Universidad Nacional de Luján
Lic. en Sistemas de Información
Bases de Datos Masivas

Predicción de nuevos casos de coronavirus/COVID-19 con Random Forest

Quiroga Mario
Nro. Legajo: 113329
Mail: marioquiro17@gmail.com

1. Introducción

El Covid-19 es una nueva enfermedad infecciosa generada por el virus SARS-CoV2. Esta enfermedad se ha convertido en una pandemia a lo largo del año 2020 [1]. Por lo tanto, conocer la dispersión del virus desde diferentes perspectivas, es un tema importante en muchas áreas de estudio. En este trabajo, se propone ajustar un modelo de predicción, utilizando el algoritmo Random Forest para regresión, con el objetivo de predecir la cantidad de nuevos casos confirmados una semana en adelante para el territorio argentino. Para lograrlo, se realiza el proceso de descubrimiento de conocimiento (Knowledge Discovery in Databases - **KDD**) sobre el Dataset "Covid-19, Casos registrados en la República Argentina"[1], perteneciente a la Dirección Nacional de Epidemiología y Análisis de Situación de Salud, del Ministerio de Salud de la República Argentina. Este Dataset contiene todos los casos de Covid-19 registrados en Argentina.

Utilizando los datos mencionados se generan las variables de predicción (targets), con el recuento de nuevos casos confirmados para cada día, y cuántos nuevos casos habrá dentro de 7, 14 y 30 días. Se ajusta un modelo para cada una de estas cuatro variables, utilizando los mismos datos de entrada para el entrenamiento.

Para las muestras de entrada a los modelos, se calculan las variables predictoras basadas en datos pasados de la cantidad de nuevos casos confirmados.

Por último, se comparan los 4 modelos ajustados utilizando las siguientes métricas de evaluación de modelos para regresión:

- **MSE**: Error cuadrático medio.
- **MAE**: Error absoluto medio.
- **RMSE**: Raíz cuadrada del error cuadrático medio.
- **Max-Error**: Error máximo.
- **R-Score:** (Coeficiente de determinación).

Comparando los cuatro modelos con estas métricas, se concluye cuál es mejor para realizar un buen pronóstico de la dispersión del virus en la Argentina, con los datos creados en la etapa de preprocesamiento.

[1]https://datos.gob.ar/dataset/salud-covid-19-casos-registrados-republica-argentina

2. Metodología

La metodología de este trabajo está guiada por el proceso de descubrimiento de conocimiento (KDD), el cual consiste en las siguientes etapas:

- **Selección**: Se obtienen los datos para realizar el estudio.
- **Preprocesamiento**: Se genera la serie de tiempo para crear un conjunto de datos para aprendizaje supervisado.
- **Transformación**: Se construyen nuevas variables predictoras que ayuden a mejorar la predicción.
- **Data Mining:** Se ajustan los modelos y se buscan los parámetros que ofrezcan los mejores resultados.
- **Evaluación:** Se utilizan las métricas de evaluación para modelos de regresión mencionadas.
- **Conclusión:** Se analizan los resultados concluyendo que tipo de predicciones pueden hacerse con los datos disponibles.

Dado que el proceso KDD es un proceso iterativo e interactivo, será necesario volver a etapas anteriores desde cualquier etapa, modificando los parámetros de los modelos o las variables predictoras utilizadas, con el objetivo de mejorar los resultados en la predicción.

3. Preliminares

Las siguientes subsecciones proveen una introducción a los conceptos abordados en este trabajo.

3.1 Series de tiempo

Como se ha mencionado en la sección 1, los datos que se utilizan en este trabajo son series de tiempo. Una serie de tiempo es un conjunto de observaciones para las cuales se obtienen mediciones en puntos discretos de tiempo [4]. También, puede pensarse a una serie de tiempo como una variable aleatoria indexada por una marca de tiempo (timestamp).

3.2 Árboles de decisión

Un árbol de decisión es un modelo de predicción utilizado tanto para clasificación como para regresión. Se construye un árbol donde en los nodos se toma una decisión sobre las variables predictoras.

En un árbol de decisión, cada nodo interno divide el espacio de instancias en dos o más subespacios de acuerdo con una determinada función discreta de los valores de los atributos de entrada. En el caso más simple y frecuente, cada prueba considera un solo atributo, de modo que el espacio de instancias se particiona de acuerdo con el valor de los atributos. En el caso de atributos numéricos, la condición se refiere a un rango. Cada hoja se asigna a una clase que representa el valor objetivo más apropiado. La siguiente imagen ilustra un ejemplo de árbol de decisión. Las flechas que unen los nodos corresponden a los posibles valores de las variables predictoras.

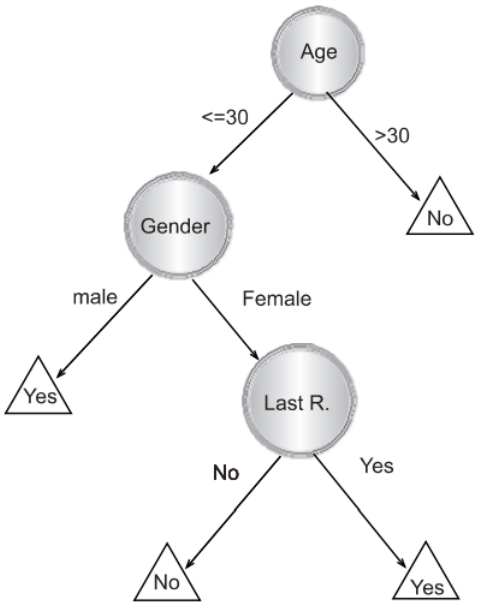

Figura 1. Ejemplo de árbol de decisión con 3 variables predictoras.

Más información sobre cómo funcionan los algoritmos para la construcción del árbol pueden encontrarse en [6].

3.3 Random Forest

Random Forest es un meta-algoritmo de aprendizaje supervisado, el cual consiste en un conjunto de árboles de decisión, donde cada árbol depende de un subconjunto de cada variable aleatoria en el conjunto de entrenamiento. Es decir, se toman muestras aleatorias con reemplazo del conjunto de entrenamiento, creando un subconjunto, y se ajusta cada árbol a cada uno de estos subconjuntos. Luego, se pueden hacer predicciones de la variable objetivo promediando los resultados de las predicciones de los árboles entrenados cuando se realiza una regresión, o por mayoría cuando se realiza una clasificación [5]. Esta técnica se denomina bagging, la cual ayuda a reducir la variabilidad de los datos y el sobreajuste. La siguiente imagen ilustra el proceso de bagging en los árboles de decisión.

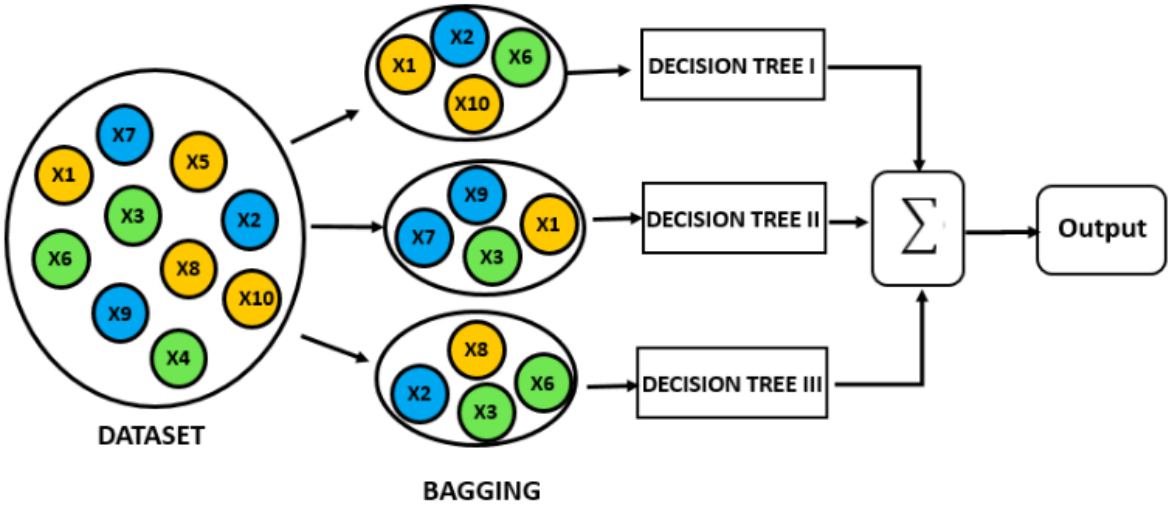

Figura 2. Ejemplo del proceso de bagging.

3.4 Métricas de evaluación para modelos de regresión

Las métricas de evaluación para modelos de regresión, miden el rendimiento de los modelos entrenados. A continuación, se presenta la definición de cada una de las métricas utilizadas en este trabajo.

- **MSE:** El error cuadrático medio se define como:

$$MSE(y,\ y') \ = \ \frac{1}{n\ ejemplos} \sum_{i=0}^{n\ ejemplos\ -\ 1} (y_i - y'_i)^2$$

- y: vector de valores reales de la variable objetivo en el conjunto de prueba.
- y´: vector de valores predichos para el conjunto de prueba

- **MAE:** El error absoluto medio se define como:

$$MAE(y,\ y') \ = \ \frac{1}{n\ ejemplos} \sum_{i=0}^{n\ ejemplos-1} |y_i - y'_i|$$

- **RMSE:** La raíz cuadrada del error cuadrático medio se define como:

$$RMSE(y,\ y') \ = \sqrt{MSE}$$

- **R score:** El coeficiente de determinación se define como:

$$R^2(y,\ y') \ = \ 1 \ - \ \frac{\sum_{i=1}^{n\ ejemplos} (y, y')^2}{\sum_{i=1}^{n\ ejemplos} (y, y'')^2}$$

$$y'' : \frac{1}{n} \sum_{i=1}^{n} y'$$

Se utilizan las implementaciones del módulo metrics de la librería sklearn para el lenguaje de programación python. Pueden encontrarse más detalles en la documentación del módulo en su sitio web[2].

4. Selección

Los datos utilizados para ajustar los modelos se obtienen mediante un archivo CSV, el cual puede ser descargado en el sitio web del Sistema Integrado de Información Sanitaria Argentino[3]. Este archivo contiene todos los casos registrados de Covid-19 en la República Argentina. Su información es actualizada diariamente. A continuación, se listan las columnas del DataSet con su tipo de dato y descripción:

Título de la columna	Tipo de dato	Descripción
id_evento_caso	Número entero (integer)	Número de caso
sexo	Texto (string)	Sexo
edad	Número entero (integer)	Edad
edad_años_meses	Texto (string)	Edad indicada en meses o años
residencia_pais_nombre	Texto (string)	País de residencia
residencia_provincia_nombre	Texto (string)	Provincia de residencia

[2] https://scikit-learn.org/stable/modules/model_evaluation.html
[3] https://sisa.msal.gov.ar/datos/descargas/covid-19/files/Covid19Casos.zip

residencia_departamento_nombre	Texto (string)	Departamento de residencia
carga_provincia_nombre	Texto (string)	Provincia de establecimiento de carga
fecha_inicio_sintomas	Fecha ISO-8601 (date)	Fecha de inicio de síntomas
fecha_apertura	Fecha ISO-8601 (date)	Fecha de apertura del caso
sepi_apertura	Número entero (integer)	Semana Epidemiológica de fecha de apertura
fecha_internacion	Fecha ISO-8601 (date)	Fecha de internación
cuidado_intensivo	Texto (string)	Indicación si estuvo en cuidado intensivo
fecha_cui_intensivo	Fecha ISO-8601 (date)	Fecha de ingreso a cuidado intensivo en el caso de corresponder
fallecido	Texto (string)	Indicación de fallecido
fecha_fallecimiento	Tiempo ISO-8601 (time)	Fecha de fallecimiento en el caso de corresponder
asistencia_respiratoria_mecanica	Texto (string)	Indicación si requirió asistencia respiratoria mecánica
carga_provincia_id	Número entero (integer)	Código de Provincia de carga
origen_financiamiento	Texto (string)	Origen de financiamiento
clasificacion	Texto (string)	Clasificación manual del registro
clasificacion_resumen	Texto (string)	Clasificación del caso
residencia_provincia_id	Número entero (integer)	Código de Provincia de residencia
fecha_diagnostico	Tiempo ISO-8601 (time)	Fecha de diagnóstico
residencia_departamento_id	Número entero (integer)	Código de Departamento de residencia
ultima_actualizacion	Fecha ISO-8601 (date)	Última actualización

Tabla 1. Columnas del DataSet utilizado.

5. Preprocesamiento

Como en el DataSet obtenido, cada instancia es un nuevo caso, el primer paso es generar una serie de tiempo con el recuento diario de casos confirmados registrados. Para contar los casos confirmados se utiliza la variable **clasificacion_resumen,** la cual indica si se ha confirmado el caso mediante una prueba por laboratorio. La fecha utilizada para la variable de tiempo (fecha), es la variable **fecha_diagnostico.** Además, se calculan las otras tres variables objetivos (target), para predecir la cantidad de casos en 7, 14 y 30 días en el futuro.

Las columnas obtenidas son:

- fecha
- nuevos casos confirmados el día de la fecha,
- nuevos casos confirmados dentro de 7 días,
- nuevos casos confirmados dentro de 14 días,
- nuevos casos confirmados dentro de 30 días.

El código en python para lograr esto puede encontrarse en una notebook[4] de preprocesamiento en Google Colaboratory. La figura 3 muestra la forma que tiene el DataFrame una vez generadas las series de tiempo.

	date	target_cases	target_d7_cases	target_d14_cases	target_d30_cases
0	2020-01-06	8	10	1	11
1	2020-01-07	3	1	1	10
2	2020-01-08	3	2	1	18
3	2020-01-12	2	1	1	29
4	2020-01-15	1	2	1	22
...	...	...	...	...	...
397	2021-03-17	7370	6447	15309	22491
398	2021-03-18	7116	10330	12998	14724
399	2021-03-19	7494	11769	10190	9899
400	2021-03-20	5350	8803	11656	18932
401	2021-03-21	2900	5254	8229	5884

402 rows × 5 columns

Figura 3. DataFrame inicial luego de obtener la serie de tiempo y calcular las nuevas variables objetivo.

El siguiente gráfico muestra la serie obtenida en la etapa de preprocesamiento para la variable **nuevos casos confirmados al día de la fecha**. Claramente pueden observarse los tres picos en la serie, donde mayor cantidad de cantidad de casos se registró por día.

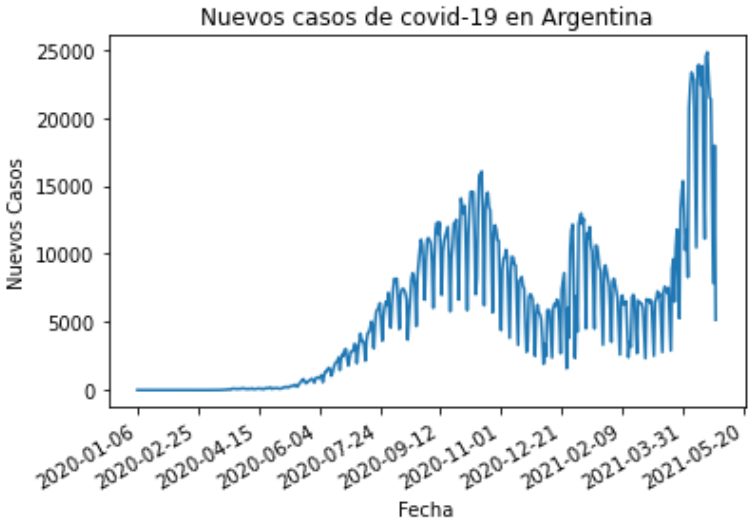

Figura 4. Gráfico cantidad de nuevos casos de covid-19 en la Argentina.

[4] https://colab.research.google.com/drive/1kxAt4hJtlSYnMJ5xlvSwSttpuijq5Txy?usp=sharing

5.1. Datos faltantes

Para algunos días en las series generadas, la cantidad de nuevos casos confirmados es cero. Para evitar que el ajuste se vea afectado por estos valores atípicos, se eliminan todas las filas donde la cantidad de casos confirmados sea igual a cero.

6. Transformación

En esta etapa se crean nuevas variables predictoras, para el conjunto de entrenamiento. Se implementaron funciones basadas en instancias anteriores en la serie para cada instancia de la variable objetivo nuevos casos.

Para obtener nuevas variables predictoras se utilizan los siguientes métodos:

- Lag features: Para cada valor a predecir en la serie de datos, se utiliza como feature la cantidad de casos registrados **n** días atrás de cada valor a predecir[2].
- Media móvil: Para cada valor a predecir se calcula el promedio de los **n** días anteriores [3].
- Desviación estándar móvil: Para cada valor a predecir se calcula el desvío estándar de los **n** días anteriores
- Features basadas en la fecha, tales como día de la semana y día del mes a predecir.
- Máxima cantidad de casos los últimos **n** días.
- Mínima cantidad de casos los últimos **n** días.

Para las variables Media Móvil, Desviación Estándar Móvil, Máxima cantidad de casos y Mínima cantidad de casos, se utilizan los valores **n=7, n=14 y n=30.** En la figura 5 se ilustran los valores de la ventana deslizante para las instancias de tiempo t8, t9, t10 y t11. El tamaño de la ventana es **n=7.** Para cada instancia de tiempo, se calculan los diferentes estadísticos mencionados sobre los valores dentro de la ventana. Para cada instancia de tiempo, puede observarse el movimiento de la ventana en la serie. Para los restantes valores de **n** se utiliza el mismo método pero cambiando el tamaño de ventana.

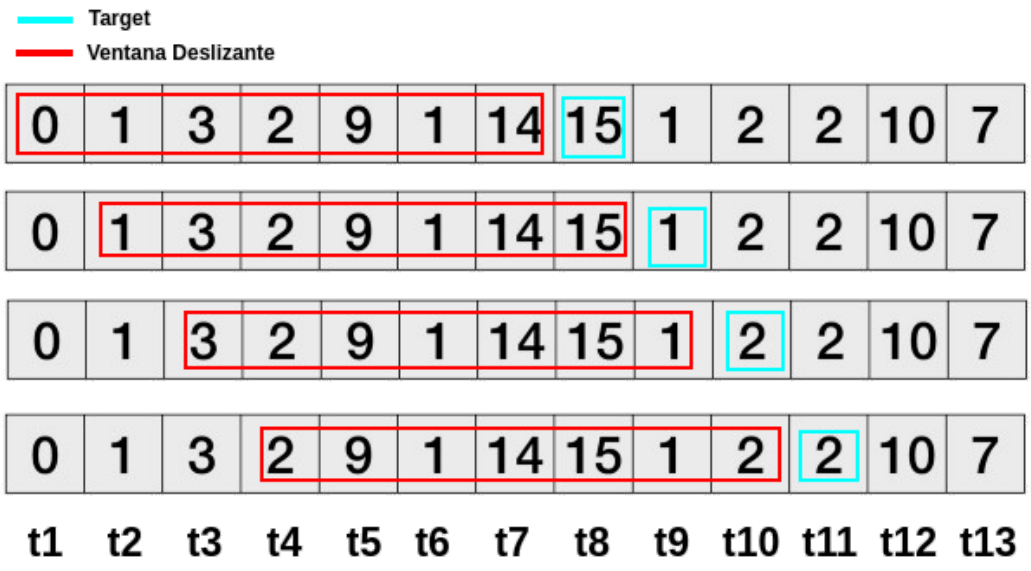

Figura 5. Ventana deslizante para **n=7.**

Para la variable Lag feature, se utiliza **n=1, n=7, n=14 y n=30**. Es decir, la cantidad de casos confirmados 1, 7, 14 y 30 días atrás. La figura 6 muestra (en rojo) los valores tomados para la variable predictora con **n=7,** para las instancias de tiempo **t8, t9, t10** y **t11.**

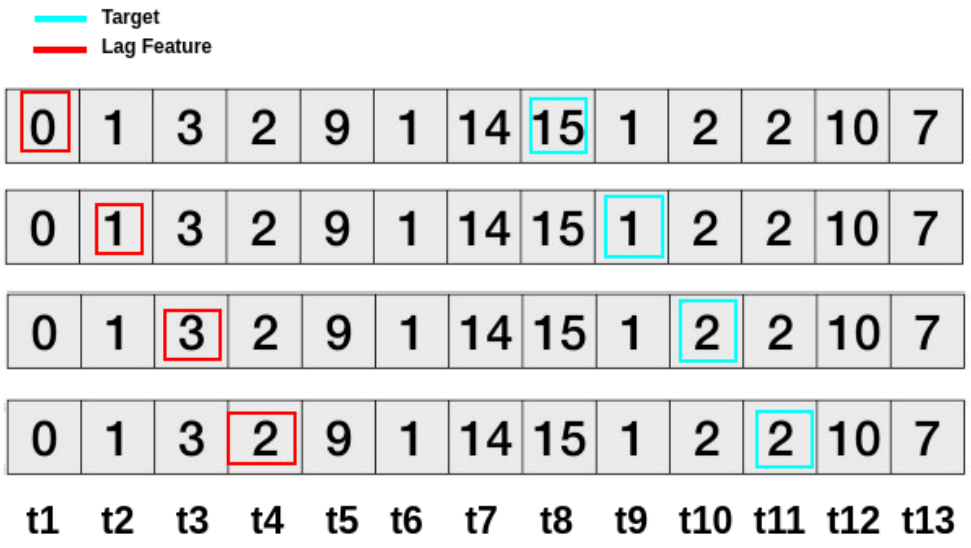

Figura 6. Lag feature para **n=7.**

6.1. Separar las muestras de entrenamiento y de prueba

Para separar las muestras de entrenamiento y de prueba, dejamos como datos de prueba las últimas 7 muestras. Por lo tanto, se ajustan los modelos con datos pasados y se los evalúa con datos futuros. Luego, se reordenan las muestras de entrenamiento con el objetivo de eliminar la temporalidad en los datos al pasarlos a los modelos.

6.2. Normalización

Se utiliza la normalización z-score sobre todo el conjunto de entrenamiento. La normalización z-score se define como:

$$z = \frac{(x-\mu)}{\sigma}$$

Donde:

- z: La muestra normalizada.
- x: La muestra a normalizar.
- μ: Media.
- σ: Desvío estándar.

7. Data Mining

En esta etapa del trabajo se ajustan los modelos para cada variable objetivo, modificando los parámetros del algoritmo Random Forest Regressor, con el objetivo de obtener el mayor rendimiento en la predicción.

7.1. Selección de modelos

Para obtener los parámetros que ofrecen mejores resultados en los modelos ajustados, se usa una búsqueda en cuadrícula de parámetros, proporcionada por la librería sklearn a través de la función **GridSearchCV().** La cual a partir de un conjunto de parámetros, con sus respectivos valores posibles, busca sobre todas las posibles combinaciones de parámetros y devuelve el estimador ajustado, con los parámetros que ofrecen la mayor puntuación (score) en el estimador, usando validación cruzada.

Los parámetros utilizados para realizar la búsqueda son:

- **n_estimators:** La cantidad de árboles en el modelo.
- **criterion:** La función para medir la calidad de una división. Los criterios admitidos son "mse" para el error cuadrático medio, que es igual a la reducción de la varianza como criterio de selección de características, y "mae" para el error absoluto medio.
- **max_depth:** La profundidad máxima de cada árbol.
- **min_samples_leaf:** El número mínimo de muestras necesarias para estar en un nodo hoja.
- **min_samples_split:** El número mínimo de muestras necesarias para dividir un nodo interno.

Los parámetros elegidos son los que no terminaban en su valor por defecto en sucesivas búsquedas en cuadrícula realizadas. La siguiente tabla muestra los valores elegidos para realizar la búsqueda en cuadrícula:

Parámetro	Valores posibles
n_estimators	5 a 5000
criterion	mse, mae
max_depth	5 a 5000
min_samples_leaf	5 a 20
min_samples_split	5 a 20

Tabla 2. Valores posibles para la búsqueda en cuadrícula.

7.2. Valores de Lag Feature

Para decidir el mejor valor de **n** para la variable Lag Feature (variable retrasada), se experimentó con distintos valores de **n**. Los mejores resultados obtenidos fueron para **n=1,** el cual se usará para mostrar los experimentos en la siguiente sección. Los resultados de estos experimentos se encuentran en notebooks en la siguiente carpeta compartida en Google Drive:
https://drive.google.com/drive/folders/1mRptfTKd-otq0mBx9tNx40xor9IRcjqX?usp=sharing

La tabla 2 compara la métrica R-Score de cada modelo ajustado, para los distintos valores de **n** en Lag Feature. Estos resultados se obtuvieron para las muestras de pruebas. La columna Modelo contiene los modelos para cada variable objetivo (target) a predecir.

Modelo	**Lag Feature 1**	**Lag Feature 7**	**Lag Feature 14**	**Lag Feature 30**
target 1	0.53	0.37	0.25	0.29
target 7	0.31	0.48	0.24	0.30
target 14	0.90	0.70	0.88	0.88
target 30	-1.62	-1.57	-1.38	-1.19

Tabla 3. Comparación R-Score de cada modelo para cada valor de **n** en Lag Feature.

Para **n = 1** se obtienen los mejores resultados, a excepción del target 7. El modelo para predecir la cantidad de casos 30 días después (target 30), no presenta un buen R-Score para ningún valor de **n** en Lag Feature.

8. Evaluación

Los mejores parámetros obtenidos en la búsqueda en cuadrícula para cada modelo ajustado se presentan en la siguiente tabla:

Modelo	**n_estimators**	**criterion**	**max_depth**	**min_samples_leaf**	**min_samples_split**
1 día después	5000	mse	5	5	5
7 días después	5000	mse	5000	5	5
14 días después	5000	mse	5000	5	5
30 días después	5000	mse	5000	5	5

Tabla 4. Comparación de los mejores parámetros elegidos en la búsqueda en cuadrícula.

En todos los casos puede observarse que se elige la mayor cantidad de estimadores, la mayor profundidad, la menor cantidad de muestras necesarias para estar en un nodo hoja y la menor cantidad de muestras necesarias para dividir un nodo interno. Los mejores parámetros encontrados para cada modelo es el mismo en todos los casos.

La siguiente tabla presenta las métricas obtenidas, sobre el conjunto de prueba, para los modelos creados para cada variable objetivo a predecir (1, 7, 14 y 30 días después).

Modelo	MSE	MAE	RMSE	Error Máximo	R-Score
1 día después	2046836.99	1384.26	1430.67	1997.75	0.37
7 día después	1130237.81	1013.44	1063.12	1592.08	0.30
14 días después	761712.80	760.12	872.76	1486.53	0.89
30 días después	7465678.84	2689.00	2732.33	3469.59	-1.64

Tabla 5. Comparación de métricas de evaluación para cada modelo ajustado.

Puede observarse que el modelo para 14 días después, es el que mejor predicción ofrece para los datos del conjunto de prueba. Claramente se observa que sus métricas de error son menores que los otros modelos y su R-Score es el mayor. El modelo para 30 días después, es el que menor rendimiento ofrece.

Los gráficos de las figuras 5, 6, 7 y 8 muestran el rendimiento de la predicción, para los 4 modelos creados, sobre el conjunto de entrenamiento. Se observa que todos los modelos se ajustan bien a sus datos de entrenamiento.

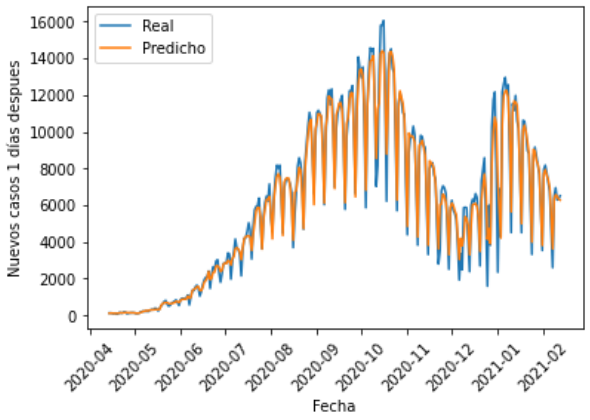

Figura 5. Predicción de nuevos casos de Covid-19 1 día después sobre el conjunto de entrenamiento..

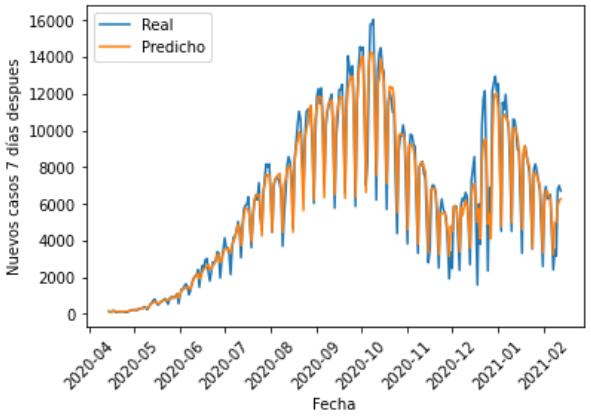

Figura 6. Predicción de nuevos casos de Covid-19 7 días después sobre el conjunto de entrenamiento.

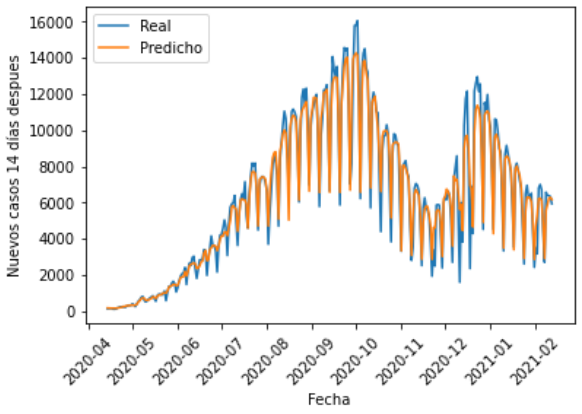

Figura 7. Predicción de nuevos casos de Covid-19 14 días después sobre el conjunto de entrenamiento.

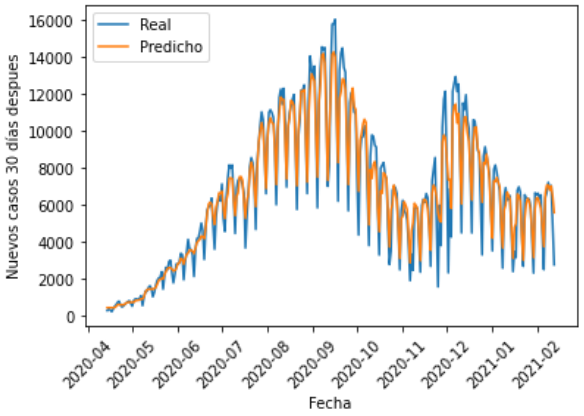

Figura 8. Predicción de nuevos casos de Covid-19 30 días después sobre el conjunto de entrenamiento.

Los gráficos de las figuras 9, 10, 11 y 12 muestran el rendimiento de la predicción, para los 4 modelos creados, sobre el conjunto de prueba. El modelo para predecir la cantidad de casos 14 días después es el mejor ajustado.

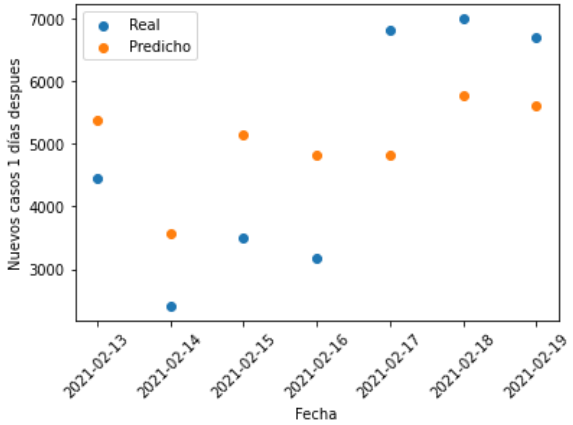

Figura 9. Predicción de nuevos casos de Covid-19 1 día después sobre el conjunto de prueba.

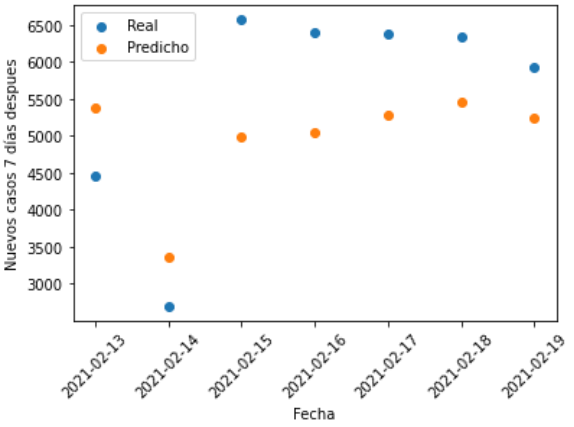

Figura 10. Predicción de nuevos casos de Covid-19 7 días después sobre el conjunto de prueba.

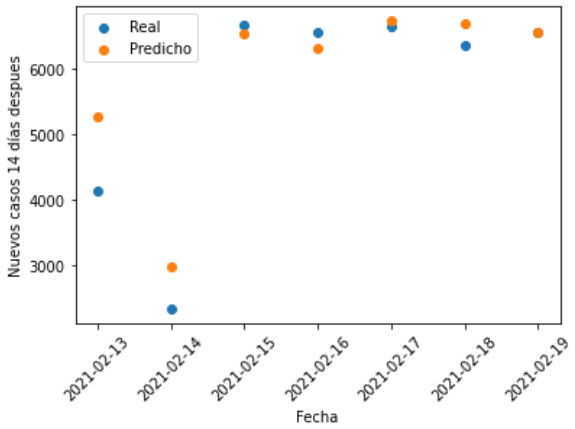

Figura 11. Predicción de nuevos casos de Covid-19 14 días después sobre el conjunto de pruebas.

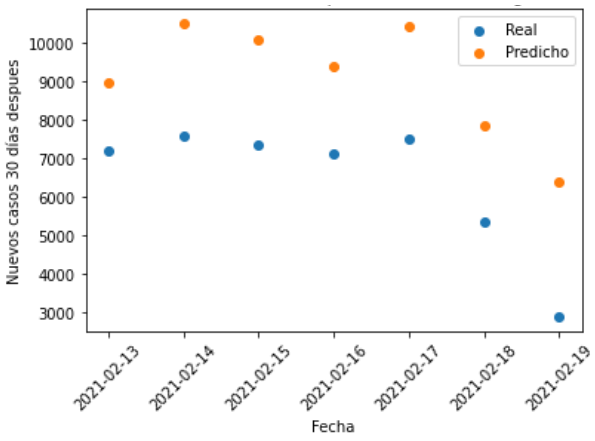

Figura 12. Predicción de nuevos casos de Covid-19 30 días después sobre el conjunto de prueba.

Los gráficos de las figuras 13, 14 y 15 representan el error que obtienen los modelos al predecir sobre el conjunto de prueba. Los gráficos corresponden a los modelos para predecir los nuevos casos entre 1, 7 y 14 días después. El modelo para predecir 30 días después no se presenta porque ofrece un muy bajo rendimiento.

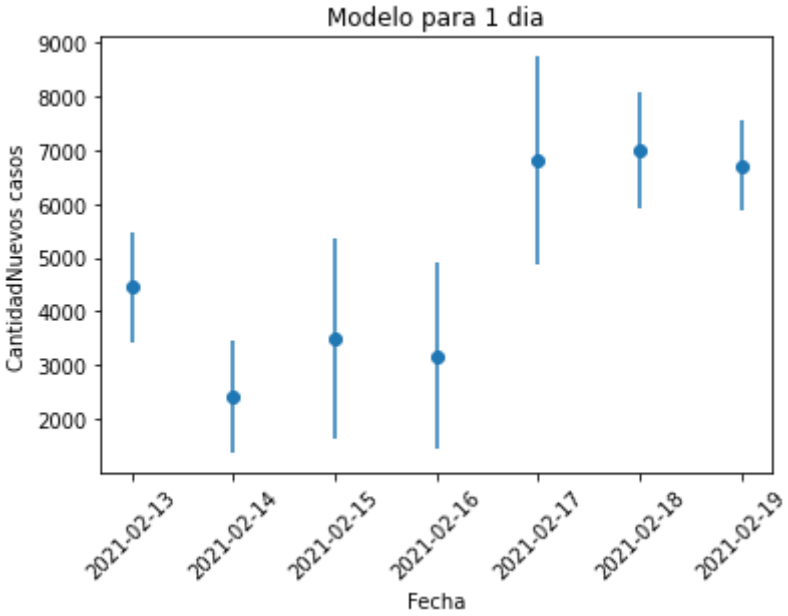

Figura 13. Gráfico barra de error para el modelo de predicción de 1 día después.

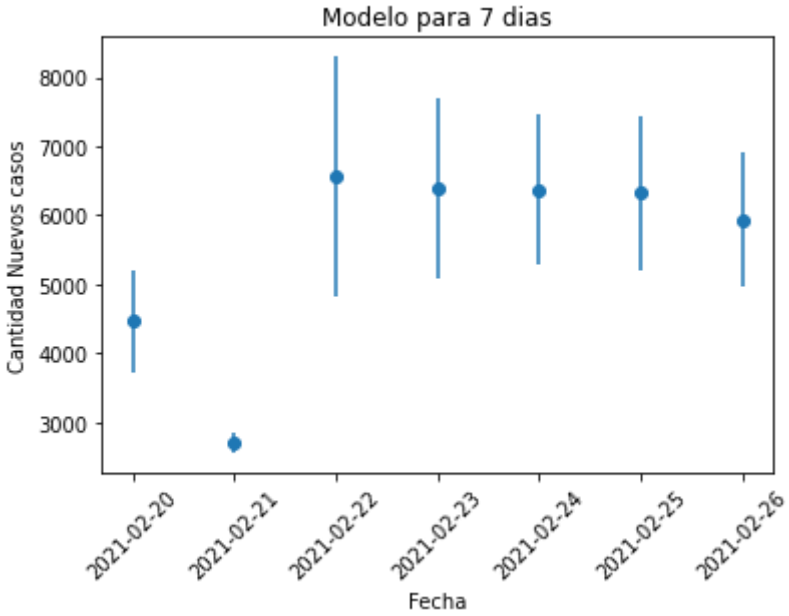

Figura 14. Gráfico barra de error para el modelo de predicción de 7 días después.

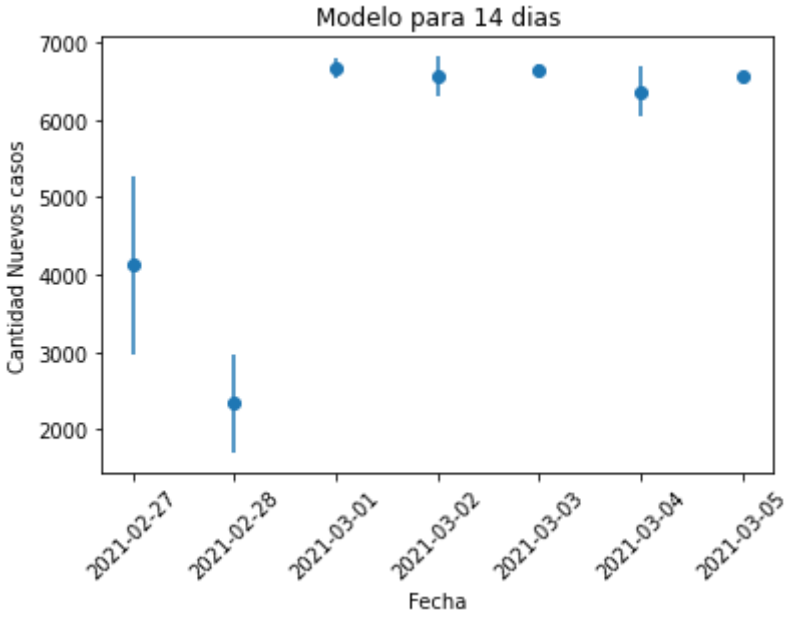

Figura 14. Gráfico barra de error para el modelo de predicción de 14 días después.

8.1. Importancia de cada variable predictora

Los gráficos de barras de las figuras 16, 17 y 18 presentan la importancia de cada variable predictora para los modelos predicción de 1, 7 y 14 días después respectivamente. La importancia de una característica se calcula como la reducción total

(normalizada) del criterio aportado por esa característica. También se le conoce como la importancia de Gini [7]. Este procedimiento está implementado como una característica, luego de ajustar el modelo Random Forest Regressor de la librería sklearn.

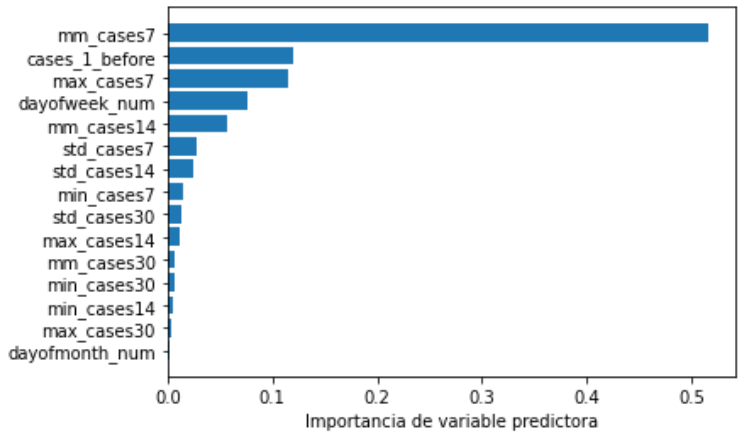

Figura 16. Comparación de importancia de variables predictoras para el modelo de 1 día después.

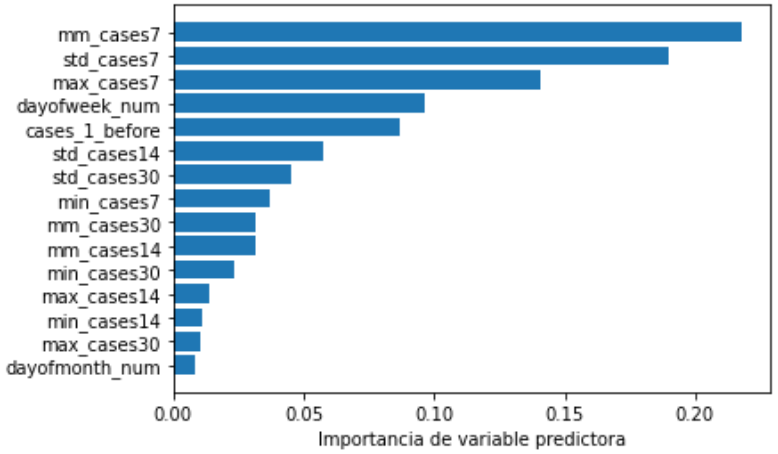

Figura 17. Comparación de importancia de variables predictoras para el modelo de 7 días después.

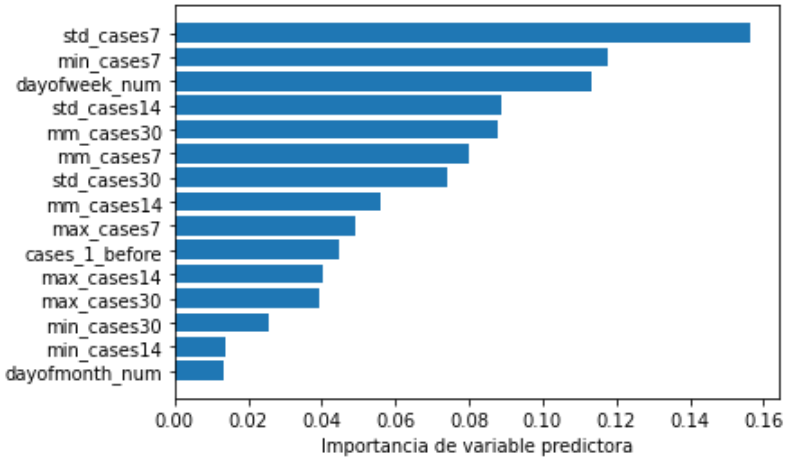

Figura 17. Comparación de importancia de variables predictoras para el modelo de 14 días después.

Puede observarse que la importancia de las variables predictoras varían en cada modelo. La única variable que se encuentra entre las primeras 4 posiciones en los modelos presentados es el día de la semana (**dayofweek_num**). La media móvil de los siete días previos (**mm_cases_7**), también tiene una alta importancia para todos los modelos.

9. Conclusiones

En este trabajo se construye un conjunto de datos para el ajuste y prueba del modelo de predicción Random Forest para regresión, a partir de los datos ofrecidos por el Ministerio de Salud de la República Argentina. Se transforman los datos a un Dataset para aprendizaje supervisado, se crean nuevas variables predictoras y nuevas variables objetivos, para estudiar el comportamiento de los modelos bajo distintas configuraciones. Además, se buscan los mejores parámetros para Random Forest Regressor a través de una búsqueda en cuadrícula. Por último, se compara la importancia de cada variable predictora para cada modelo ajustado.

A partir de los resultados de los experimentos de la sección anterior, puede concluirse que el mejor modelo ajustado corresponde al que predice cuántos nuevos casos habrá dentro de 14 días. El modelo peor ajustado es el que predice los nuevos casos dentro de 30 días. Además, los mejores parámetros elegidos para cada modelo en la búsqueda en cuadrícula son los mismos. Por último, las variables predictoras más importantes para los modelos ajustados varían según cual sea la variable objetivo a predecir.

10. Trabajos Futuros

Al finalizar los experimentos de este trabajo surgen nuevas preguntas y nuevos experimentos que pueden realizarse. A continuación, se detalla una lista de posibles trabajos que pueden realizarse en investigaciones futuras:

- Experimentar con otros posibles valores de parámetros en la búsqueda en cuadrícula.
- Experimentar con distintos tamaños de las muestras de entrenamiento y prueba.
- Filtrar los casos por ciudad o provincia y estudiar el comportamiento de los modelos para un lugar determinado de la Argentina.

- Agregar al Dataset las series cantidad de recuperados y cantidad de muertos por días y estudiar cómo afectan estas series al ajustar los modelos.
- Transformar la variable objetivo a escala logarítmica y observar si es posible mejorar la eficiencia en la predicción.
- Ajustar los modelos utilizando solo las variables predictoras que tienen mayor importancia en cada caso.
- Estudiar el comportamiento de los modelos con nuevas variables predictoras, no abordadas en este trabajo.

11. Referencias

[1] C.C. Lai, T.P. Shih, W.C. Ko, H.J. Tang, P.R. Hsueh. (2020). Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): the epidemic and the challenges. Int J Antimicrob Agents.

[2] Francesca Lazzeri. (2020). Machine Learning for Time Series Forecasting with Python. John Wiley & Sons.

[3] Hydrologic Variability of the Cosumnes River Floodplain. (2006). Booth et al., San Francisco Estuary and Watershed Science, Volume 4, Issue 2.

[4] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (p. 94). New York: springer.

[5] Cutler, Adele & Cutler, David & Stevens, John. (2011). Random Forests. 10.1007/978-1-4419-9326-7_5.

[6] Lior Rokach and Oded Maimon (2008). *Data mining with decision trees: theory and applications*. World Scientific. ISBN 978-981-277-171-1.

[7] Nembrini, Stefano & König, Inke & Wright, Marvin. (2018). The revival of the Gini Importance?. Bioinformatics (Oxford, England). 34. 10.1093/bioinformatics/bty373.