on a different number of columns) we have printed some clustering statistics in decreasing order, in which we can easily see that by far the most populated is **Cluster 2** with more than a million and a half data points. As we can see in Figure 12 this cluster is located on the east coast of the United States and mainly affects states such as California and Nevada.

Cluster 4 is the second largest and describe mostly the Alaska zone. **Cluster 3** follows directly with Japan, Indonesia, New Guinea and small islands bordering Australia as areas of interest. *Cluster 1* contains most of the earthquakes recorded in Europe.

Cluster 7 and **Cluster 5** have basically the same number of data points and represent zones such as Mexico, Jamaica and some areas north of Latin America, and Cluster 5 with New Zealand.

Cluster 6 contains some areas from China, India, Singapore, Thailand etc. The smallest is **Cluster 0** and contains all the zones on the east coast of South America.

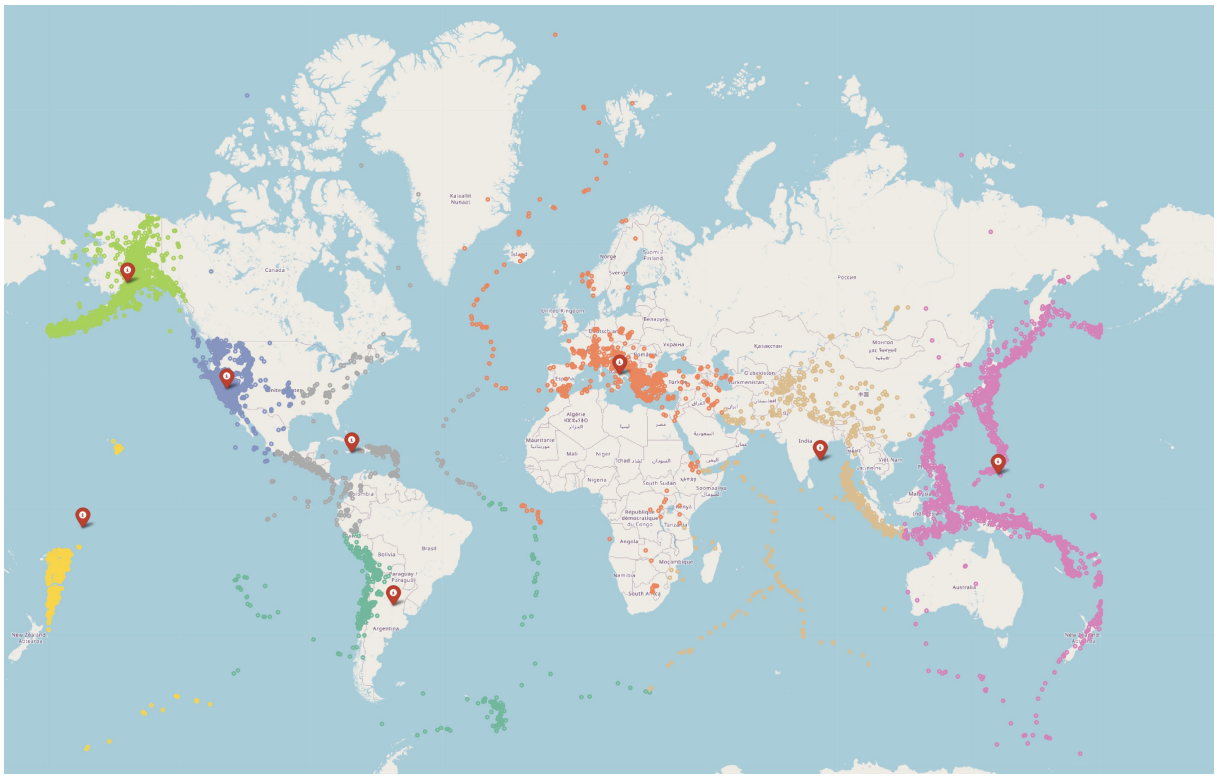

Figure 12: Clusters and their centroids mapped globally with Folium

4 Classification

For classification phase, we have chosen an imbalanced binary classification approach, in fact one class is significantly rarer than the other as we can see in Figure 11.

This imbalance implies that simple *Accuracy* is not anymore an adequate measure to assess the effectiveness of our classification model.

In such skewed contexts, metrics like *Precision* and *Recall* (and consequently the *F1* score, which consists in their harmonic mean) become crucially important. We will also create confusion matrices, which will help us understand the underlying components of metrics like Precision, Recall and Sensitivity: we're talking about the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

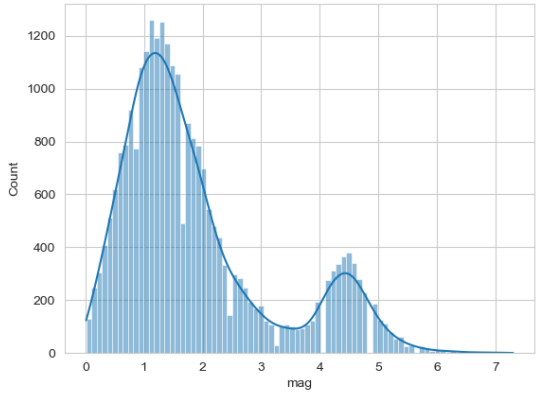

Figure 13: Non-standardised distribution of the mag variable

mag_class	count
Less Severe	2,148,488
More Severe	459,068

Table 11: Distribution of Earthquake Severity

Moving forward, let's delve into what we aim to classify. The objective is to categorize earthquakes based on their magnitude. To achieve this, we have discretized the ***mag*** variable into two distinctly imbalanced categories. The first, less rare category, labeled *0*, represents less severe earthquakes—essentially, basically those that are less dangerous.

Conversely, the rarer category, labeled *1*, is used to predict earthquakes with a high magnitude, i.e., those that are considerably more hazardous.

This distinction has also been done by examining the distribution of the *mag* variable (Figure 13) on its original scale. As we can observe, the distribution is bimodal, suggesting the implicit existence of two distributions within the variable: the largest one for lighter earthquakes and the other, with fewer values, representing the distribution of more powerful and dangerous earthquakes.

Therefore, all magnitude values less than 3 are categorized as minor earthquakes, while all others are considered major.

Initially, for our classification models, we decided to use a single ***Decision Tree***, carefully calibrated, with 5-fold cross-validation and a grid of parameters to derive the optimal model. Subsequently, as a second method, we chose an ensemble approach, specifically a ***Random Forest Classifier***, which is essentially a collection of decision trees.

We believe that numbers and figures speak more than a hundred words, therefore we will use them to understand the effectiveness of our classification models. As such, we rely on tables and data to provide a clear and quantifiable evaluation.

4.1 Decision Tree

We will classify earthquakes according to this:

Less Severe: This category includes earthquakes that might be felt but rarely cause damage. It combines the original categories of "Very Minor", "Minor", "Light", and "Moderate". These earthquakes can cause noticeable shaking but typically do not lead to significant damage to buildings and other structures.

More Severe: This category encompasses the original "Strong", "Very Strong", "Major", and "Devastating" classifications.Earthquakes in this category can cause serious damage in extensive areas. They are capable of causing severe damage in very wide areas, potentially affecting several hundreds or thousands of kilometers.

Table 12: Parameter grid for tuning our Decision Tree Classifier

Parameter	Current Value	Other possible Values
impurity	gini	['gini', 'entropy']
maxDepth	5	[5, 10, 15, 20, 30, 50]
maxBins	32	[16, 32, 64, 128]
minInstancesPerNode	3	[1, 2, 3, 5, 7]
minInfoGain	0.0	[0.0, 0.1, 0.3]
minWeightFractionPerNode	0	[0]
maxMemoryInMB	256	[256, 512, 1024, 2048, 4096]
cacheNodeIds	False	[True, False]
seed	42	[42]

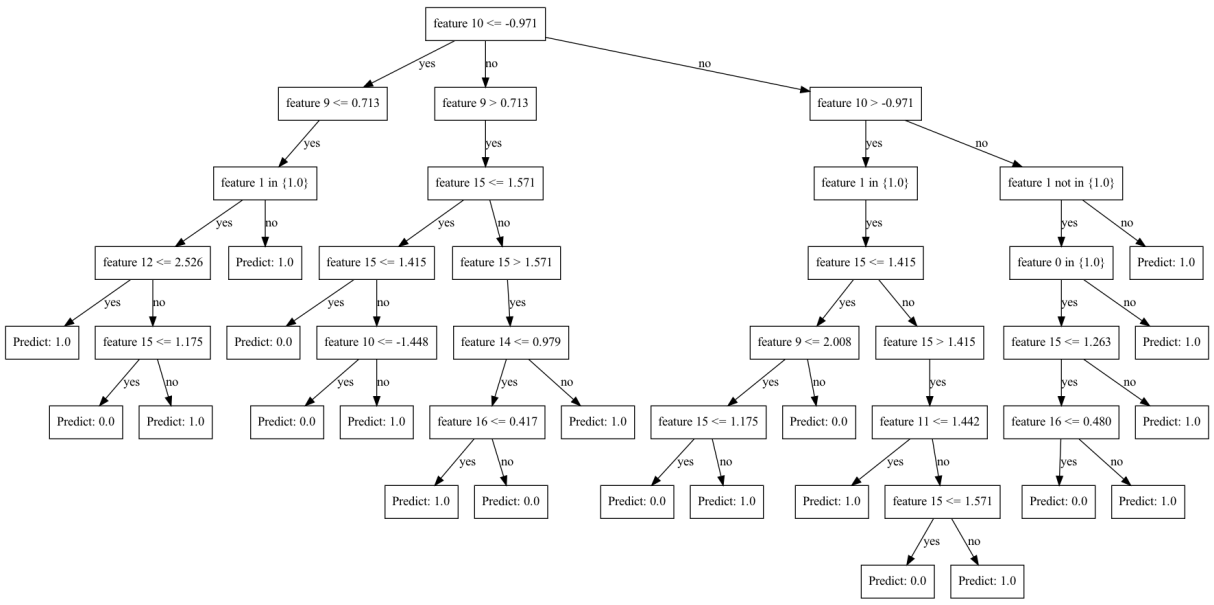

Figure 14: Plot of the Decision Tree obtained using the Graphviz library

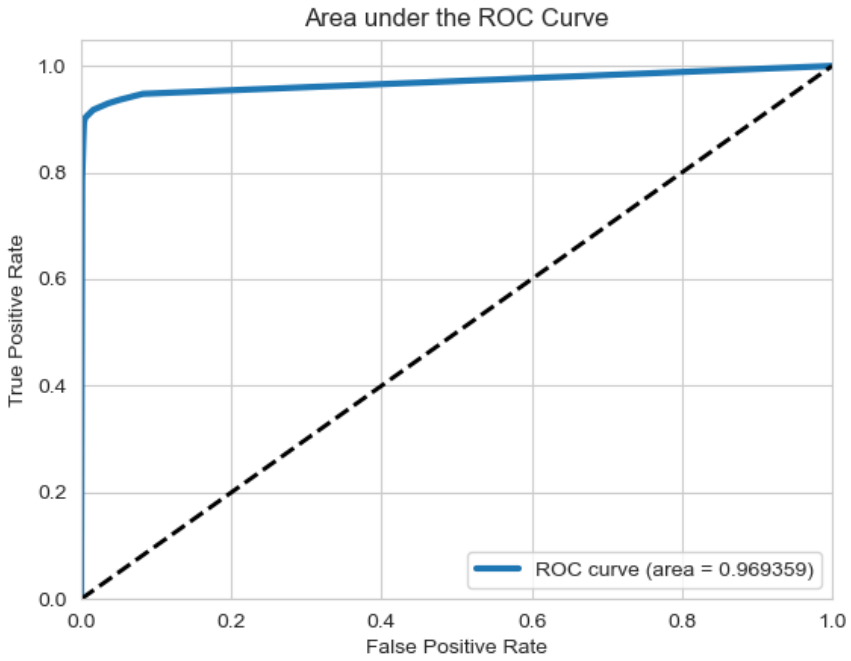

Figure 15: Receiver Operating Characteristic (ROC) curve and AUC obtained for the Decision Tree

Metric	**Value**
Area Under ROC	0.9693587
Accuracy	0.9785664
Precision	0.9785359
Recall	0.9785664
F1 Score	0.9782149

Table 13: Model Performance Metrics

Confusion Matrix		**Predicted**	
		0	**1**
Actual	**0**	TN: 427506	FP: 1963
	1	FN: 9205	TP: 82378

Table 14: Confusion Matrix of the Classifier

Confusion Matrix with Threshold = 0.3		**Predicted**	
		0	**1**
Actual	**0**	TN: 426997	FP: 2472
	1	FN: 8864	TP: 82719

Table 15: Updated Confusion Matrix of the Classifier with Threshold = 0.3 to enhance the prediction of FN

4.2 Random Forest Classifier

Table 16: Parameter grid for tuning a Random Forest Classifier.

Parameter	Current Value	Other Possible Values
maxDepth	10	[5, 10, 15, 20, 25, 30]
impurity	gini	['gini', 'entropy']
numTrees	10	[10, 20, 50, 100, 150]
bootstrap	True	[True, False]
featureSubsetStrategy	auto	['auto', 'sqrt', 'log2', '(0.5)', '(0.25)', '(0.75)']
minInstancesPerNode	50	[1, 5, 10, 20, 50, 100]
maxMemoryInMB	256	[256, 512, 1024, 2048, 4096]

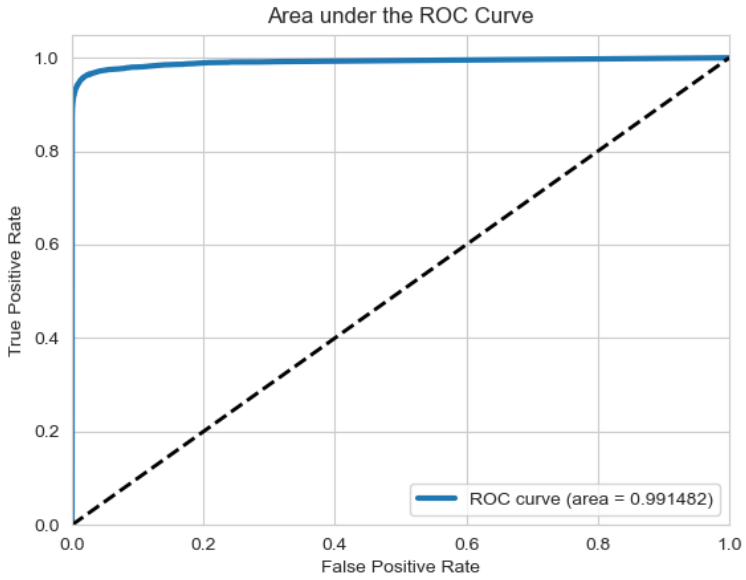

Figure 16: Receiver Operating Characteristic (ROC) curve and AUC obtained for our Random Forest Classifier

Metric	**Value**
Best Model Area Under ROC	0.9914845
Accuracy	0.9834796
Precision	0.9835176
Recall	0.98347957
F1 Score	0.9832477

Table 17: Performance Metrics of the Best Model

Confusion Matrix for Best Model		**Predicted**	
		0	**1**
Actual	**0**	TN: 428286	FP: 1183
	1	FN: 7425	TP: 84158

Table 18: Confusion Matrix of the Best Model

Confusion Matrix with Threshold = 0.3		**Predicted**	
		0	**1**
Actual	**0**	TN: 426011	FP: 3458
	1	FN: 5368	TP: 86215

Table 19: Updated Confusion Matrix of the Best Model with Threshold = 0.3 to enhance the prediction of FN