Predicting the Tide of the Pandemic: An In-Depth Analysis of Forecasting Models for COVID-19 in Cantabria

Alberto Lezcano Lastra
Cantabrian Institute of Statistics. Government of Cantabria
lezcano_a@cantabria.es

Gonzalo Llamosas García
Applied Economic Department
University of Malaga
ORCID: 0000-0002-4407-2845
gonzalo.llamosas@uma.es

Alejandro López Cagigas
Cantabrian Institute of Statistics. Government of Cantabria
lopez_ale@cantabria.es

Francisco Javier Parra Rodríguez
Economic Analysis
National University of Distance Education
parra@cee.uned.es

Abstract

Amidst the COVID-19 pandemic, astute public health interventions, including mobility constraints, are paramount. The bedrock of such strategies lies in the precision of forecasting models. Harnessing data from the Cantabrian Health Service, this study critically evaluates and contrasts time series analysis and cutting-edge machine learning techniques in predicting 30-day COVID-19 case trajectories. Additionally, it demystifies the technological scaffolding and methodologies of the Cantabrian Institute of Statistics' web portal for streamlined collation and display of socio-health indicators. The analysis underscores the indispensability and acumen of predictive modeling in steering agile responses to public health crises.

Keywords: forecasting, machine learning, COVID-19, non-parametric models, performance analysis, Cantabria.

MSC Subject classifications: 91B82, 90C30, 62M10, 62M20, 92B20.

1. Introduction

The advent of the COVID-19 pandemic, characterized by the emergence of a novel coronavirus, 2019-nCoV, in Wuhan, China, on December 31, 2019, posed an enormous and multifaceted challenge to global health systems (Gao et al., 2023; Rismanbaf, 2020). This newly identified virus, subsequently designated as SARS-CoV-2 owing to its molecular kinship with SARS-CoV-1, was associated with acute cases of pneumonia and manifested staggeringly high infection and transmission rates relative to other known viral agents (Rismanbaf, 2020). Its prodigious capacity for transmission precipitated its meteoric dissemination worldwide, earning it notoriety as a formidable pathogen. The alarming pace at which this virus spread cast a shadow over global healthcare, making it imperative to devise and implement robust containment strategies.

In scenarios evocative of the COVID-19 pandemic, the success of containment measures is heavily predicated on the accessibility to comprehensive and temporally relevant data on the trajectory of the epidemic (Chakraborty and Ghosh, 2020). Thus, the adoption of scrupulous case monitoring procedures, concomitant with the continuous refinement of predictive accuracy as new data materialize, is vitally imperative (Chakraborty and Ghosh, 2020). Moreover, the availability of precise predictive models can aid policymakers in devising strategies that are more informed and, thus, more likely to curb the transmission of the virus effectively.

An eclectic array of methodologies has been harnessed for prognosticating statistical series associated with infectious diseases (Assad, Cara, and Ortega-Mier, 2023). Noteworthy among these are epidemiological models, with SIR being a quintessential example. SIR models are predicated on the principle that transmission dynamics eclipse demographic dynamics in their temporal evolution, providing valuable insights into the factors that dictate the spread of infections. Additionally, the Autoregressive Integrated Moving Average (ARIMA) model has gained traction in the predictive domain and has been applied to an assortment of ailments, encompassing hemorrhagic fever, brucellosis, influenza, and COVID-19 (Liu et al., 2011; Cao et al., 2020; He and Tao, 2018; Ceylan, 2020). The flexibility and versatility of ARIMA make it a powerful tool in time series analysis, particularly in epidemiological studies. Moreover, local regression merits mention, amalgamating the linearity and simplicity of the least squares regression with the adaptive capacity of nonlinear regression. Local regression techniques, by capitalizing on both linear and nonlinear aspects, enable more nuanced modeling of complex relationships within data.

Of late, machine learning-driven models have gained commendable recognition in the academic realm, attributable to their proficient acumen in forecasting COVID-19 cases (Kwekha-Rashid, Abduljabbar, and Alhayani, 2021; Sujath, Chatterjee, and Hassanien, 2020). These models typically comprise a learning phase (training), during which the algorithm adapts to the characteristics of the data, and an evaluation phase (testing), which assesses the model's predictive performance on unseen data. Neural networks, random forests, and the K-nearest neighbors algorithm are among the most widely employed machine learning techniques (Kwekha-Rashid, Abduljabbar, and Alhayani, 2021). Their capacity for capturing intricate patterns and relationships in data has made them invaluable in tackling complex predictive challenges.

In this scholarly endeavor, using daily PCR positive case data procured from the Cantabrian Health Service, we undertake a meticulous comparative appraisal of the aforementioned modeling techniques over a 30-day interval, with an emphasis on forecasting active cases. We utilize a diverse set of performance metrics to gauge the efficacy of these models. Through this treatise, we seek to enrich the scholarly corpus in this domain, provide insights into the comparative advantages and limitations

of various modeling techniques, and offer valuable perspectives that may guide future research and inform policy-making aimed at containing the spread of infectious diseases like COVID-19.

2. Methodology

2.1. Data

To conduct this comparative analysis of the performance of prediction models, data were sourced from the platform of the Cantabrian Health Service. This repository is diligently updated on a daily basis and encompasses a wide array of variables, which include the number of daily cases, the aggregated tally of cases over 7 and 14 days, the count of patients currently hospitalized, and the overall sum of hospital admissions within the region. The time frame for data collection commenced on the day when the first positive case of SARS-CoV-2 was reported in Cantabria, and stretches up to the present time. This ensures a comprehensive dataset that is both extensive in its coverage and timely in reflecting the ongoing developments.

2.2. Models

In this section, we delve into a plethora of modeling methodologies that have been harnessed for the purpose of forecasting and analyzing epidemiological data. Among the traditional statistical models, the Autoregressive Integrated Moving Average (ARIMA) stands as a paragon for time series analysis and forecasting. Additionally, the scope of our exploration extends to machine learning models which have gained momentum in recent years for their prowess in modeling complex and non-linear relationships within data. Neural networks and random forests exemplify the kind of machine learning models considered in this study. As we navigate through each model, we will explicate their underpinnings, applications, and implications in the context of COVID-19 data analysis.

2.2.1. ARIMA Model

The Autoregressive Integrated Moving Average (ARIMA) model stands as a staple in time series analysis, enjoying widespread application in various fields (Aditya Satrio et al., 2021; Roy, Bhunia, and Shit, 2021). Essentially, the ARIMA model is a composite of three components: Autoregressive (AR), Integrated (I), and Moving Average (MA).

The Autoregressive (AR) part captures the influence of the preceding observations on the current observation. It assumes that the current value of the series, denoted by Y_t, can be expressed as a linear combination of its past values. For an AR process of order p, the mathematical representation is:

$$Y_t = \delta + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t,$$

where δ is a constant, ϕ_i are the parameters of the model, and ε_t is the error term.

The Moving Average (MA) part, on the other hand, accounts for the dependency between an observation and a residual error from a moving average model applied to lagged observations. For an MA process of order q, the mathematical representation is:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q},$$

where μ is the mean of the series, θ_j are the parameters of the model, and ε_t is the error term.

The Integrated (I) part involves differencing the time series data to render it stationary, that is, data with a constant mean and variance over time. When the series is non-stationary, differencing can be applied d times to achieve stationarity. The difference ΔY_t is calculated as:

$$\Delta Y_t = Y_t - Y_{t-1} = Y_t - LY_t = Y_t',$$

Combining the AR and MA models yields the ARMA(p, q) model:

$$Y_t = \delta + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q},$$

Integrating the difference into the ARMA model culminates in the ARIMA(p, d, q) model, generally represented as:

$$(1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p)\Delta^d Y_t = \delta + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q},$$

Here, the Partial Autocorrelation Function (PACF) can help in determining the AR order, while the autocorrelation function plots aid in ascertaining the MA order.

For the selection of the most fitting ARIMA model, Akaike's Information Criterion (AIC) is frequently employed to assess the performance. The AIC accounts for the goodness of fit along with the simplicity of the model. The formula for AIC is:

$$\text{AIC} = -2\log(L) + 2(p + q + k),$$

where L is the likelihood of the data, p is the order of the autoregressive part, q is the order of the moving average part, and k represents the intercept of the ARIMA model. A lower AIC value suggests a more suitable model relative to others with higher AIC values.

2.2.2. Local regression

Local regression, also known as LOESS (Locally Estimated Scatterplot Smoothing), employs a distinct approach to curve fitting. This method involves computing the fit at each data point X_0 by considering only the neighboring training observations. The algorithm can be outlined in the following steps:

1. Select a subset S of the training data such that it contains the k observations whose X_i values are nearest to X_0. Here, the fraction of the training data included in this subset is $s = \frac{k}{n}$.
2. Assign a weight function $K_{i0} = K(X_i, X_0)$ to each point in the subset. The weight function decreases as the distance between X_i and X_0 increases, such that the point farthest from X_0 has zero weight and the one nearest has the highest weight. All points outside this subset are assigned a weight of zero.
3. Fit a weighted least-squares regression of Y_i on X_i using the weights assigned in the previous step. Obtain estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the weighted sum of squared residuals:

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 X_i)^2$$

4. The fitted value at X_0 is computed using the estimated coefficients:

$$\hat{f}(X_0) = \hat{\beta}_0 + \hat{\beta}_1 X_0$$

Local regression can be generalized to accommodate multiple predictors. However, this method becomes inefficient for datasets with more than 3 or 4 predictors, mainly due to the scarcity of observations in the vicinity of X_0.

One important aspect to be aware of regarding local regression is the relatively low interpretability of the results, as the method focuses on fitting the data locally rather than providing an explicit functional form that can be easily interpreted.

2.2.3. Neural network

A neural network is inspired by the biological neural networks that constitute animal brains. It consists of a large number of artificial neurons, known as processing units, which are organized into layers. Each unit within a layer is connected to every unit in the preceding layer through synaptic weights. These weights are pivotal as they encode the knowledge of the network, and can be adjusted during the learning phase.

Data is fed into the network through the input layer and propagates through the subsequent layers until it reaches the output layer. In its standard operation, particularly when employed as a classifier, there is no feedback between layers – this is why this class of neural networks is referred to as feed-forward neural networks.

Feed-forward neural networks are predominantly used for supervised learning tasks, where the goal is to learn a mapping from inputs to outputs. They are particularly effective for datasets where the instances are independent and identically distributed, meaning there is no sequential or temporal dependency between them. More formally, feed-forward neural networks approximate a function f such that $f(x) \approx y$ for example pairs (x, y).

In contrast, recurrent neural networks (RNNs) are designed for sequences of data where the order is important. They achieve this by maintaining a state that can capture information about previous time steps. Mathematically, RNNs approximate a function g on a sequence of inputs $X_k = \{x_1, \ldots, x_k\}$ such that $g(X_k) \approx y_k$ for sequences (X_n, Y_n), where $1 \leq k \leq n$.

In order to execute the neural network within the R Studio environment, the model employs Keras, a versatile and high-level neural networks library written in Python. Keras is renowned for its user-friendliness, modularity, and extensibility, which makes it particularly suitable for rapid prototyping of deep learning models. Alongside Keras, the model integrates TensorFlow, an open-source machine learning library developed by Google. TensorFlow is employed as the backend engine for the neural network, enabling high-performance numerical computations and providing a sophisticated infrastructure essential for training complex models.

By combining Keras's high-level interface with TensorFlow's efficient computation capabilities, the implementation delivers an optimized neural network architecture for the comparative analysis of models. This synergistic utilization of Keras and TensorFlow allows for a streamlined workflow, making it both efficient and accessible for experimenting with neural networks within R Studio.

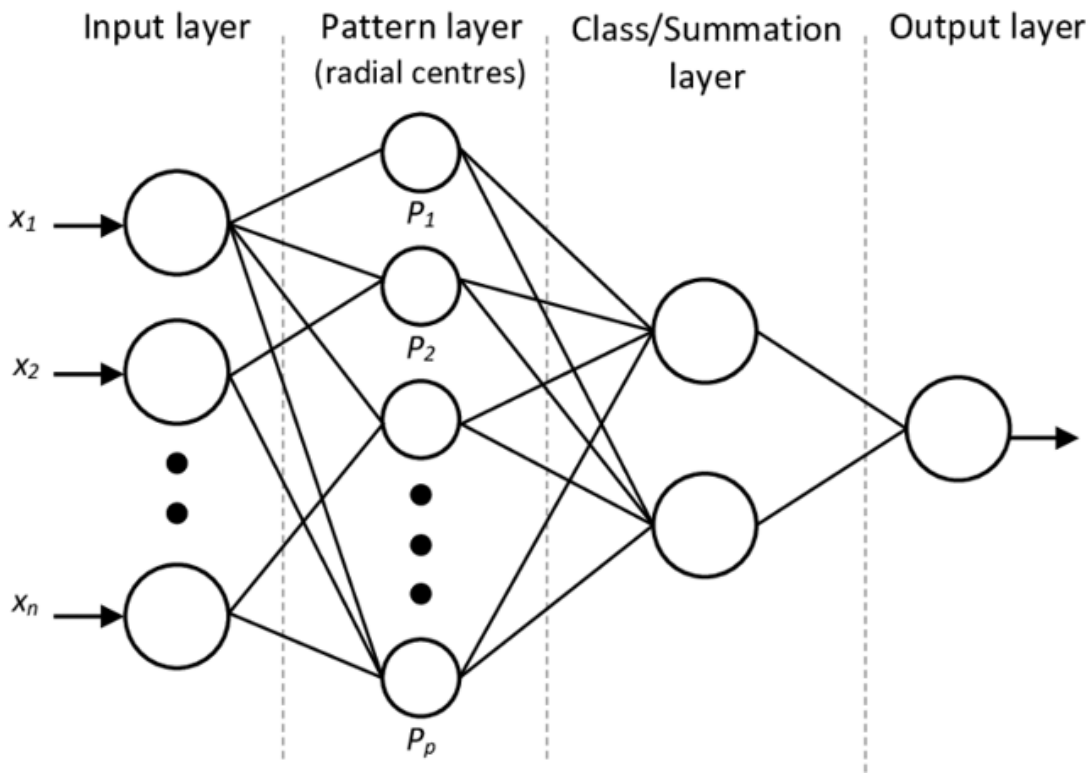

Fig. 1: Graphic Illustration of an Artificial Neural Network

2.2.4. Random forest

The Random Forest algorithm, heralded as a linchpin in supervised machine learning, boasts a compelling trifecta of accuracy, simplicity, and versatility. Its illustrious repute emanates from its dual aptitude in tackling both classification and regression conundrums, compounded by its intrinsically non-linear character, which lends it exceptional adaptability across diverse datasets and scenarios.

Coined as the "random decision forest" by Kam Ho, 1995, this algorithm was a tour de force, deriving predictive acumen through an ingenious amalgamation of randomized datasets. This ensemble algorithm derives its nomenclature, "forest", from its intricate architecture, comprising an amalgamation of decision trees. Rather than relying on the singular output of an isolated decision tree, Random Forests culminate data across an ensemble of decision trees to augment the precision of predictions. Its capacity to inject randomness into model construction by optimizing features within a random subset of data bestows an additional layer of robustness.

At its core, the Random Forest training algorithm deploys the celebrated bootstrap aggregation, or bagging, technique. Given a training set $X = \{x_1, \ldots, x_n\}$ with corresponding responses $Y = \{y_1, \ldots, y_n\}$, bagging iteratively selects random samples with replacement (B times) from the training set and fits trees to these sub-samples. Concretely, for $b = 1$ to B:

- Randomly sample, with replacement, n training instances from X, Y, denoted as X_b, Y_b.

- Train either a classification or regression tree f_b on X_b, Y_b.

Subsequent to the training phase, predictions for unseen instances x' are ascertained by averaging the predictions from all the individual regression trees, or by majority voting in the case of classification trees:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

This bootstrapping protocol bolsters model performance by attenuating variance sans amplifying bias. The predictions from an individual tree may be highly susceptible to noise, but the ensemble mean proves resilient, provided the trees are not correlated.

Furthermore, Random Forests provide an estimate of predictive uncertainty, assessed via the standard deviation of predictions across all the individual regression trees for x':

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B} (f_b(x') - \hat{f})^2}{B - 1}}$$

The number of trees, B, constitutes a hyperparameter, with values spanning from a few hundred to several thousand, contingent upon the dimensions and characteristics of the training dataset. Cross-validation, or examining the out-of-bag error (an average prediction error on each training sample X_i, utilizing the bootstrap sample), can be instrumental in identifying an optimal B. It is noteworthy that both training and test error tend to plateau beyond a certain number of trees.

In sum, Random Forests embody a robust ensemble method that capitalizes on decorrelated trees, spawned through bootstrapped samples, to engender predictions with diminished variance and entrenched reliability, which is quintessential for intricate and high-stakes applications.

2.2.5. K-Nearest Neighbors

The K-Nearest Neighbors algorithm (K-NN), initially proffered by Evelyn Fix and Joseph Hodges in 1951 (Silverman and Jones, 1989) and later augmented by Cover and Hart, 1967, epitomizes a non-parametric modus operandi for classification and regression. Utilizing this algorithm, input data encompasses an array of training instances predicated on the K-NN methodology applied to a dataset. The nature of the outputs is contingent upon whether K-NN is employed for classification or regression.

In the context of K-NN classification, the output denotes class membership. An object's classification is ascertained via a plurality voting mechanism amongst its neighbors, with the object being allocated to the class that is most prevalent among its K nearest neighbors, where K is a positive integer, typically diminutive. In the instance where K is equal to 1, the object is unequivocally allocated to the class of its single nearest neighbor.

Conversely, within K-NN regression, the output represents the value of a particular attribute for the object. This value is computed as the mean of the values attributed to its K nearest neighbors.

K-NN is thus categorized as a type of classifier wherein the function is approximated locally, and computation is deferred until classification is performed. As the algorithm is reliant on distance for classification, normalization of the training dataset is instrumental in significantly bolstering the

accuracy, particularly when the features encompass heterogeneous physical units or are disparate in scale.

In both classification and regression, it is efficacious to ascribe weights to the contributions of neighbors, thereby enabling closer neighbors to exert greater influence on the average than those more remote. For instance, a prevalent weighting scheme might involve conferring upon each neighbor a weight of $\frac{1}{d}$, where d signifies the distance to the neighbor in question.

The neighbors are derived from a corpus of objects for which the class (in K-NN classification) or the object property value (in K-NN regression) is already known. This corpus can be conceptualized as the training set for the algorithm, albeit there is no requirement for an explicit training phase. A notable characteristic of the K-NN algorithm is its sensitivity to the local data structure.

The statistical underpinnings of the K-NN methodology can be elucidated as follows: Suppose we have the following pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, taking values in $\mathbb{R}^d \times \{1, 2\}$, where Y is the class label of X, such that $X/Y \sim P_r$ for $r = 1, 2$ (probability distributions P_r). Given a norm $\|\cdot\|$ on $\mathbb{R}^d$ and $X \in \mathbb{R}^d$, let $(X_{(1)}, Y_{(1)}), \ldots, (X_{(n)}, Y_{(n)})$ be an ordering of the training data such that $\|X_{(1)} - x\| \leq \ldots \leq \|X_{(n)} - x\|$.

2.3. Technological Stack for Cantabria's COVID-19 Data Application

As of the current landscape, there exist three dominant frameworks in the realm of frontend web development: Angular, 2023, React, 2023, vue, 2023. A meticulous evaluation of these frameworks indicates a parity concerning the functionalities and design patterns. However, discrepancies are evident in terms of architectural complexities and learning curves. Angular, though robust, is relatively intricate. React is renowned and in vogue Sala, 2021, albeit with a steep learning curve. In contrast, Vue is more lightweight and intuitive, making it particularly suitable for compact or solitary development teams.

Upon assessing the contemporary trends and trajectories of frontend JavaScript frameworks, Vue was selected for the frontend development due to its potential and accessibility for newcomers.

GitHub

Github, 2023 serves as a cloud-based repository for version control using the Git system. It encompasses an array of tools that foster collaborative programming and offers a suite of functionalities tailored for project management. Its epitome lies in augmenting collaboration and productivity via a Kanban-style project manager.

Vuetify

Vuetify, a UI library, was utilized for crafting the user interfaces. This library furnishes an extensive repertoire of reusable components, facilitating the creation of aesthetically appealing and visually coherent applications adhering to Google's Material Design principles *Material Design* 2023.

Chart.js

Chart.js, 2023 was employed for chart development. As a popular open-source library, it has garnered substantial acclaim with approximately 55,000 stars on GitHub. Chart.js strikes a balance between simplicity and a diverse set of functionalities, offering eight chart types including lines, bars, radar, donuts, pies, polar areas, bubbles, and scatter plots.

Python

Python, 2023 is a dynamic, interpreted, and cross-platform programming language renowned for its agility, simplicity, and versatility. Conceived in the late 1980s, Python is backed by the Python Software Foundation, a non-profit entity committed to fostering the Python programming language and nurturing a global community of Python developers. Python, alongside R, is predominantly utilized in data processing and holds the distinction of being one of the most sought-after programming languages worldwide.

Python's versatility enables the development of data processing modules and packages, integration with other information systems via microservices, and publication in data dissemination portals. The maturity of Python's web frameworks also allows for the creation of auxiliary applications for automating tasks.

Firebase

Firebase, 2023 is a cloud-based platform catering to the development of web and mobile applications. It is particularly lauded for the Firebase Realtime Database, a NoSQL cloud database that permits storing and synchronizing data in JSON format across users in real-time.

Posit

The domain of statistical analysis has witnessed rapid advancements, and the software tools have evolved commensurately with the progression in computational capabilities. Posit, 2023 is a cloud-based solution that offers seamless online multiplatform collaboration in real-time. In the context of public administration, it circumvents issues pertaining to version and package incompatibility, serving as a unified Cloud offering for R Studio. However, a limitation is the constraint in installing Python extensions, such as Keras, due to size considerations and cloud storage limitations even within the Premium tier.

3. Results

In contrast to extant literature on COVID-19, this study presents a comprehensive analysis of five distinct estimation and classification methodologies for projecting active PCR cases. A rigorous two-phase validation framework is employed. The initial phase encompasses training, wherein machine learning algorithms glean insights from the historical data of statistical series. Subsequently, the second phase evaluates the efficacy of these models during the forecasting stage.

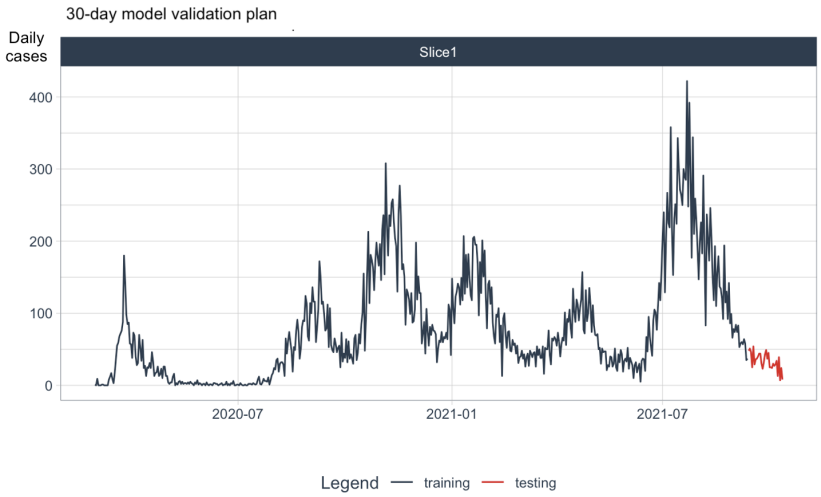

Fig. 2: 30-Day Model Validation Plan

The evaluation strategy for models is systematically crafted through a coherent approach, premised on the validation framework. This study juxtaposes the prognostications from ARIMA, local regression (LM), random forest (Random Forest), neural network (Keras), and K-Nearest Neighbors (K-NN) models against the empirically observed values from the daily PCR case series for Cantabria.

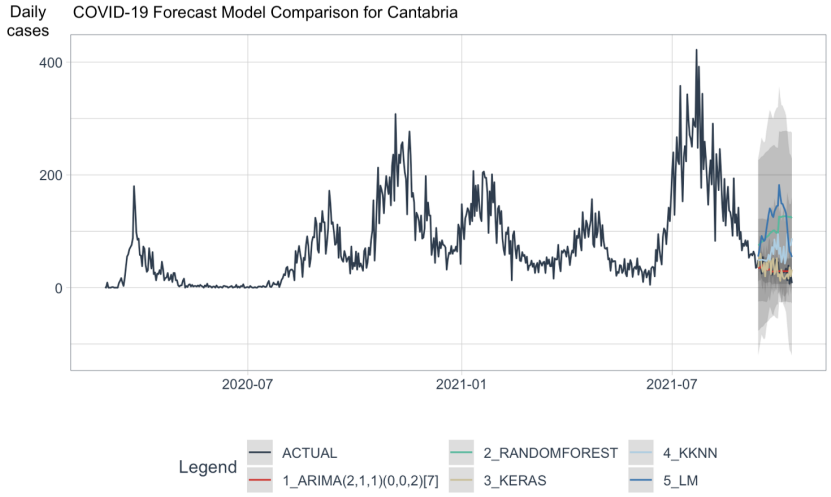

Fig. 3: Comparative Forecast Models for Cantabria

As delineated in Figure 3, the ARIMA, neural network, and K-NN models conspicuously approximate the actual data with a higher degree of fidelity. Conversely, local regression and random forest models exhibit suboptimal performance, deviating considerably from the empirical data.

The supplementary section of this article furnishes granular forecasts from each model. A meticulous juxtaposition of the model projections with the observed data enables an assessment of the relative efficacy of each model in forecasting COVID-19 daily cases. Notably, the ARIMA and neural network models manifest superior predictive accuracy, as evidenced through an analysis of the detailed forecasts provided in the appendix.

This observation is further supported by a careful examination of the performance metrics listed in Table 1. The metrics reveal an inverse relationship between model fit and their corresponding values. For example, the Mean Absolute Percentage Error (MAPE) highlights the superiority of the ARIMA model with a score of 43.48, in contrast to local regression, which evidently emerges as the least effective model based on the aggregate performance indicators.

Tab. 1: Comparative Analysis of Performance Metrics Across Models.

	Performance Factor				
Model	**MAE**	**MAPE**	**MASE**	**SMAPE**	**RMSE**
ARIMA	9.53	43.48	0.88	31.62	11.49
Random Forest	71.32	325.14	6.59	102.43	76.58
Neural Network	14.33	61.46	1.32	47.71	15.95
K-NN	29.77	158.46	2.75	61.71	36.99
Local regression	80.51	304.25	7.44	107.30	88.33

Note: MAE (Mean Absolute Error) measures the average magnitude of errors in a set of predictions, without considering their direction. *MAPE* (Mean Absolute Percentage Error) expresses the forecast errors as a percentage. textitMASE (Mean Absolute Scaled Error) measures the accuracy of forecasts, scaled based on the training data's error. *SMAPE* (Symmetric Mean Absolute Percentage Error) is an improved version of MAPE that accounts for over-forecasts and under-forecasts equally. *RMSE* (Root Mean Square Error) measures the square root of the average squared differences between predictions and observed values.

A holistic analysis of the performance metrics yields a conclusive ranking of the models predicated on their efficacy in forecasting daily COVID-19 cases in Cantabria: ARIMA ranks first, followed by Neural Network, K-NN, Random Forest, and Local Regression.

These results thus contribute novel insights into the relative efficacy of diverse forecasting methodologies, serving as an invaluable resource for policy makers and stakeholders in strategizing interventions and resource allocation.

4. Conclusions

This study undertakes a robust comparison of time-series forecasting models to analyze the trajectory of epidemiological outbreaks amid the COVID-19 pandemic. The paper introduces an evaluative framework, comprising a spectrum of performance metrics, through which a collection of artificial intelligence models is appraised alongside conventional statistical models, including ARIMA.

Efficacy of the models is substantiated through performance parameters as well as 30-day forecasts juxtaposed with actual observations, which delineate the superiority of ARIMA and neural network methodologies in predicting daily PCR data for Cantabria. The commendable performance of these models is consistent with the empirical insights accrued by the Cantabrian Institute of Statistics (*Instituto Cántabro de Estadística*) staff in their persistent endeavor to evaluate forecasting models

during the COVID-19 pandemic. In particular, neural networks exhibited exemplary adaptation to fluctuations in daily cases, as recorded by the Cantabrian Health Service and later subjected to scrutiny at the Institute. The forecasts, with a 14-day horizon, demonstrated artificial intelligence models' slight edge in terms of precision.

Additionally, the technological facet of this study deserves special mention. The process of data collection and analysis unraveled the prodigious capabilities of cloud computing tools. The availability of online repositories such as GitHub, cloud-based development environments like Firebase, and intuitive programming languages including Python, coupled with statistical programming applications such as R Studio Cloud (Posit), have significantly streamlined the task of processing and analyzing COVID-19 data. Although the infancy of the pandemic was marked by rudimentary data infrastructure, the tenacity in information gathering led to the expeditious development of a COVID-19 interface, which furnished indispensable insights into the pandemic's progression to the citizens of Cantabria.

In summary, this manuscript encapsulates a vital component of the relentless efforts exerted by the Cantabrian Institute of Statistics in data-driven pandemic management. The journey was replete with learning and innovations, encompassing data acquisition, processing, modeling, and real-time analysis, which were predominantly facilitated by the versatility of Cloud Computing in handling statistical information.

Acknowledgments

The authors extend their heartfelt gratitude to the Cantabrian Institute of Statistics and the Department of Health of the Government of Cantabria for their essential collaboration. This partnership was crucial in the development of a project focusing on the monitoring of epidemiological data pertaining to COVID-19, utilizing data from the Cantabrian Health Service. This joint effort provided a solid foundation for the research presented in this study. We also wish to acknowledge and thank all those who contributed, directly or indirectly, to the successful completion of this research.

About the authors

Alberto Lezcano Lastra Computer Systems Analyst at the General Directorate of Organization and Technology, Government of Cantabria. With a Statistics degree from the Complutense University of Madrid, his career is centered around data management, specializing in processing and analytical systems. Alberto's expertise contributes to efficient data handling and decision-making within the government.

Gonzalo Llamosas García As an Assistant Professor of Economics at the University of Málaga, his forte is in fiscal systems and public economics. He obtained his Ph.D. in Economics through a collaborative program encompassing the University of Cantabria, Oviedo, and the Basque Country. Before delving into the academic realm, he made notable contributions to research projects such as H2020 Citadel at the University of Cantabria, and later employed his skills as a statistician at the Cantabrian Institute of Statistics, a division of the Government of Cantabria. His research intricately weaves together behavioral finance, economic policy, and statistics, and has been augmented by international experiences, including a stint as a Visiting Research Fellow at Bangor University.

Alejandro López Cagigas Native of Rubayo (Cantabria), holds a Computer Engineering degree from the University of Cantabria, attained in 2016. He entered the civil service in 2018 and now thrives as a member of the Assistant Technical Corps of Computer Science at the General Directorate of Organization and Technology, Government of Cantabria. His role is pivotal in maintaining and developing computer systems.

Francisco Javier Parra Rodríguez Professor in the Department of Applied Economics at the University of Cantabria and the Head of Economic and Sociodemographic Statistics at Cantabrian Institute of Statistics. He holds a degree in Economics from Universidad Autónoma de Madrid and a Ph.D. from UNED, earning the Extraordinary Doctorate Award in 1998/1999. He began his career in statistical analyses at Junta de Castilla y León and now leads various research publications at Government of Cantabria. He lectures in Econometrics and co-directs the Master of Big Data and Data Science at UNED, actively participating in R+D+i projects with multiple institutions.

References

Aditya Satrio, CB., W. Darmawan, B. Unrica Nadia, and N. Hanafiah (2021). «Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET». In: *Procedia Computer Science* 179, pp. 524–532. DOI: https://doi.org/10.1016/j.procs.2021.01.036.

Angular (2023). *Angular: One framework: mobile and desktop.* URL: https://angular.io/.

Assad, D.B.N., J. Cara, and M. Ortega-Mier (2023). «Comparing Short-Term Univariate and Multivariate Time-Series Forecasting Models in Infectious Disease Outbreak». In: *Bulletin of Mathematical Biology* 85.9. DOI: 10.1007/s11538-022-01112-5. URL: https://doi.org/10.1007/s11538-022-01112-5.

Cao, L., H. Liu, J. Li, X. Yin, Y. Duan, and J. Wang (2020). «Relationship of meteorological factors and human brucellosis in Hebei Province, China». In: *Sci Total Environ.*

Ceylan, Z. (2020). «Estimation of COVID-19 prevalence in Italy, Spain, and France». In: *Sci Total Environ.* DOI: 10.1016/j.ijid.2018.07.003.

Chakraborty, T. and I. Ghosh (2020). «Real-time forecasts and risk assessment of novel coronavirus (COVID-19) cases: a data-driven analysis». In: *Chaos Solitons Fractals*. DOI: 10.1016/j.chaos.2020.109850.

Chart.js (2023). *Chart.js: Simple yet flexible JavaScript charting library for the modern web*. URL: https://www.chartjs.org/.

Cover, Thomas M. and Peter E. Hart (1967). «Nearest neighbor pattern classification». In: *IEEE Transactions on Information Theory* 1, pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

Firebase (2023). *Firebase Products*. URL: https://firebase.google.com/products-build.

Gao, Xiaoyang, Yeting Xia, Xiaofang Liu, Yinlan Xu, Pengyang Lu, Zhipeng Dong, Jing Liu, and Gaofeng Liang (2023). «A perspective on SARS-CoV-2 virus-like particles vaccines». In: *International Immunopharmacology* 115, p. 109650. ISSN: 1567-5769. DOI: 10.1016/j.intimp.2022.109650. URL: https://www.sciencedirect.com/science/article/pii/S1567576922011353.

Github (2023). *Github: let's built from here*. URL: https://github.com/.

He, Z. and H. Tao (2018). «Epidemiology and ARIMA model of positive-rate of influenza viruses among children in Wuhan, China. A nine-year retrospective study». In: *Int J Infect Dis*. DOI: 10.1016/j.ijid.2018.07.003.

Kam Ho, T. (1995). «Random decision forests». In: *Proceeding of 3rd International Conference on Document Analysis and Recognition*. Vol. 1, pp. 278–282. DOI: 10.1109/ICDAR.1995.598994.

Kwekha-Rashid, A.S., H.N. Abduljabbar, and B. Alhayani (2021). «Coronavirus disease (COVID-19) cases analysis using machine-learning applications». In: *Appl Nanosci*. DOI: https://doi.org/10.1007/s13204-021-01868-7.

Liu, Q., X. Liu, B. Jiang, and W. Yang (2011). «Forecasting incidence of hemorrhagic fever with renal syndrome in China using ARIMA model». In: *BMC Infect Dis* 11.1. DOI: 10.1186/1471-2334-11-218.

Material Design (2023). URL: https://material.io/design/.

Posit (2023). *Posit: a Cloud platform for R Studio*. URL: https://login.rstudio.cloud/.

Python (2023). *Python, a programming language that lets you work quickly and integrate systems more effectively*. URL: https://www.python.org/.

React (2023). *React: a javascript library for building user interfaces*. URL: https://reactjs.org/.

Rismanbaf, A. (2020). «Potential treatments for COVID-19: a narrative literature review». In: *Arch Acad Emerg Med* 8.1.

Roy, S., G.S. Bhunia, and P.K. Shit (2021). «Spatial prediction of COVID-19 epidemic using ARIMA techniques in India». In: *Model. Earth Syst. Environ.* 7, pp. 1385–1391. DOI: https://doi.org/10.1007/s40808-020-00890-y.

Sala, X. (2021). *Angular vs react vs vue demanda de empleo (2015-2018)*. URL: https://www.jobfluent.com/es/blog/angular-vs-react-una-evolucion-de-la-demanda-de-empleo.

Silverman, B. W. and M. C. Jones (1989). «E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)». In: *International Statistical Review / Revue Internationale de Statistique* 57.3, pp. 233–238. DOI: 10.2307/1403796.

Sujath, R., JM. Chatterjee, and AE. Hassanien (2020). «A machine learning forecasting model for COVID-19 pandemic in India». In: *Stochastic Environmental Research and Risk Assessment* 34, pp. 959–972. DOI: https://doi.org/10.1007/s00477-020-01827-8.

vue (2023). *Vue: The progressive javascript framework*. URL: https://vuejs.org/.