Estimación de un modelo de aprendizaje con regresión logística utilizando métodos de optimizacion numérica

1 Introducción

Tomando como referencia el trabajo de investigación realizado por Colubri et al. (2019), se busca entrenar un modelo de regresión logística que permite pronosticar la supervivencia a enfermedad por virus del ébola. Para tal propósito, se utilizan datos recolectados por el Cuerpo Internacional de Medicina (IMC, por sus siglas en inglés) durante 2014 y 2016 en Liberia y Sierra Leona. El fin del presente proyecto es evaluar el desempeño de diferentes métodos de optimización numérica en un contexto de *machine learning* utilizando métodos cómputo simples y en paralelo. En particular, se mide el desempeño tanto en términos del uso y tiempo de recursos computacionales para lograr convergencia, como con relación a métricas de desempeño del modelo. Finalmente, se realiza un análisis de los resultados en torno a la pregunta planteada inicialmente y el modelo de regresión asociado.

Este reporte está organizado como se menciona a conitinuación. La sección 2 menciona los objetivos a cumplir, e introduce el problema a resolver. La sección 4 introduce literatura relacionada al problema y a la implementación de métodos numéricos en un contexto de *machine learning*. La sección 5 describe el conjunto de datos a utilizar. La sección 6 describe la implementación y algoritmos utilizados para solucionar el problema. La sección 7 discute los resultados obtenidos. Finalmente, la sección 8 presenta las conclsuiones

2 Objetivos

El objetivo principal de este trabajo es evaluar el rendimiento de los métodos de optimización mencionados en la sección 6. Dado lo anterior, definimos rendimiento en términos de métricas de desempeño del modelo, y también en métricas de desempeño durante el proceso de computo. A continuación, se describe el planteamiento del problema a resolver.

Problema de clasificación

Como es resaltado por Murphy (2012), el problema de regresión logística es una generalización del problema de regresión lineal, convirtiendolo hacia un problema de clasificación, siempre y cuando la variable de respuesta sea de carácter binario (i.e. $y \in \{0, 1\}$), y por tanto se pueda asumir que sigue una distribución Bernoulli. En este caso, dicha variable de respuesta corresponde a si el paciente muere a casua del virus del ébola (1), o no (0).

Dado lo anterior, utilizamos:

$$Pr[y \mid \boldsymbol{x}, \boldsymbol{w}] = Ber(y \mid \mu(x)) \tag{1}$$

donde la media, se define en términos de la probabilidad de que el paciente muera, es decir, $\mu(x) = E[y \mid x] = p(y = 1 \mid x)$. Donde el conjunto de variables explicativas está representado por x.[1]

Por otro lado, para la realización del computo de la media, se utiliza la función sigmoide, $\sigma(x)$, la cual garantiza que dada una combinación lineal de las variables explicativas, con los parámetros del modelo (i.e. $\beta^T x$), se cumpla que $0 \leq \mu(x) \leq 1$:

$$\mu(x) = \sigma\left(\beta^T x\right) \tag{2}$$

donde σ está definida por:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} = \frac{e^x}{e^x + 1} \tag{3}$$

Los resultados asociados a la ecuación (2) están dados en términos de probabilidades. Así, es necesario definir un umbral clasificatorio para definir si el modelo predice que el paciente fallezca, o no. En este ejercicio, como se hace usualmente, se toma como umbral el valor de 0.5. Es decir:

$$\hat{y} = \begin{cases} 0, & \text{si } \sigma(\hat{\beta}^T x) \geq 0.5 \\ 1, & \text{si } \sigma(\hat{\beta}^T x) < 0.5 \end{cases} \tag{4}$$

Los parámetros asociados a las variables regresoras, $\hat{\beta}$, son estimados con el conjunto de datos de entrenamiento.

Función de pérdida

La función de pérdida asociada a este problema es la log-verosimilitud negativa, o entropía cruzada, definida de las siguiente forma:

$$LVN(\beta) = -\sum_{i=1}^{N} log\left[y_i \mu_i^{I(y_i=1)} \times (1 - y_i)^{I(y_i=0)}\right] \tag{5}$$

o de forma equivalente:

$$LVN(\beta) = -\sum_{i=1}^{N} [y_i log\mu_i + (1 - y_i) log(1 - \mu_i)] \tag{6}$$

Dada la definición de (6), se tiene que este problema no tiene solución analítica. Es por esta razón que para minimizar la función de pérdida en torno a β es requerido utilizar métodos de optimización numérica.

Para los algoritmos a utilizar, es imprecindible la utilización del gradiente y la matriz hessiana asoaciados a (6), los cuales se definen a continuación:

$$\nabla LVN = \boldsymbol{g} = \frac{d}{d\boldsymbol{\beta}} f(\boldsymbol{\beta}) = \sum_i (\mu_i - y_i) x_i = X^T(\mu - y) \tag{7}$$

$$\nabla^2 LVN = \boldsymbol{H} = \frac{d}{d\boldsymbol{\beta}} g(\boldsymbol{\beta})^T = \sum \left(\nabla_\beta \mu_i\right) x_i^T = \sum \mu_i(1 - \mu_i) x_i x_i^T = \boldsymbol{X}^T \boldsymbol{S} \boldsymbol{X} \tag{8}$$

[1] Las variables independientes son detalladas en las sección 5.

donde $\boldsymbol{S} \triangleq diag(\mu_i(1-\mu_i))$. Adicionalmente, como resalta Murphy (2012), dado que $\boldsymbol{H}$ es positiva definida, entonces (6) tiene un mínimo global que puede ser alcanzable utilizando métodos de optimización.

En la siguiente sección se definen los algoritmos numéricos a implementar.

3 Algoritmos de optimización

Para cumplir el objetivo de minimizar la entropía cruzada utlizaremos cuatro algoritmos de optimización numérica para encontrar el vector $\hat{\beta}$. Los algoritmos a utilizar serán: el método de máximo descenso, método de Newton, y método BFGS (Broyden, Fletcher, Goldfarb y Shanno). En particular, se aplican algunas variantes a los mismos, y se comparan las iteraciones para converger, además del tiempo de ejecución.

En lo que sigue de esta sección utilizaremos la notación de Nocedal and Wright (2006), en la cual se plantea que en un método de optimización se inicia con β_0 y el algoritmo genera una secuencia de iteraciones $\{\beta_k\}_{k=0}^{\infty}$ que se detienen cuando no sea se pueda progresar más en el proceso, o sea haya alcanzado una solución con precisión suficientemente buena. Así, los algoritmos incorporan un proceso según el cual la decisión entre moverse de β_k a β_{k+1} depende de haber alcanzado un valor menor en la función objetivo.

3.1 Método de máximo descenso

Una de las ventajas del método de descenso de gradiente es que solo requiere el cálculo del gradiente, sin embargo, también puede tener un desempeño bastante lento para problemas difíciles.

Algorithm 1: How to write algorithms

```
Result: Write here the result
initialization;
while While condition do
    instructions;
    if condition then
        instructions1;
        instructions2;
    else
        instructions3;
    end
end
```

3.2 Método de Newton

Algorithm 2: Método de Newton

Result: Write here the result
Inicializar β^0;
while *Mientras "no converge"* **do**
  instructions;
  if *condition* **then**
    Resolver d^k en $(\nabla^2 F(\beta^k))d^k = -\nabla F(\beta^k)$;
    $\beta^{k+1} \leftarrow \beta^k + \alpha^k d^k$;
  else
    detenerse;
  end
end

3.3 Método BFGS

1. Aproximar la inversa $\nabla^2 F(\beta^k)$ por medio de una función de rango-2 dada por:

$$\begin{aligned} H^{k+1} &= H^k + \frac{w^k (w^k)^T}{(w^k)^T z^k} - \frac{H^k z^k (H^k z^k)^T}{(z^k)^T H^k z^k} \\ z^k &= \beta^{k+1} - \beta^k \\ w^k &= \nabla F(\beta^(k+1) - \nabla F(\beta^k) \end{aligned} \tag{9}$$

2. Reescribir $F(\beta)$ como función de riesgo empírico.

3. Derivar $\nabla l(\beta^T x_i, y_i)$.

Algorithm 3: método de BFGS

Result: Write here the result
initialization;
while *While condition* **do**
  instructions;
  if *condition* **then**
    instructions1;
    instructions2;
  else
    instructions3;
  end
end

4 Revisión de Literatura

Como se mencionó al principio, el presente trabajo se apoya principalmente en el artículo *Machine Learning Models from 2014-2016 Ebola Outbreak-Data-harmonization Challenges, Validation, Strategies and mHealth Applications*, cuyo objetivo principal fue diseñar una app que, dado los síntomas de un paciente con ébola, calcula la probabilidad de muerte del paciente y recomienda

un tratamiento personalizado. Lo anterior, es resultado de la implementación de diferentes modelos de regresión logística, mediante los cuáles se identificaron los principales factores asociados a la muerte de una persona con ébola (case fatality rate, CFR).

4.1 Internal Medical Corps (IMC): Conjunto de datos de Entrenamiento

Considerando la situación sanitaria del 2014-2016, la app fue focalizada en regiones con mayor afectación y menor posibilidad de acceso a atención médica. Así, los modelos se entrenaron con datos provenientes de cinco unidades de tratamiento del ébola (ETU, por sus siglas en inglés), localizadas en Sierra Leona y Liberia. La base constaba con 470 observaciones (correspondientes a los pacientes atendidos), con variables como temperatura, escala de bienestar, time to presentantion (TTP), triaje, entre otras variables; además, a partir de la reacción en cadena de polimerasa se calculó el cycle threshold (CT).

4.2 Análisis bivariado y Exploratorio

Con estos datos se procedió con un análisis bivariado y exploratorio entre la variable dependiente (muerte o no muerte)y el resto de las variables, identificando la asociación entre cada uno de los posibles regresores y la variable dependiente. Para evaluar la asociación entre las variables numéricas y la variable dependiente, se realizaron *point biseral correlation tests*; para evaluar la asociación entre dos variables categóricas, se realizaron pruebas chi cuadradas con corrección de Yates.

El resultado de este primer análisis fue que más de 50 por ciento de los pacientes que murieron por ébola reportaron síntomas de pérdida de apetito, fiebre, astenia, dolor musculoso esquelético, dolor de cabeza y diarrea. Además, se identificó prevalencia de triajes notablemente diferente entre los resultados (mortalidad y no no mortalidad de los pacientes). Finalmente, observando el *p-value* se identificó que pocas variables se asocian significativamente con el resultado del paciente: CT, edad, ictericia (jaundice) –con un p-value menor a 0.05; conjuntivitis, confusión, disnea, dolor de cabeza y sangrado – con un p-value menor a 0.15.

4.3 Imputaciones Múltiples

Un problema suscitado al trabajar con estos datos, fue la cantidad de valores faltantes (22 por ciento respecto del total de potenciales predictores). Con base en lo anterior, se tomaron las siguientes acciones: se eliminaron los predictores potenciales que tuvieran más de 50 por ciento de valores faltantes; se evaluó la idoneidad de imputar los valores faltantes mediante la prueba Little's MCAR, cuya hipótesis nula establece que los datos faltantes se debe totalmente al azar y se implementa probando si las medias de las variables de interés, a través de distintos grupos no cambian. Resultado del test, se aceptó que los datos faltantes se debían completamente al azar y se realizó una imputación múltiple con la función aregImpute del paquete de R Hmisc. En resumen, la imutación consistió en la generación de: una distribución predictiva bayesiana a partir de los datos conocidos y un número N de conjuntos de datos imputados. Cada valor faltante en la i-ésima imputación se predijo a partir de un modelo aditivio ajustado en una muestra de bootstrap con reemplazo de los datos originales.

4.4 Selección de Variables

Con el fin de contar con un modelo parsimonioso, los autores ejecutaron un Elastic Net Regularization, misma que combina las normas $L1$ y $L2$ de las regresiones de Lasso y Ridge, para la selección

del subconjunto de características relevantes y no redundantes. Los criterios de selección de los regresores candidatos no redudantes y relevantes fueron los siguientes: se mantuvieron las variables que tuvieron un coeficiente positivo en al menos 50 por ciento de las regresiones penalizadas; se eliminaron las variables con p-value bajo y se incorporaron las variables que tuvieron una asociación débil en el análisis bivariado. El resultado fue un conjunto de variables que se usaron en un modelo denominado parsimonioso.

Además, las variables seleccionadas fueron evaluadas por sesgo; luego, se aplicaron splines cúbicos restringidos para modelar las relaciones no lineales entre CFR y edad y CFR y temperatura corporal, donde los valores reales de edad y temperatura fueron la entrada del procedimiento de ajuste del spline, tal como se implementó en el paquete Hmisc.

4.5 Modelos de Regresión Logística: Modelo Parsimonioso y Modelos Alternativos

Derivado del análisis y procedimiento anterior, se implementaron varios modelos: el modelo parsimonioso – edad del paciente, edad del paciente al cubo, temperatura corporal, temperatura corporal al cubo, amarillamiento, sangrado, disnea, disfagia, TTP y CT x TTP– y tres modelos adicionales que se emplearon cuando los datos de triaje eran limitados: parsimonioso sin temperatura –idéntico al parsimonioso pero con eliminación de la temperatura corporal–, sólo clínico –ictericia, sangrado, disnea, disfagia, debilidad y diarrea– y el modelo mínimo –TC y edad. Cada modelo final se obtuvo ajustando N copias del modelo, en cada conjunto de datos imputados, y promediando esas copias en un solo modelo usando 'fit.mult.impute' de Hmisc.

Una predicción se clasificó como muerte cuando la puntuación del modelo superó el umbral de 0.5. Los intervalos de confianza (IC) de todas las estimaciones de rendimiento se calcularon utilizando la transformación de Fisher. Las razones de probabilidad (OR) de la regresión logística se convirtieron en razones de riesgo (RR).

4.6 Resultados

El modelo parsimonioso representó de manera adecuada las probabilidades reales de muerte, con cierta subestimación para pacientes de bajo riesgo de muerte.

Los modelos mínimos y solo clínicos resultaron menos calibrados, con el primero subestimando las probabilidades reales tanto en el lado de bajo y alto riesgo, y el segundo sobreestimando las probabilidades reales para una amplia gama de riesgos por encima del 60 por ciento.

Los índices de validación de todos modelos, junto con sus intervalos de confianza (al 95 por ciento), se superponen. La especificidad y R2 fue mayor en el modelo parsimoniosos. Mientras que la sensibilidad fue ligeramente mayor en el modelo mínimo. El modelo clínico muestró un rendimiento consistentemente menor.

Para el caso del modelo parsimonioso, la clasificación de todas las variables por su importancia, medida por el estadístico Wald $\tilde{\chi}^2$ – prueba estadística paramétrica que se utiliza para poner a prueba el verdadero valor del parámetro basado en la estimación de la muestra–, están dadas en el siguiente orden: CT, edad del paciente, ictericia, hemorragia, temperatura corporal y la interacción $CT - TTP$.

En términos de los odds ratios del modelo parsimonioso, se concluyó que la presentación de ictericia o sangrado se asocia con más del doble de las probabilidades de muerte, aunque su prevalencia es baja (al 5 por ciento). Finalmente, al transformar los riesgos de probabilidad en razones de riesgo, se identificó una estimación del 4 por ciento de aumento en el riesgo asociado con la aparición de esos síntomas.

4.7 Validación externa: KGH y GOAL

Los resultados fueron validados con dos bases de datos –provenientes del Kenema Government Hospital (KGH) y GOAL–; el primero constaba con 106 casos con ébola y una tasa de fatalidad (CFR, por sus siglas en inglés) de 73 por ciento; el segundo constaba de 158 casos con ébola, con una tasa de fatalidad de 60 por ciento.

Al igual que la base de datos de entrenamiento, ambas bases de datos empleadas durante la validación, contaban con registros incompletos, razón por la cuál se les aplicaron también imputaciones múltiples.

En general, los resultados de la validación fueron los siguientes: todos los modelos, con excepción del modelo clínico, exhibieron un rendimiento consistente en términos del área bajo la curva (AUC) y precisión general, con diferencias análogas en términos de especificidad y sensibilidad. El modelo parsiomonioso presentó mayor especificidad, mientras que el modelo mínimo fue más sensible. El impacto de la imputación fue intrascendente.

5 Conjunto de Datos

Por la restricciones de uso de la base de datos de *entrenamiento* original (además de una serie de procedimientos díficiles de seguir como protocolos de investigación y evidencia de aprobación por nuestro protocolo de investigación por parte de un Comité de ética independiente), optamos por trabajar con una de las dos bases de datos que los autores emplearon para *validar* sus modelos: KGH. La base de datos en mención, como se mencionó en la sección anterior, consta de 106 casos positivos de pacientes con ébola y un CFR global de 73 por ciento. Originalmente, previo al tratamiento de los datos, la base tenía únicamente 44 registros de triaje, 58 registros de carga viral, con un total de 78 valores faltantes en todo el data set.

Para harmonizar los datos, los autores transformaron la carga viral en CT, conforme con la curva estándar qPCR:

$$log_{(carga\ viral)} = m * CT + c_0$$

Los resultados de la transformación no dependieron del origen geográfico de los datos IMC. Sin embargo, el origen geográfico tuvo un impacto significativo ($p < 0.0001$) en la distribución de CT a través de los sitios.

Para corregir dicho sesgo analítico y garantizar que los resultados fueran internamente consistentes en cada sitio y, por lo tanto, comparables entre los sitios, se normalizaron los valores de CT de cada sitio, mediante el escalado de características (restando la media y dividiendo por la desviación estándar).

Como las diferencias en CT también podrían provocar diferencias en los comportamientos de búsqueda de atención, entonces se normalizaron los valores de la CT dentro de los sitios también contribuiría a reducir el efecto de este posible factor de confusión -una variable o factor que distorsiona la medida de asociación entre otras dos variables.

Nosotros, para fines del presente trabajo, empleamos una de las versiones imputadas de esta base de datos, dispuesta en el siguiente sitio: ebola-imc-public, misma que cuenta con 11 variables.

Así, para evaluar el rendimiento de los métodos de optimización mencionados en la sección 5, realizaremos dos regresiones logísticas: el modelo de regresión logística clínicas (ictericia, sangrado, disfagia, debilidad y diarrea), junto con el CT y el modelo de regresión logística simple (CT y edad como regresores). Esto principalmente se debe al limitado insumo inicial (la base de datos dispuesta

Tipo	Nombre	Descripción
Variables Numéricas	CT	El cycle threshold (CT) es una variable que se calcula a partir de una relación médica bien conocida (qPCR) y la carga viral (una expresión númerica de la cantidad de virus dado un volúmen de fluido que normalmente se correlaciona con la severidad de una infección viral activa).
	TEMP	Temperatura corporal del paciente. Toma valores de 36 a 39.9
	AGE	Edad del paciente. Toma valores de 1 a 73.
Variables Categóricas	HEADCH	Presencia o no dolores de cabeza. Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no dolores de cabeza.
	BLEED	Presencia o no de sangrado. Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no sangrado.
	DIARR	Presencia o no de diarrea. Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no diarrea.
	VOMIT	Dificultad para comer, conocido como disfagia, término técnico para describir el síntoma consistente en dificultad para la deglución (problemas para tragar). Esta dificultad suele ir acompañada de dolores, a veces lancinantes (disfagia dolorosa u odinofagia . Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no de disfacia.
	PABD	Presencia o no de PADB.
	WEAK	Presencia o no de debilidad o fatiga general.Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no debilidad.
	JAUN	Condición en la cuál la piel, los ojos y los miembros mucosos que vuelven amarillos debido a altos niveles de bilirubina. Toma valores valores 1 o 0, dependiendo de si el paciente presenta o no ictericia.
	OUT	Muerte o no muerte del paciente. Toma valores 1 o 0. Dependiendo de si el paciente muere o no muere.

Table 1: Variables del Dataset Kenema Government Hospital (KGH): Versión Imputada 50.

al público), mismo que contiene un sub-conjunto reducido de las variables originales empleadas por los autores para el entrenamiento del modelo.

6 Implementación

La implementación de este ejercicio se realizará por medio de código escrito en lenguaje Python. En particular, con el fin de minimizar la función de pérdida de log-verosimilitud negativa (y el riesgo empírico en SGD), se incluirán módulos propios para resolver el problema. Los pasos a seguir son los siguientes:

1. Implementar *grad_F* y hess_F en Python.
2. Implementar el método de máximo descenso para minimizar $F(\beta)$. Elegimos un β^0 aleatorio y una tolerancia de $\epsilon = 10^{-8}$.
3. Implementar un clasificador de regresión logística para obtener $\hat{y}$.
4. Implementar el método de Newton para minimizar $F(\beta)$
5. Implementar el método BFGS

6. Implementar el método SGD
7. Implementar variabilidad en tasa de aprendizaje (condición de Armijo).
8. Paralelizar procesos; por ejemplo: resolver en paralelo la dirección del descenso.
9. Dockerizar ambiente
10. Comparar tiempos de ejecución
11. Unittest

Con el fin de generar un entorno aislado que permita evaluar el desempeño del proceso de entrenamiento en término de recursos computacionales, se utilizará una instancia EC2 de Amazon Web Services (AWS). Adicionalmente, para tener un control del entorno virtual asociado, se creará una imagen de docker asociada al repositorio donde se encontrarán todos los códigos de este proyecto.

7 Resultados

Los resultados del entrenamiento del modelo descritos en las sección 2, se presentan enl a tabla en la tabla 2.

	$\hat{\beta}$ **por método**		
Variable	**Máximo descenso**	**Newton**	**BFGS**
CT	-9.92694e+02	-768.31808	-3.10562e+02
HEAD	1.67710e+03	1294.40177	5.16584e+02
TEMP	-5.738344e+02	-443.39949	-1.7789e+02
BLEED	-1.477851e-01	-290.57819	-4.97200e-01
DIARR	-1.15112e+00	-1.95730	-3.51086e+00
VOMIT	6.49333e+01	296.70658	2.25010e+01
PABD	5.98420e+01	-201.20433	2.08288e+01
Weak	4.39599e+02	339.7508	1.36628e+02
hasta22	2.93896e+02	227.64517	9.26912e+01
entre23y36	4.407495e+02	342.07282	1.41270e+02
entre37y45	7.21806e+02	559.97208	2.32028e+02
mayor45	6.39504e+02	495.88330	2.03073e+02
Error de clasificación	9.09 %	13.64 %	13.64 %
Iteraciones	201355	1880	1000000
Criterio de parada	Convergencia	Convergencia	Max. Iteraciones
Norma del gradiente	1.19990e-06	1.19449e-06	0.00590
User time	45.5 s	678 ms	4 min 50 s
Sys time	2.26 s	60.2 ms	4.09 s
Total time	47.7 s	738 ms	4 min 54 s

Table 2: Resultados de los parámetros estimados del modelo. **Nota:** La variable CT denota *cycle threshold;* TEMP denota tempratura corporal; HEADCH denota presencia o no de dolor de cabeza, BLEED denota presencia o no de sangrado; DIARR denota presencia o no de diarrea; VOMIT denota disfagia; PABD denota presencia o no de PADB, WEAK denota presencia o no de debilidad o fatiga. Los tiempos están en las siguientes unidades: ms denotan microsegundos, min denotan minutos y s denotan segundos.

A modo general, tres conclusiones grandes se pueden presentar de lo presentado en la tabla 2:

- El algoritmo más eficiente en términos de computo (que incluye iteraciones y tiempo de cómputo) fue el método de Newton. POr otro lado, el más ineficiente fue el BFGS.
- El algoritmo más preciso, fue el de máximo descenso, teniendo una superioridad de aproximadamente.
- Tanto el método de Newton como el de máximo descenso lograron terminar el proceso iterativo por criterio de convergencia. Por otro lado, luego de 1,000,000 de iteraciones no logró encontrar dicho criterio.

7.1 Interpretación de los resultados en el modelo de regresión logística

8 Conclusiones

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:




import pandas as pd

df_proc = df_raw
df_proc['INTER_AGE'] = "NA"
print()

1/10*a
df_proc.dtypes



def grad_cost_func(X,y,beta):
    mu=calcula_mu(X,beta)
    return np.matmul(np.transpose(X),mu-y)

def mini_lotes(X,y,q=10):
    cols=X.shape[1]
    data=np.hstack((X,y[:,None]))
    np.random.shuffle(data)
    data=data[0:q]
    X=data[:,0:cols]
    y=data[:,cols]
    return X,y
```

9 Conclusiones

References

Colubri, A., Hartley, M.-A., Siakor, M., Wolfman, V., Felix, A., Sesay, T., Shaffer, J. G., Garry, R. F., Grant, D. S., Levine, A. C., et al. (2019). Machine-learning prognostic models from

the 2014–16 ebola outbreak: Data-harmonization challenges, validation strategies, and health applications. *EClinicalMedicine*, 11:54–64.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Appendix A Funciones de Python

Appendix B Código de Python