The year without a summer against 1980–2012 climate: a comparison through machine learning

Simona Gallo, University of Milano-Bicocca, MSc Data Science

Abstract

'The year without a summer' (1816) was an extreme climatic event triggered by one of the most violent volcanic eruptions in human history. In the Northern Hemisphere, the volcanic winter made temperatures drop by several degrees. Although its effect is documented, in scientific literature there are a few impactful comparisons with modern climate. This study aims to show temperature differences between 1816 and 1980-2012, classifying the latter cities' climates (following the Köppen-Greiger system) through machine learning tools and then applying them to the former. In a context of complex time and space relationship within the dataset, traditional statistical techniques are able to manage these issues, and distance-based algorithms demonstrate to be effective and easy to interpret. In particular, K-Nearest Neighbour succeeds in capturing climate characteristics using Manhattan distance. In 1816, several cities switched to the colder climate class, with respect to the average of the period 1980-2012.

Summary

Figures summary

Tables summary

1 Introduction

One of the most recent abnormalities in climate in the Northern Emisphere (especially Europe and North America) dates back to 1816, known as the 'year without summer' [1]. During the early 19th century a succession of cataclysms happened, such as the eruption of Mount Tambora in Indonesia in 1815 and of other volcanoes in the Caribbeans and Philippines: the gases and ashes emitted into the atmosphere obstructed sunlight and caused a drop in temperatures in the Northern Hemisphere by up to 3 Celsius degrees with respect to the period 1971-2000. These phenomena occurred during the Little Ice Age (16th to 19th centuries) and a minimum solar activity, which strengthened the ongoing climate anomalies.

The objective of this project is to compare 1816 to the period 1980-2012 with a different perspective, which is the Köppen-Greiger climate classification, in order to summarize the climate features. This classification, the most used one, categorizes areas based on their temperatures and precipitations:

1. "**A (tropical):** temperature of coolest month 18 °C or higher.
2. **B (arid):** average rainfall below the aridity limit; it may be hot or cold based on the average temperature.
3. **C (temperate):** temperature of warmest month greater than or equal to 10 °C, and temperature of coldest month less than 18 °C but greater than –3 °C.
4. **D (continental):** temperature of warmest month greater than or equal to 10 °C, and temperature of coldest month –3 °C or lower.
5. **E (polar):** temperature of warmest month less than 10 °C." [2]

All climates are exclusive, except for B, and have different microclimates. In scientific literature, the classification is revisited every three decades to keep up with climate evolution. [2] [3]

The data is retrieved from the dataset available on Kaggle.com: https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByCity.csv which contains integrated with data from https://www.weatherandclimate.info/; both of them contain air temperature measured at a height of 2 meters above the ground of cities around the world. Climate classification is obtained via https://www.perplexity.ai/, using https://koeppen-geiger.vu-wien.ac.at/shifts.htm as a source [4].

The dataset considers exclusively temperatures; therefore, the analysis does not examine microclimates and neither cities classified as B. The objective of the study is to train a classification model that can correctly identify the Köppen climate type, based on attributes like coordinates and monthly temperatures from 1980 to 2012, and then apply it to 1816 data, to evaluate if the classification has changed. To do so, the Knime platform is used with the support of R and Python programming languages.

2 Dataset description

The Kaggle dataset has the following columns [5]:

- *dt:* date of the revelation with the format YYYY-MM-DD
- *AverageTemperature:* average land monthly temperature in Celsius degrees
- *AverageTemperatureUncertainty:* the 95% confidence interval around the average
- *City*
- *Country*
- *Latitude*
- *Longitude*

3 Dataset manipulation

The 'dt' column was divided into three different columns ('Year', 'Month', 'Day') and the rows selected are the ones with 'Year' equal to 1816 and in the interval [1980; 2012]. Two columns are removed: 'AverageTemperatureUncertainty', since the documentation does not specify whether the uncertainty is due to the tools used to acquire information or to the temperature excursions during the month, and 'Day', because the data is monthly. To highlight the temperature evolution during the year, 12 new columns are created, each for a month. The rows with the same monthly temperatures in different years or places are deleted, keeping only the first occurrence: removing duplicates reduces the dimension of the dataset and eliminates the data that diminishes the variance. Furthermore, the coordinates (Latitude and Longitude) are adjusted using the Python library '*geopy*', which, with the aid of cities' name and country, also provides altitude for each city. Homonyms of the same nation are removed because the code is not able to manage them.

3.1 Geographical grid and density

This dataset has also a density problem: it contains exclusively cities' coordinates, which leads to an overrepresentation of densely populated areas, while regions with lower population density are significantly underrepresented, resulting in an unbalanced spatial distribution of data points.

The relative frequency of records at high latitudes (over ±60° latitude), which are the coldest and located in tundra or never-melting ice caps, is equal to 4%. Other records at high latitudes and altitudes are added to the dataframe, obtained via https://www.weatherandclimate.info/. To reduce sparsity across the globe, cities with longitudes greater than 170° are removed, i.e. cities from Eastern Russia and New Zealand. To avoid extremely dense areas, a grid with a resolution of 70 kilometers is computed for cities at low latitudes (from 60° to -60°, corresponding to the most populated areas). The distance choice is based on the most used grids in scientific literature, which distinguishes macroclimates between different areas at least 50 km distant [3], and avoids a consistent reduction of the dataset dimension. For higher latitudes the grid has a lower resolution, that is 10 km, because data is already rare. The algorithm removes 107 cities and 3566 rows.

3.2 Feature engineering

Cities from the Southern Hemisphere have a season shift of six months, compared to the Northern Hemisphere. The R function aligns the data, substituting the month variables with the following ones:

‘WI2’, ‘WI3’, ‘SP1’, ‘SP2’, ‘SP3’, ‘SU1’, ‘SU2’, ‘SU3’, ‘AU1’, ‘AU2’, ‘AU3’, ‘WI1’, where ‘WI’ stands for Winter, ‘SP’ for Spring, ‘SU’ for Summer and ‘AU’ for Autumn. The shift is based on the Northern Hemisphere seasons because the 1816 disaster hit Europe and North America.

Once the dataset is divided into the two different periods, the column called ‘Climate’, that contains the Köppen climate classification for each city (excluding the ones with climate B), is joined to the dataframe, leaving only the cities with climate A, C, D, E.

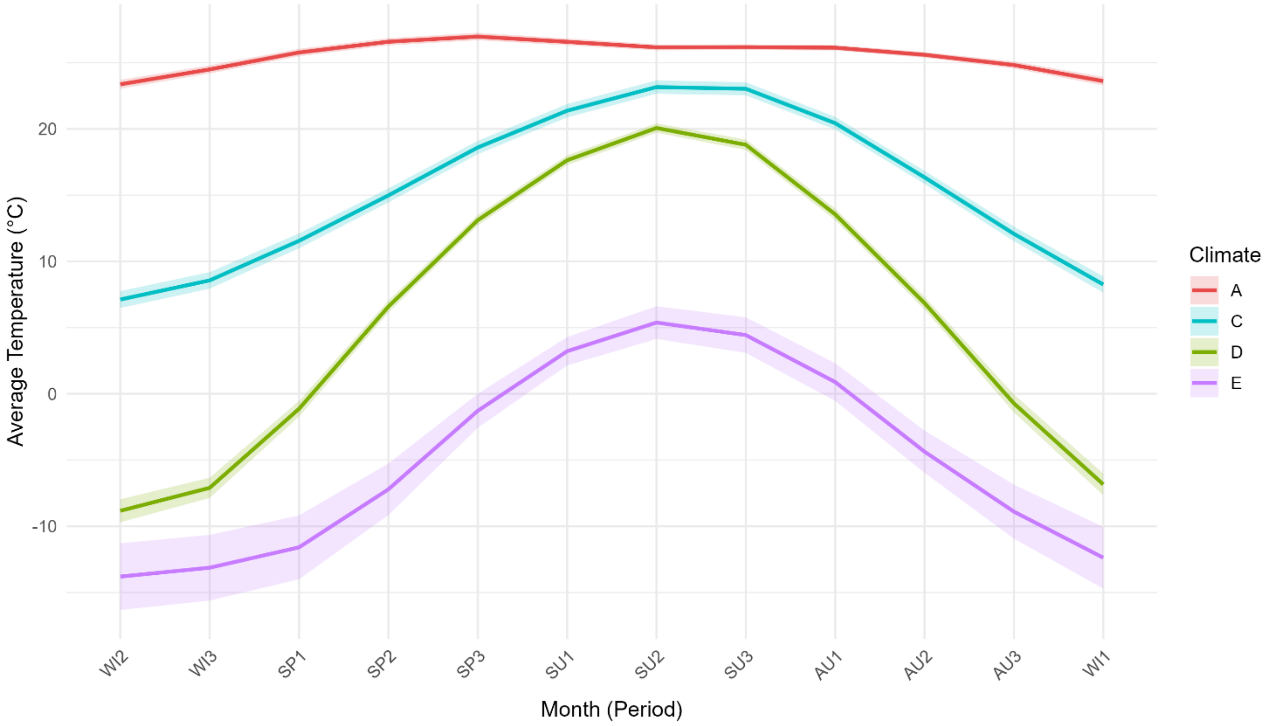

Figure 1. Monthly average temperatures during the year by climate (1980 - 2012).

The temperature evolution during the year is evaluated through Figure 1. The multiple line chart shows the mean temperature (on 1980-2012 period) of the climates during the months, including their standard deviation. During the colder months, both D and E reach -10°C, but E can go down to -15°C [6]. On the other hand, the hotter C months almost arrive at A temperatures, which is over 20°C. However, tropical climates are characterized by constant temperatures. The lines do not intersect, reflecting the fact that the climates are mutually exclusive, and this configuration does so that the clusters are well separated and decision boundaries are not extremely complex, making the classification process easier.

The correlation matrix suggests that all variables, excluding Longitude and Altitude, have a strong linear relationship, as often happens with temporal variables. To have a comprehensive overview of the relationship between the variables that is not strictly related to time, aside from the classic correlation matrix, a map is a useful tool.

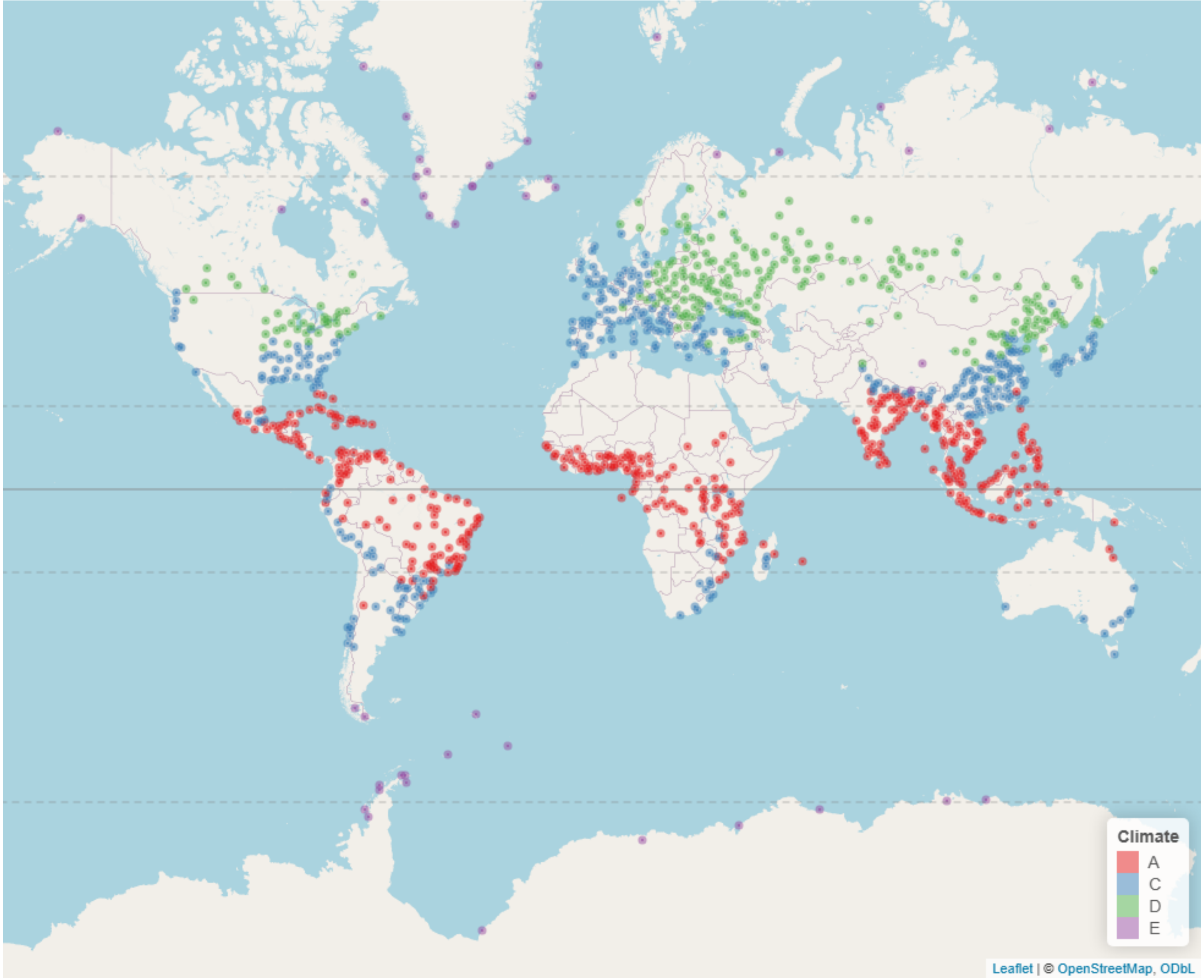

Figure 2. World map of Köppen-Greiger climate for each city.

The map, realized with R library '*leaflet*', shows the direct relationship between latitude and climate type: tropical locations are positioned between the Northern and Southern Tropics (23°26'09 North and South of the Equator), while tundra and polar locations are in proximity of Arctic and Antarctic Circles and a few at extremely high altitudes, like the ones in Tibet. In the Southern Hemisphere there are few cities with climate D (some of them are in New Zealand, excluded for their latitude), because it requires a certain distance from the coast. The ocean mitigates temperature ranges, and it covers about 80% of the Southern Hemisphere, against the 60% of the Northern Hemisphere where there are most of cities with climate D [3].

3.3 Temporal autocorrelation

The dataset with data from 1980 to 2012 is subject to another cleansing: observations that have a significant temporal autocorrelation are removed, using the R library '*purrr*'. This practice does not delete or reduce correlation between temporal variables, neither makes it possible to obtain independent and identically distributed samples because the data has strong temporal and spatial structure. However, it transforms the single time series in a white noise, a process without memory where the past does not have any impact on the future [7]. This way, the effect of seasonal phenomena is mitigated, such as El Niño and La Niña, that respectively warm and cool the climate. The oscillations are not predictable and have a length of at least 12 months. The partial autocorrelation function is computed for each row (i.e. for each city for each year) and column, with a lag equal to 12, that is one year, and a confidence interval of 99%. If the PACF is significant, the row is removed and only the rows with 0 significant lags are kept. For example, if in London the temperature of January 1981 is correlated with the temperature of January 1980, the former is deleted from the dataset. This procedure cancels several rows: for example, 143 observations of the year 2006 are removed (during La Nina) and 101 from 1998, the year of 'Super El Nino' [8]. This process also removes noise and potential outliers caused by seasonal phenomena.

4 Spatial autocorrelation and iterated hold-out

The dataset has two other problems, which are the presence of a rare class and spatial autocorrelation, that can create biased estimates when training machine learning algorithms.

Climate	**A**	**C**	**D**	**E**
Absolute frequency	11224	9957	7422	1221
Relative frequency	37.63%	33.39%	24.89%	4.09%

Table 1. Absolute and relative frequency of Köppen-Greiger climates in the dataset.

The empirical variogram of the variables regarding the temperature (calculated as the mean for each location) and climate is computed with the R library '*gstat*'. The variogram is adapted to the spherical model (a parametric function) to evaluate the point where the spatial correlation fades and the variance becomes asymptotically constant. The points are found minimizing the difference between the residual sum of squares of the empirical variogram $\hat{\gamma}(h)$ and the model $\gamma(h;\theta)$ [9]:

$$\min_{\theta}\sum_{h}\left(\hat{\gamma}(h)-\gamma(h;\theta)\right)^{2}$$

Variable	**Range (km)**
WI2	7897
WI3	8287
SP1	8534
SP2	8846
SP3	8401
SU1	7840
SU2	7656
SU3	7618
AU1	8323
AU2	8294
AU3	8094
WI1	7937
Climate	2238

Table 2. Estimated range of spatial independence (asymptotic variance) for climatic variables.

The function '*cv_spatial*' of the R library '*blockCV*' solves both issues through spatial iterated hold-out: it ensures independence between train and test sets, stratified by 'Climate'. First, data is grouped by coordinates and converted into a spatial object using the geographical system WGS 84; then, the data is projected using the LAEA (Lambert Azimuthal Equal Area) Coordinate Reference System, suitable for statistical analysis [10]. The projection is centred on the point with latitude -20° and longitude -5°, located in the Atlantic Ocean, south-west to Senegal. The centre is chosen empirically, as well as the shape of the spatial blocks, which are hexagonal. This way, the map includes all cities and there is no block with empty classes. The size of the blocks is 8900 km (greater than the maximum value of spatial autocorrelation) to guarantee independence between them.

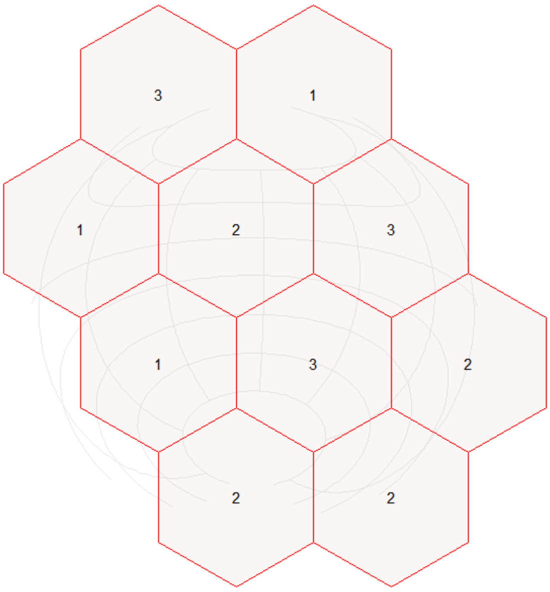

Figure 3. Partition of the globe into 3 folds, with center in latitude -20° and longitude -5°.

The function '*cv_spatial*' divides 3 times the dataset in train and test, which are independent and whose 'Climate' distribution is similar to the original dataset, indicated by the red line in Figure 4. Every train and test pair are made up of about 2/3 of the data for training and 1/3 for testing.

Hold-out	**Train**	**Train (%)**	**Test**	**Test (%)**
1	20821	69.81%	9003	30.19%
2	20489	68.7%	9335	31.3%
3	18338	61.49%	11486	38.51%

Table 3. Distribution of records across Train and Test sets for each hold-out.

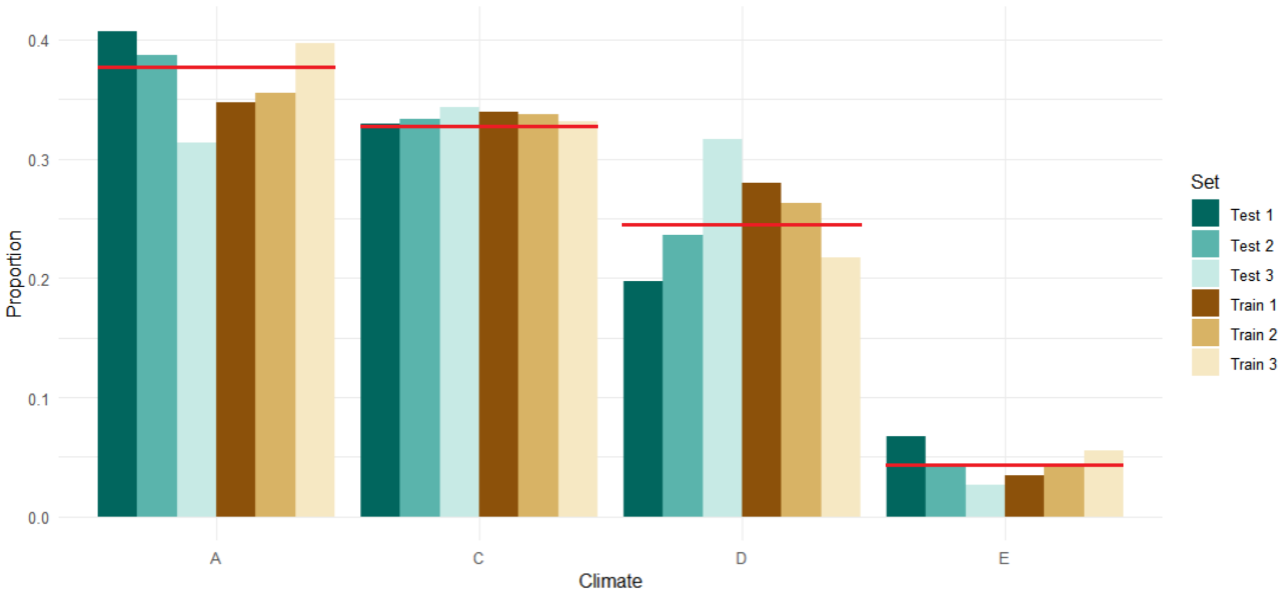

Figure 4. Distribution of 'Climate' across the Train and Test sets.

5 Choice of the models

The temporal and spatial structure of the dataset do so that observations are not independent, hypothesis required by many classification models such as Random Forest, Multinomial Logistic Regression, Gradient Boosting and others. Furthermore, there is a strong correlation between the temporal variables (the ones that contain temperature data); consequently, they are not independent and identically distributed (i.i.d.). This framework makes incorrect the application of one of the aforementioned models: the results would suffer from bias and estimation instability problems. The most appropriate solution is the employment of distance-based algorithms that are able to classify already labelled data [11].

Two algorithms are implemented for the classification of the data points: Nearest Centroid and K-Nearest Neighbours, the only ones that are exclusively based on distance between observations with a supervised approach. Two variations of the Minkowski distance (in formula) are considered: Manhattan (called L1, with q = 1) and Euclidean (called L2, with q = 2) [12].

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^q\right)^{\frac{1}{q}}$$

While L2 is the most popular one, it may be sensitive to outliers and overestimate distances. On the other hand, L1 calculates the distance exclusively on the n-axis of the n-features, making it more robust. However, Cityblock distance may suffer in a context where data is not uniformly distributed. Given that, both distances are used in the two algorithms and compared to evaluate which one is most appropriate. Mahalanobis distance is not used, even though variables have strong linear correlation: the aim is not to model time-space relations and attributes have distributions that are difficult to transform into a gaussian. For example, '*Latitude*' and temperature-related variables are attributable to Beta functions, '*Longitude*' follows a sinusoidal shape and '*Altitude*' a Chi-square distribution.

K-NN and NCC do not estimate feature importance and treat them as equally relevant. In this context it is not a problem since all variables have a consistent impact on climate distinction. Latitude, longitude and altitude are unique locations' identifiers and WI2, WI3, …, AU3, WI1 are significant in climate definition, e.g. tropical climate needs that the temperature is above 18°C all year round.

The metrics used to evaluate models' performances are Sensitivity and Specificity, computed for each class and each hold-out on the test set.

5.1 K-Nearest Neighbour

K-Nearest Neighbour algorithm classifies data points based on the class of the k objects close to the one considered, closeness determined by values of the distance matrix. Thus, it is essential that each group has a certain number of neighbours, otherwise classification of the most frequent group dominates over the other ones and performances are worsened. Given that and the relative frequency of the climates (seen in Table 1), before the implementation of the K-NN, the Synthetic Minority Over-Sampling Technique (SMOTE) is applied on the three train sets, using the '*DMwR*' R library. The percentage of oversampling and undersampling is chosen empirically, with the goal of having balanced classes, and differently for each train set. After the SMOTE, climate A has a relative frequency of 28%, climate C 26%, climate D 21% and climate E 25% in all train sets; then, the variables are normalized within the [0;100] interval.

The K-NN algorithm is computed with Manhattan and Euclidean distance and for each k from 1 to 25, using the '*kknn*' R library. Sensitivity and specificity are put together in a table that places metrics of each hold-out side by side, differentiating by k. Then, results are compared, counting how many times K-NN with L1 outperforms K-NN with L2: this procedure makes the comparison immediate and facilitate a deeper evaluation. The table highlights that K-NN with cityblock distance classifies better, especially with climate C and E. Climate C is the most difficult to classify because, as shown in Figure 1 and Figure 2, it is near both climate A and climate D, not just geographically, but also in terms of similar summer temperatures. On the other hand, climate E has similarities with climate D in terms of winter temperatures.

The metrics of K-NN with Manhattan distance are now evaluated across different k values, shown in Figure 5.

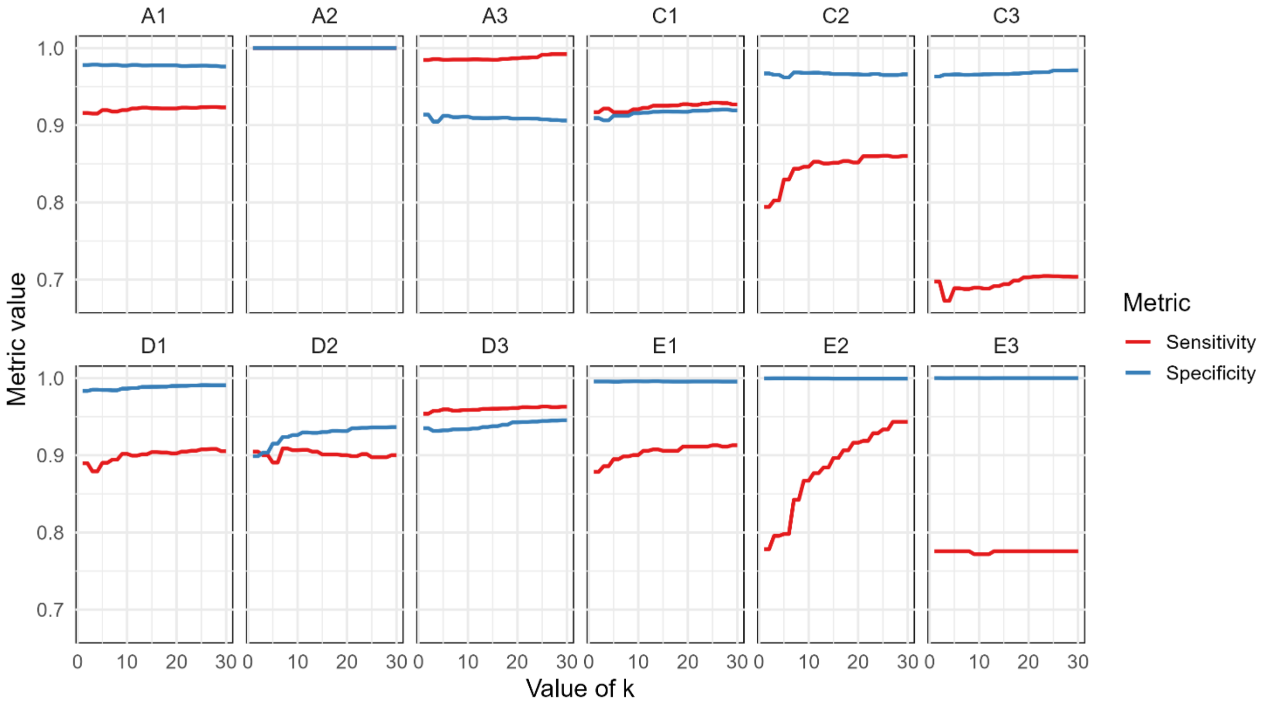

Figure 5. Sensitivity and Specificity of the Test sets across k values of K-Nearest Neighbour with Manhattan distance.

The highest Sensitivity and Specificity values are in correspondence of k ≥ 25. The k chosen for the K-NN is equal to 25 because the results are satisfying, having all metrics greater than 0.70, and there is no considerable gain with k > 25, saving computational time.

5.2 Nearest Centroid Classifier

The Nearest Centroid Classifier (NCC) is one of the simplest machine Learning algorithms: it computes the centroid for each class and assigns the data point to the class whose centroid is closer [13].

The Train variables are normalized using their range, in order to configure them with the same scale, and applied to the Test. The algorithm structure does not require any other pre-processing technique, such as SMOTE, because its focus is on the distance between the data point and the centroid of the class, which cannot be consistently modified since there lies the distinction between climate classes.

Two classifiers are computed, the first with Manhattan distance, the second with Euclidean distance. The results are first compared counting how many times NCC-L2 performs better than NCC-L1. Class C is, once again, the most difficult to classify; while its True Negative Rate is high with both distances, its True Positive Rate is improved when Euclidean distance is used. All classes benefit somehow from L2, especially D, and the cases where L1 is better than L2 do not have a magnitude that justifies the use of L1 instead of L2.

5.3 Comparison between algorithms

The chosen algorithms are K-Nearest Neighbours with K = 25 and Manhattan distance and Nearest Centroid Classifier with Euclidean distance. The models are compared using their values of Sensitivity and Specificity for each hold-out structure: results are not aggregated to have a comprehensive overview.

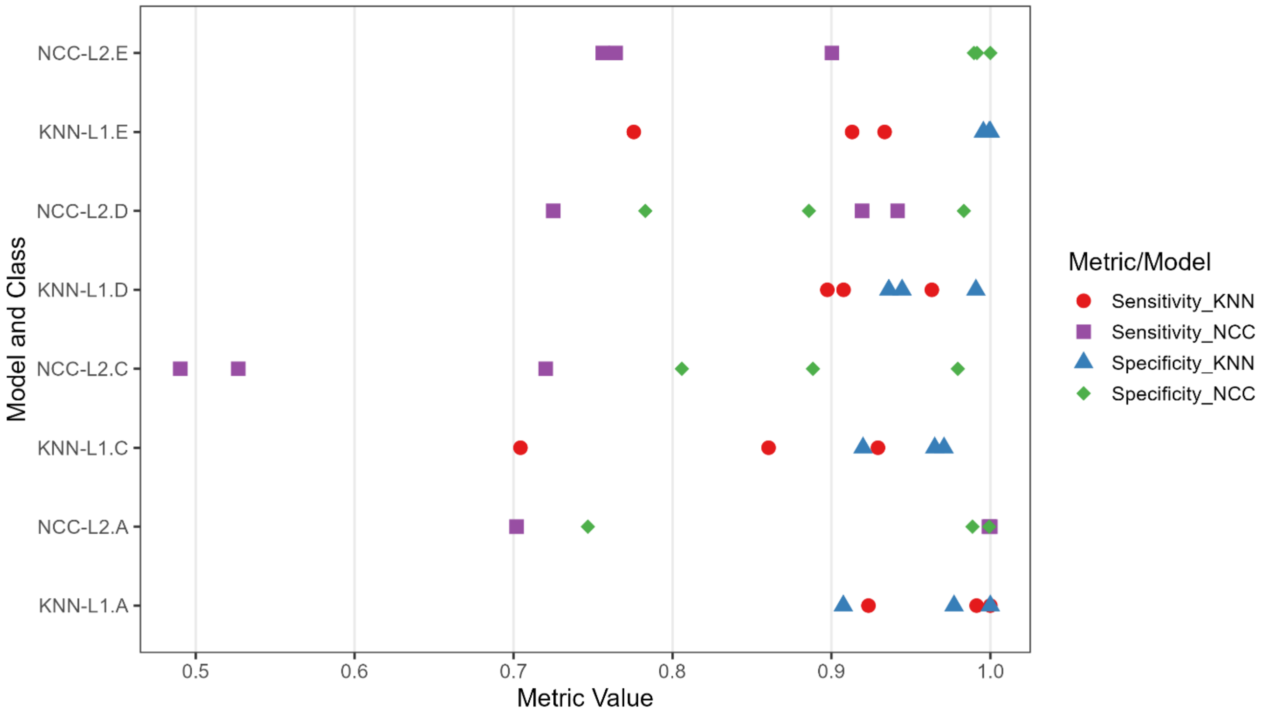

Figure 6. Sensitivity and Specificity of the Test sets by class (A, C, D, E) and models (K-NN and NCC).

K-Nearest Neighbour outperforms NCC, excepts for climate E, where their results are in the same range that spans from 0.75 to 1. In particular, in the worst scenario (that is hold-out 3) KNN is able to correctly classify climate C 7 times out of 10, while NCC only 5 times out of 10, with a TPR = 0.49. Thus, K-NN is the best one, even if it is way slower than its competitor.

The metrics average of each class, weighted by the number of records of the test sets, are computed with the following formula:

$$\bar{M}_i = \frac{\sum_{j=1}^{3} M_{i,j} \cdot N_j^{test}}{\sum_{j=1}^{3} N_j^{test}}$$

In short, the K-Nearest Neighbour with k = 25 and Manhattan distance correctly classifies observations 9 times out of 10.

	A	**C**	**D**	**E**
Sensitivity	0.973	0.821	0.926	0.867
Specificity	0.958	0.954	0.956	0.998

Table 4. Weighted average of K-NN metrics.

6 Results

To compare whether the classification has changed from 1816 to the period 1980-2012, only the cities in both datasets are considered, which are 347. From the 1980-2012 dataset only the second hold-out configuration is chosen, because, as shown in Figure 4, it is the one with the classes' distribution closer to the original dataset. Rows belonging to the Test2 set are removed, SMOTE is applied the same as in the previous section and the dataset is normalized. Then, data from 1816 is classified. To facilitate the comparison, it is created a new dataframe with rows from 1816 and rows with the mean temperature of the period 1980-2012. Of these 347 cities, the K-Nearest Neighbour algorithm finds that 49 of them changed their climate over the two centuries and switched to the hotter climate class, that is climate C to A, D to C and E to D.

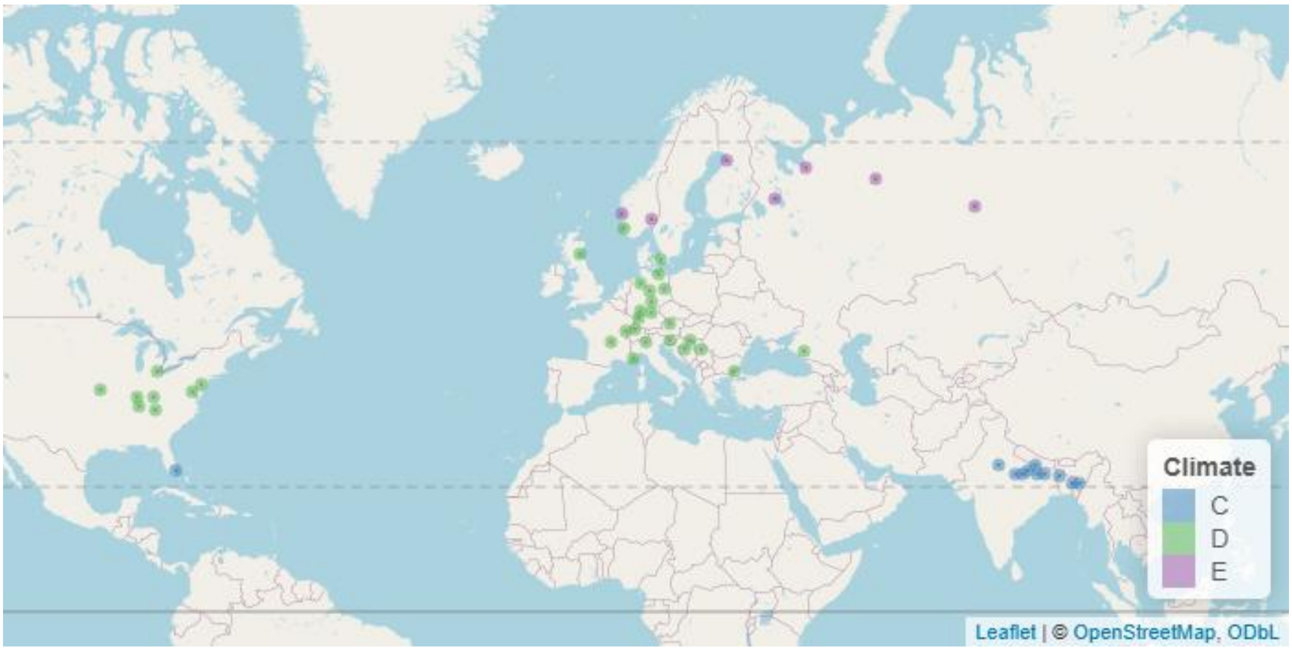

Figure 7. Map of cities and their Köppen-Greiger climate in 1816.

In 1816, there were few cities with tundra climate, even if they are below the Polar Circle. There are no cities with tropical climate, including the ones that lies above the Tropic of Cancer. The mean monthly temperature diminished of several degrees, with a peak in Norwegian and Russian cities.

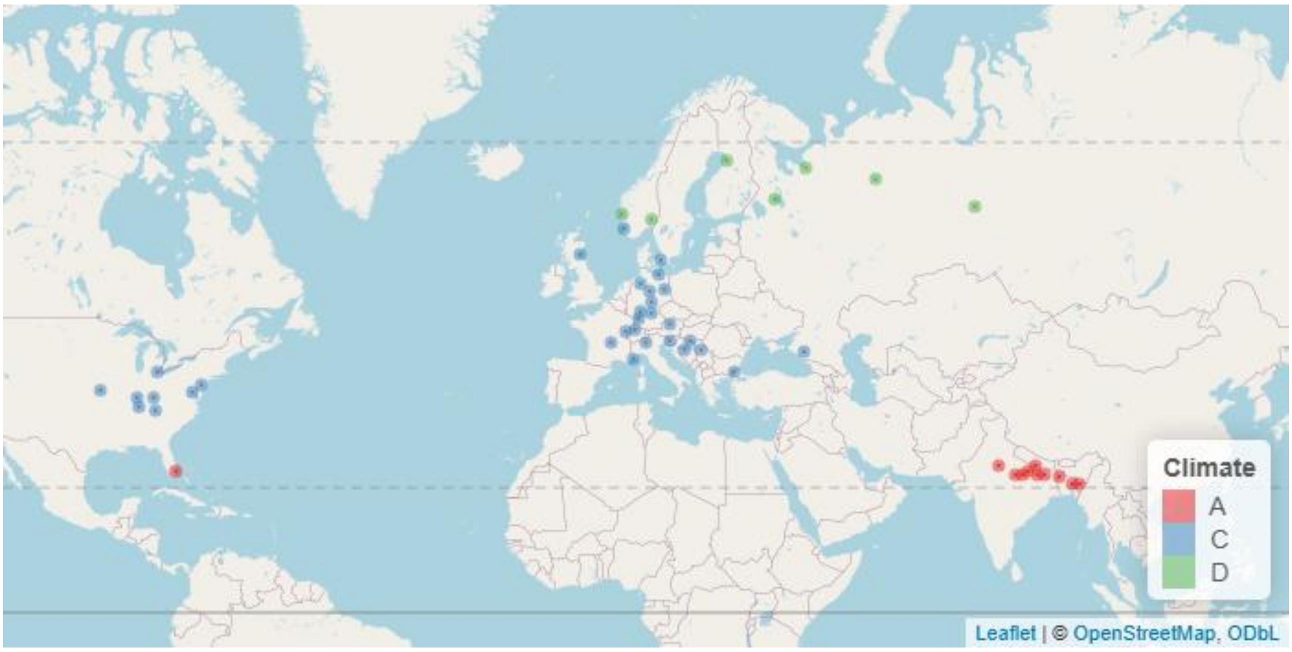

Figure 8. Map of cities and their Köppen-Greiger climate in 1980 - 2012.

In 1980-2012, all cities previously hit by the cataclysm became hotter by 2 Celsius degrees in average. The month that experienced the higher temperature change is February, with its maximum difference equal to 7 degrees.

7 Evaluations

7.1 Limitations of the study and possible extensions

The project is not exempt from limitations. On a methodological point of view, even if the results of the algorithms are satisfactory, using only two can be restricting. A possible solution consists of the implementation of more complex climate models, but it may be expensive in terms of time and computation.

Another difficulty is the dataset granularity: it contains only monthly temperature, does not have any data about precipitations, and 1816 data are little. It may be interesting to make a comparison not just on monthly data but also on daily data, in such a way that considers extreme temperatures and their range. Another aspect that must be taken into account is the expansion of major cities in recent years. The pollution and artificial surfaces (like tarmac) creates 'heat islands', especially in metropolitan areas [14]. This and climate change leads to higher temperatures during the period 1980-2012 and consequently may overestimate the difference with 1816. The project excludes climate B cities due to the lack of data; with another dataset, they can be considered to explore the change in rainfalls too.

The correlation between variables is not managed in this study, since distance metrics such as Manhattan and Euclidean are used. To overcome this issue, Mahalanobis distance can be applied, after transforming all attributes in a normal distribution, but it can be onerous and requires lots of tests for each variable, among which Shapiro-Wilk and Normal Q-Q plot [12].

Even though K-NN classifies records really well, it takes with it some issues: the first is the computational time, the second is its variability across different train and test configurations. Sensitivity of climate C and E varies significantly from the first hold out to the third hold out, as shown in Figure 5, with a range in [0.7; 0.925] of the former, with a relative variability of 24.32%, and [0.783; 0.917] of the latter, with a relative variability of 14.61%. However, sensitivity remains high, and the model is capable of discerning most of the records: this fluctuation is acceptable and could have been avoidable with better distributed data points.

7.2 Conclusions

The project successfully demonstrates that the impact of 1816 climate event made 49 cities shift to a colder climate class, as compared to 1980-2012, which summarizes the effect of the volcanic winter on yearly temperatures.

The methodological approach allows results to be interpretable, thanks to classification that makes the evolution understandable by non-experts, and robust. This includes the transformation of variables into white noises, a geographical grid to manage spatial density and spatial iterated hold-out based on Lambert Azimuthal Equal Earth projection. The choice of algorithms and their distance metrics enhances the analysis' robustness, since it excludes models that require independent records, normally distributed variables, and distance measures like Mahalanobis that rely on a normal distribution.

K-Nearest Neighbour with Manhattan distance and k = 25 stands out with its performance: with a sensitivity over 0.82 and a specificity over 0.95, it can achieve an overall classification accuracy close to 90%, both calculated as the weighted mean across all iterated hold-out runs and for each Köppen-Geiger climate class.

8 Sources

8.1 Bibliography and web sources

1. Massachusetts Historical Society, "1815: The Year Without a Summer," 2016. https://www.masshist.org/beehiveblog/2016/11/1815-the-year-without-a-summer/.
2. Britannica, "Köppen Climate Classification." https://www.britannica.com/science/Köppen-climate-classification.
3. C. Beck, N. E. Zimmermann, T. R. McVicar, N. Vergopolan, R. Rojas, and K. J. Feeley, "Present and Future Köppen-Geiger Climate Classification Maps at 1-km Resolution," *Scientific Data*, vol. 10, p. 215, 2023. https://doi.org/10.1038/s41597-023-02549-6.
4. P. R. Peres-Neto, A. Z. G. Bastos, et al., "An Efficient and Interpretable Classification Approach for Large-Scale Remote Sensing Data," *Scientific Data*, vol. 10, p. 2549, 2023. https://www.nature.com/articles/s41597-023-02549-6.
5. Vienna University of Economics and Business, "Köppen-Geiger Climate Classification Shifts." https://koeppen-geiger.vu-wien.ac.at/shifts.htm.

6. Berkeley Earth, "Climate Change: Earth Surface Temperature Data." https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByCountry.csv.
7. Weather and Climate Info, "Historical Temperatures for European Cities." https://www.weatherandclimate.info/history/.
8. M. Fattore, *Fundamentals of Time Series Analysis, for the Working Data Scientist (DRAFT)*, 2023.
9. NOAA, "Oceanic Niño Index (ONI)." https://ggweather.com/enso/oni.htm.
10. J. Guelat, "Autocorr." https://rpubs.com/jguelat/autocorr.
11. Wolfram MathWorld, "Lambert Azimuthal Equal-Area Projection." https://mathworld.wolfram.com/LambertAzimuthalEqual-AreaProjection.html.
12. T. Hastie, R. Tibshirani, and J. Friedman, *An Introduction to Statistical Learning*, 2013. https://www.stat.berkeley.edu/~rabbee/s154/ISLR_First_Printing.pdf.
13. B. S. Everitt and T. Hothorn, *An Introduction to Applied Multivariate Analysis with R*, Springer, Berlin, 2011.
14. Wikipedia, "Nearest Centroid Classifier." https://en.wikipedia.org/wiki/Nearest_centroid_classifier
15. U.S. Environmental Protection Agency, "Heat Islands." https://www.epa.gov/heatislands.

8.2 Software and libraries

- Python Geopy library. "Geopy." https://geopy.readthedocs.io/.
- R Core Team. *leaflet: Create Interactive Maps.* https://cran.r-project.org/web/packages/leaflet/index.html
- R Core Team. *purrr: Functional Programming Tools.* https://cran.r-project.org/web/packages/purrr/index.html
- R Core Team. *DMwR: Data Mining with R.* https://cran.r-project.org/web/packages/DMwR/index.html
- R Core Team. *kknn: Weighted k-Nearest Neighbors.* https://cran.r-project.org/web/packages/kknn/index.html
- R Core Team. *blockCV: Spatial Cross-Validation Tools.* https://www.rdocumentation.org/packages/blockCV/versions/3.1-4/topics/cv_spatial.
- ChatGPT and Perplexity were used as a support for coding, finding libraries and sources.

Appendix

- **Partial Autocorrelation Function (PACF):** having a stationary stochastic process Y, it is defined as the partial correlation between Y_t and Y_{t+k}, given the variables Y_{t+1}, …, Y_{t+k-1}, with $k = 2, 3, ...$ and $\pi_0 = 1$ e $\pi_1 = \varrho_1$, net of Yt+1, … Yt+k-1. In formulas:

$$\rho_{Y_t,Y_{t+k}} = \frac{\rho_{t,t+k} - \rho_{t,Z} \cdot R_Z^{-1} \cdot \rho_{t+k,Z}^T}{\sqrt{(1-\rho_{t,Z} \cdot R_Z^{-1} \cdot \rho_{t+k,Z}^T)(1-\rho_{t+k,Z} \cdot R_Z^{-1} \cdot \rho_{t+k,Z}^T)}}$$

Where:

  - $Z = \{Y_{t+1}, \dots, Y_{t+k-1}\}$
  - $\rho_{t,Z}$ vector of correlation between each Yt and Yt+i with i = 1,…,k-1
  - R_Z correlation matrix between the variables in Z

  It removes the 'echo effect' of autocorrelation on the lags.

- **Temporal lag:** having a general time series t, the lag is defined as $\tau = t_2 - t_1$.
- **White noise:** stationary process in covariance; it is a sequence of uncorrelated casual variables with finished constant variance.
- **Variogram:** the variogram is a function of spatial dependency of a variable measured in different spatial point, defined as:

$$\gamma(h) = \frac{1}{2} E\left[\left(Z(x) - Z(x+h)\right)^2\right]$$

  Where Z(x) is the value of the variable in x and h the spatial lag between two points.

- **Empirical variogram:** with N(h) the number of points with distance equal to h.

$$\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i,j:|x_i - x_j| \in h} \left(Z(x_i) - Z(x_j)\right)^2$$

- **Spherical model:** where r is the range, nugget is the noise and sill the total variance.

$$\gamma(h) = \begin{cases} 0 & h = 0 \\ nugget + sill \left[\frac{3h}{2r} - \frac{h^3}{2r^3}\right] & 0 < h \le r \\ nugget + sill & h > r \end{cases}$$

- **Lambert Azimuthal Equal Area projection:** map projection having transformation equations, where ϕ_1 is the standard parallel, λ_0 is the central longitude, and

  $x = k' \cos\phi \sin(\lambda - \lambda_0)$

  $y = k'[\cos\phi_1 \sin\phi - \sin\phi_1 \cos\phi \cos(\lambda - \lambda_0)]$

  $$k' = \sqrt{\frac{2}{1 + \sin\phi_1\, sin\phi + \cos\phi_1 \cos\phi \cos(\lambda - \lambda_0)}}$$

- **Manhattan distance:** $d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^1)^1$
- **Euclidean distance:** $d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^2)^{\frac{1}{2}}$
- **True Negative Rate (TNR)** or **Specificity**: $\boldsymbol{TNR = \frac{TN}{TN+FP}}$

  Fraction of negative records predicted correctly by the Classification Model.

- **True Positive Rate (TPR) or Sensitivity, or Recall r:** $\boldsymbol{TPR = \frac{TP}{TP+FN}}$

  Fraction of positive records predicted correctly by the Classification Model. Large recall means very few positive records misclassified as negative class.

- **Normalization:** $x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \cdot 100$