



Innovación y Software

ISSN: 2708-0927

ISSN: 2708-0935

facin.innosoft@ulasalle.edu.pe

Universidad La Salle

Perú

Tesillo Gomez, Cynthia Mayumi; Escobar Arcaya, Yuri Alexander; León Gutierrez, Edwin Daniel  
Predicción de hipertensión arterial a través de un sistema de regresión logística  
Innovación y Software, vol. 2, núm. 2, 2021, Septiembre-Febrero, pp. 60-74  
Universidad La Salle  
Perú

Disponible en: <https://www.redalyc.org/articulo.oa?id=673870839005>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso  
abierto

Tipo de artículo: Artículos originales

Temática: Inteligencia artificial

Recibido: 23/06/2021 | Aceptado: 05/08/2021 | Publicado: 30/09/2021

Identificadores persistentes:

ARK: [ark:/42411/s6/a44](https://nbn-resolving.org/urn:nbn:org:ark:42411-s6-a44)

PURL: [42411/s6/a44](https://nbn-resolving.org/urn:nbn:org:ark:42411-s6-a44)

# Predicción de hipertensión arterial a través de un sistema de regresión logística

## *Prediction of arterial hypertension through a logistic regression system*

Cynthia Mayumi Tesillo Gomez <sup>1</sup>[\[0000-0002-1769-9845\]](https://orcid.org/0000-0002-1769-9845), Yuri Alexander Escobar Arcaya <sup>2</sup>[\[0000-0001-5220-058X\]](https://orcid.org/0000-0001-5220-058X)\*, Edwin Daniel León Gutierrez <sup>3</sup>[\[0000-0002-2519-1785\]](https://orcid.org/0000-0002-2519-1785).

<sup>1</sup> Universidad Nacional Jorge Basadre Grohmann, Tacna, Perú [cynthia.tesillo@unjbg.edu.pe](mailto:cynthia.tesillo@unjbg.edu.pe)

<sup>2</sup> Universidad Nacional Jorge Basadre Grohmann, Tacna, Perú. [yuri.escobar@unjbg.edu.pe](mailto:yuri.escobar@unjbg.edu.pe)

<sup>3</sup> Universidad Nacional Jorge Basadre Grohmann, Tacna, Perú. [edwin.leon@unjbg.edu.pe](mailto:edwin.leon@unjbg.edu.pe)

\* Autor para correspondencia: [yuri.escobar@unjbg.edu.pe](mailto:yuri.escobar@unjbg.edu.pe)

---

### Resumen

En el Perú y el mundo entero la hipertensión es una enfermedad que puede avanzar sin manifestar ningún síntoma o éstos ser muy leves. Se puede tener hipertensión arterial y no sentir ninguna manifestación, la hipertensión arterial es un serio problema de salud pública en países en desarrollo como el nuestro: según la Encuesta Demográfica y de Salud Familiar de 2017, aunque la prevalencia de hipertensión en personas de 15 años a más se habría reducido de 14,8 % en 2014, a 13,6 %, implica que más de 3 millones de peruanos viven con hipertensión arterial. Por ese motivo nuestro objetivo es el rápido diagnóstico de esta enfermedad silenciosa, en el presente trabajo se utilizó el sistema de regresión logística, para el cual se posee un *dataset* de 5615 registros analizados. Este artículo presenta la posibilidad de detectar una enfermedad como la hipertensión arterial basado en inteligencia artificial, ya que este mal ha ido aumentando en los últimos años. Por ese motivo el objetivo es predecir de manera rápida un posible diagnóstico de hipertensión arterial, para ello se analizó un *dataset* de 5615 registros en la aplicación web Jupyter Notebook, estableciendo 9 variables de entrada y 1 de salida, además se utilizó el sistema de regresión logística, tratamientos de datos *missing* y *outliers*, gráficas de variables, obteniendo como resultado una precisión media aceptable del 87%.

**Palabras clave:** Hipertensión arterial, Inteligencia Artificial, Regresión logística, Presión arterial.

### **Abstract**

*In Peru and the entire world, hypertension is a disease that can progress without showing any symptoms or these being very mild. You can have high blood pressure and not feel any manifestations, arterial hypertension is a serious public health problem in developing countries like ours: According to the 2017 Demographic and Family Health Survey Survey, although the prevalence of hypertension in people aged 15 years and over would have decreased from 14.8% in 2014 to 13.6%, it implies that more than 3 million Peruvians live with high blood pressure. For this reason, our goal is the rapid diagnosis of this silent disease. In the present work, the logistic regression system was used, for which there is a dataset of 5615 analyzed records. This article presents the possibility of detecting a disease such as high blood pressure based on artificial intelligence, since this evil has been increasing in the last years. For this reason, the objective is to quickly predict a possible diagnosis of arterial hypertension, for this, a dataset of 5615 records was analyzed in the Jupyter Notebook web application, establishing 9 input variables and 1 output, in addition, the logistic regression system was used, missing data treatments and outliers, graphs of variables, obtaining as a result an acceptable average precision of 87%.*

**Keywords:** Arterial hypertension, Artificial Intelligence, Blood Pressure, Logistic Regression.

---

## **Introducción**

Actualmente se estima que en el mundo hay 1130 millones de personas con hipertensión, y la mayoría son de bajos recursos, en donde una de cada cinco personas hipertensas no lo tiene controlado, esta enfermedad es una de las causas principales de muerte prematura en el mundo es por ello que la Organización Mundial de Salud tiene como meta reducir la prevalencia de la hipertensión en un 25% para el año 2025 [1].

En el Perú la hipertensión arterial es una preocupación constante entre los médicos e investigadores, aún más en estas fechas de pandemia, ya que el Ministerio de Salud del Perú estima que el número personas con esta enfermedad aumentaría en un 20% lo que constituye a un problema de salud pública y conlleva a la aparición de nuevas enfermedades [2].

Por ese motivo la importancia del rápido diagnóstico de esta enfermedad silenciosa, la prevención primaria, es decir anticipar la aparición de una enfermedad que como la hipertensión arterial esencial no tiene una causa conocida, no

es tarea fácil, sin embargo, hoy es evidente la existencia de factores que aumentan el riesgo de padecer la enfermedad y que deben ser conocidos por la población [3].

Se trabajó con *dataset* externo almacenado en página web de zenodo.org. del 27 de febrero de 2021, Criterios de cambio hipertensión Perú, para aplicar Predicción de hipertensión arterial a través técnica de Regresión Logística para clasificar patrones. A partir de un conjunto de datos de entrada con una variable de salida.

Para la realización del presente trabajo se tuvo que aplicar métodos de limpieza de datos a nuestro *dataset*, como la eliminación de columnas, la evaluación del procesamiento datos *missing* y la eliminación de datos *outliers*. seguidamente en este artículo se presenta la aplicación del sistema de regresión logística la cual consiste en una técnica estadística multivariante que nos permite estimar la relación existente entre una variable dependiente no métrica, en particular dicotómica y un conjunto de variables independientes métricas o no métricas [4].

Lo que se desea lograr en este artículo es predecir si una persona tiene o es propensa a sufrir de hipertensión, para así tener un diagnóstico oportuno o tratamiento adecuado. La realización del presente artículo se divide en diferentes secciones; en primer lugar, mostramos una introducción, la cual nos da una visión general del problema a atacar, luego se ve a detalle los métodos utilizados para poder realizar este trabajo conociendo a profundidad la regresión logística, debido a que esta se utiliza como base. Seguidamente pasamos a los resultados obtenidos, los cuales son satisfactorios obteniendo un 87 % de aciertos, llegando así a la parte final del trabajo expresando las conclusiones.

## **Materiales y métodos o Metodología computacional**

### **Herramientas**

**Google Colaboratory:** Es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube.

**Google Drive:** Es un servicio de alojamiento de archivos que fue introducido por la empresa estadounidense Google el 24 de abril de 2012. Es el reemplazo de Google Docs que ha cambiado su dirección URL, entre otras cosas. Es uno de los sitios de alojamiento más conocidos en el mundo.

### **Librerías**

**NumPy:** proporciona una estructura de datos universal que posibilita el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos [5].

**Pandas:** es una de las librerías de python más útiles para los científicos de datos. Las estructuras de datos principales en pandas son Series para datos en una dimensión y DataFrame para datos en dos dimensiones. Estas son las estructuras de datos más usadas en muchos campos tales como finanzas, estadística, ciencias sociales y muchas áreas de ingeniería. Pandas destaca por lo fácil y flexible que hace la manipulación de datos y el análisis de datos [5].

**scikit learn:** es una de estas librerías gratuitas para Python. Cuenta con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Además, presenta la compatibilidad con otras librerías de Python como NumPy, SciPy y matplotlib [6].

**sklearn.model\_selection import train\_test\_split:** nos permite dividir un dataset en dos bloques, típicamente bloques destinados al entrenamiento y validación del modelo (llamemos a estos bloques "bloque de entrenamiento " y "bloque de pruebas" para mantener la coherencia con el nombre de la función) [7].

**sklearn.metrics import accuracy\_score:** En la clasificación de etiquetas múltiples, esta función calcula la precisión del subconjunto: el conjunto de etiquetas predichas para una muestra debe coincidir exactamente con el conjunto de etiquetas correspondiente en y true [8].

**sklearn.metrics import classification\_report:** crea un informe de texto que muestre las principales métricas de clasificación [8].

**matplotlib:** es una librería de Python especializada en la creación de gráficos en dos dimensiones, como histograma, diagramas de sectores, diagramas de caja y bigotes, diagramas de violín, diagramas de dispersión o puntos, diagramas de líneas, diagramas de áreas, diagramas de contorno y mapas de color [9].

### **Métodos o Metodología computacional**

El método elegido para realizar la investigación es el método Regresión Logística. Por lo tanto, se tratará de investigar el estudio de los principales factores de riesgo de la hipertensión arterial, cómo influyen las características, como un problema de salud.

La investigación se realiza mediante Regresión Logística con variables de entrada y salida, relacionados a la hipertensión, para poder diagnosticar esta enfermedad a partir de sus características clasificando resultados en valores discretos. Para el trabajo se ha creado un archivo hipertension.csv con datos de entrada, para el presente artículo

científico se considera nueve características para el diagnóstico de la enfermedad hipertensiva, según referencia del ministerio de salud, variables como: sexo, edad, presión sistólica, presión diastólica, peso, talla, fuma, actividad física y región. como resultado de salida se considera hipertensión.

La regresión logística es el conjunto de modelos estadísticos utilizados cuando se desea conocer la relación entre

- Una variable dependiente cualitativa, dicotómica (regresión logística binaria o binomial) o con más de dos categorías (regresión logística multinomial).
- Una o más variables explicativas independientes, llamadas covariables, ya sean cualitativas o cuantitativas.

Las covariables cualitativas deben ser dicotómicas, tomando valor 0 para su ausencia y 1 para su presencia. Si la covariable tuviera más de dos categorías debemos realizar una transformación de la misma en varias covariables cualitativas dicotómicas ficticias (variables *dummy*). Al hacer esta transformación cada categoría de la variable entraría en el modelo de forma individual.

Los modelos de regresión logística tienen tres finalidades:

- Cuantificar la importancia de la relación existente entre cada una de las covariables y la variable dependiente.
- Clarificar la existencia de interacción y confusión entre covariables respecto a la variable dependiente (es decir, los *odds* ratio para cada covariable).
- Clasificar individuos dentro de las categorías (presente/ausente) de la variable dependiente.

Los valores posibles de estas ecuaciones varían entre 0 y 1. Un valor cercano a 0 significa que es muy improbable que Y haya ocurrido, y un valor cercano a 1 significa que es muy probable que tuviese lugar.

Como en la regresión lineal cada variable predictora de la ecuación logística tiene su propio coeficiente. Los valores de los parámetros se estiman utilizando el método de máxima verosimilitud que selecciona los coeficientes que hacen más probable que los valores observados ocurran [10].

**Para la obtención de un sistema de Predicción de hipertensión arterial, el cual para su desarrollo se utilizaron varias librerías las cuales se tienen que implementar al inicio del sistema. Las cuales se detallan a continuación.**

**1. Importar las librerías las cuales nos ayudarán con el procesamiento y tratamiento de datos.**

*import numpy as np*

```
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
#from matplotlib import cm
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn import model_selection
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

2. Importamos el *dataset* almacenado en nuestra nube de Google Drive, para ello se usará la librería Pandas, así mismo se ejecutará `encoding='latin-1'` para evitar el Error UTF-8.

```
dataset = pd.read_csv('/content/gdrive/My Drive/hipertension.csv', encoding='latin-1')
```

3. Imprimimos en pantalla parte del *dataset*, para poder visualizar total las variables (columnas) y sus datos.

```
dataset.head()
```

Tabla 1: Descripción del *dataset*.

	id	city	masl	sex	age_years	systolic_bp	diastolic_bp	weight_kg	height_cm	body_mass_index	diabetes_mellitus	dm_treatment
0	3574	Huancayo	3250	Female	23	119	74	58.0	163.0	22.0	No	No
1	1092	Loreto	100	Male	60	110	70	54.0	160.0	21.0	No	No
2	861	Lima	500	Female	38	120	80	65.0	163.0	25.0	No	No
3	835	Lima	500	Female	43	110	80	60.0	157.0	24.0	No	No
4	4654	Hu?nuco	1900	Female	30	95	60	50.0	152.0	22.0	No	No

4. Después de un análisis, se concluyó que las variables id, ciudad, masa corporal, diabetes mellitus, tratamiento, enfermedades, años de fumador, años con hipertensión, tratamiento de la hipertensión, msnm, presión arterial máxima antigua/nueva y mínima antigua/nueva no son causas fundamentales para el desarrollo de la variable de salida, o son variables las cuales tiene similitud con otras, por lo que se procedió a eliminarlas.

```
dataset = dataset.drop(['id', 'city', 'masl', 'body_mass_index', 'diabetes_mellitus', 'dm_treatment', 'cv_diseases',
                        'cd_treatment', 'smoking_years', 'hypertension_years', 'hypertension_treatment', 'msnm', 'sist_old', 'diast_old',
                        'sist_new', 'diast_new'], axis=1)
dataset.head()
```

Tabla 2: Descripción de las variables finales.

	sex	age_years	systolic_bp	diastolic_bp	weight_kg	height_cm	smoking	physical_activity	region	hypertension_dx
0	Female	23	119	74	58.0	163.0	No	Yes	Mountain	No
1	Male	60	110	70	54.0	160.0	No	No	Jungle	No
2	Female	38	120	80	65.0	163.0	No	Yes	Coast	No
3	Female	43	110	80	60.0	157.0	Yes	No	Coast	No
4	Female	30	95	60	50.0	152.0	No	Yes	Mountain	No

## DICCIONARIO DE VARIABLES.

- sex: SEXO Condición orgánica que se distingue entre hombres y mujeres.
- age\_years: EDAD Lapso de tiempo que transcurre desde el nacimiento hasta el momento de referencia.
- systolic\_bp: PRESIÓN SISTÓLICA Es la presión arterial máxima
- diastolic\_bp: PRESIÓN DIASTÓLICA Es la presión arterial mínima
- weight\_kg: PESO\_kg Cantidad de masa de la persona expresada en kilogramos
- height\_cm: TALLA\_cm Medida de la estatura de la persona expresada en centímetros
- smoking: FUMA Condición de la persona donde se detalla si fuma o no
- physical\_activity: ACTIVIDAD FÍSICA Define si la persona realiza actividad física continuamente
- region: REGION Define el lugar donde vive la persona
- hypertension\_dx: HIPERTENSIÓN Define si la persona es Hipertenso o no



5. Al ser nuestro sistema una regresión logística, necesitamos que nuestros datos sean de tipo numérico, por lo que consultamos qué tipo de variables tenemos:

*dataset.dtypes*

```
sex                object
age_years          int64
systolic_bp        int64
diastolic_bp       int64
weight_kg          float64
height_cm          float64
smoking            object
physical_activity   object
region             object
hypertension_dx     object
dtype: object
```

Figura 1: Tipos de datos de las variables.

En Python, el tipo de datos de texto se conoce como secuencia de caracteres (*string*). En Pandas se los conoce como objetos (*object*). Las secuencias de caracteres pueden contener números y / o caracteres.

6. Al ver que hay datos de tipo objeto se optó por reemplazar los valores de las columnas *sex*, *smoking*, *physical\_activity*, *region*, *hypertension\_dx* por valores tipo *int* o *float* con el método *map* y un diccionario entre llaves.

```
dataset['sex'] = dataset['sex'].map({'Male':0,'Female':1})
dataset['smoking'] = dataset['smoking'].map({'No':0,'Yes':1})
dataset['physical_activity'] = dataset['physical_activity'].map({'No':0,'Yes':1})
dataset['region'] = dataset['region'].map({'Coast':1,'Mountain':2,'Jungle':3})
dataset['hypertension_dx'] = dataset['hypertension_dx'].map({'No':0,'Yes':1})
dataset.head()
```

Tabla 3: Variables con los mapeos correspondientes.

	sex	age_years	systolic_bp	diastolic_bp	weight_kg	height_cm	smoking	physical_activity	region	hypertension_dx
0	1	23	119	74	58.0	163.0	0.0	1	2	0.0
1	0	60	110	70	54.0	160.0	0.0	0	3	0.0
2	1	38	120	80	65.0	163.0	0.0	1	1	0.0
3	1	43	110	80	60.0	157.0	1.0	0	1	0.0
4	1	30	95	60	50.0	152.0	0.0	1	2	0.0

Se muestra el nuevo *dataset* con valores numéricos.

- Se realizó el tratamiento de datos *missing*, para este caso se tomó la decisión de eliminarlos, se opta por esto ya que esos datos pueden introducir errores mayores en los resultados, por otro lado el sistema de regresión logística solo acepta números, y el dataset debe estar con valores vacíos o “NaN”

*dataset.describe()*

Tabla 4: Eliminación de *outliers*.

	sex	age_years	systolic_bp	diastolic_bp	weight_kg	height_cm	smoking	physical_activity	region	hypertension_dx
count	5615.000000	5615.000000	5615.000000	5615.000000	5418.000000	5414.000000	5483.000000	5615.000000	5615.000000	5495.000000
mean	0.627427	43.463224	112.792698	71.932146	64.344592	156.390469	0.095386	0.483882	1.876759	0.140855
std	0.483533	17.030004	15.844951	11.324560	11.408517	8.592941	0.293774	0.499785	0.463625	0.347904
min	0.000000	18.000000	55.000000	25.000000	30.000000	105.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	29.000000	101.000000	65.000000	56.000000	150.000000	0.000000	0.000000	2.000000	0.000000
50%	1.000000	43.000000	112.000000	71.000000	64.000000	156.000000	0.000000	0.000000	2.000000	0.000000
75%	1.000000	55.000000	122.000000	80.000000	71.000000	163.000000	0.000000	1.000000	2.000000	0.000000
max	1.000000	97.000000	200.000000	119.000000	150.000000	185.000000	1.000000	1.000000	3.000000	1.000000

Se observa en la fila *count* que algunos valores difieren del total de filas del *dataset* 5615, lo cual se deduce la existencia de datos *missing*.

*dataset = dataset.dropna()*

*dataset.describe()*

Tabla 5: Eliminación de *missing*.

	sex	age_years	systolic_bp	diastolic_bp	weight_kg	height_cm	smoking	physical_activity	region	hypertension_dx
count	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000	5170.000000
mean	0.634623	43.586654	112.962669	72.061122	64.308897	156.377950	0.097099	0.479884	1.867311	0.146422
std	0.481582	17.090609	15.848137	11.352873	11.446475	8.603421	0.296121	0.499644	0.476772	0.353563
min	0.000000	18.000000	55.000000	25.000000	30.000000	105.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	29.000000	102.000000	65.000000	56.000000	150.000000	0.000000	0.000000	2.000000	0.000000
50%	1.000000	43.000000	112.000000	71.000000	64.000000	156.000000	0.000000	0.000000	2.000000	0.000000
75%	1.000000	56.000000	122.000000	80.000000	71.000000	163.000000	0.000000	1.000000	2.000000	0.000000
max	1.000000	97.000000	200.000000	119.000000	150.000000	185.000000	1.000000	1.000000	3.000000	1.000000

El método *dropna* permite, de una forma muy conveniente, filtrar los valores de una estructura de datos pandas para dejar solo aquellos no nulos.

## 8. Se realizó en tratamiento de datos *outliers*, para los cual se graficó las variables de entrada y salida `plt.show` de la librería Matplotlib

```
dataset.drop(['hypertension_dx'],1).hist()
plt.show()
```

Luego de un análisis se determinó la no existencia de datos *outliers* en las variables de entrada.

```
dataset.drop(['sex','age_years','systolic_bp','diastolic_bp','weight_kg','height_cm','smoking','physical_activity','region'],1).hist()
plt.show()
```

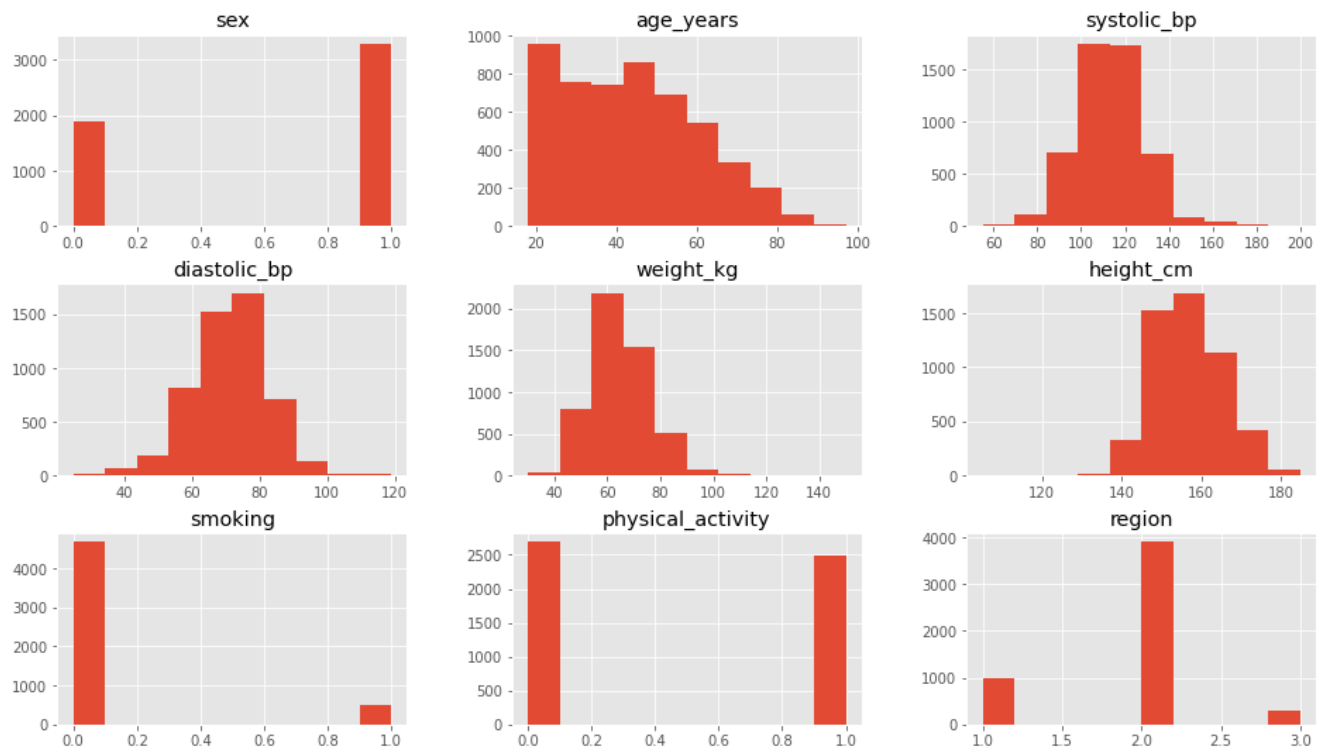


Figura 2: Histograma de las variables de entrada.

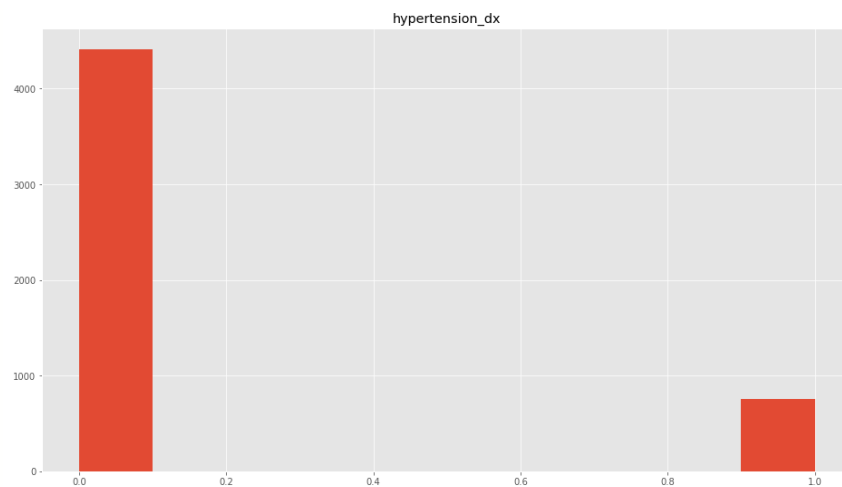


Figura 2: Histograma de las variable de salida.

Luego de un análisis se determinó la no existencia de datos *outliers* en la variable de salida.

## 9. Creación del modelo de Regresión Logística

Almacenamos en una matriz las 9 variables de entrada en la variable X y la variable de salida “hypertension\_dx” en la variable Y.

```
X = np.array(dataset.drop(['hypertension_dx'],1))  
y = np.array(dataset['hypertension_dx'])  
X.shape
```

Se creó nuestro modelo y se ajustó a nuestro conjunto de entradas X y salidas Y.

```
model = linear_model.LogisticRegression()  
model.fit(X,y)
```

## 10. Una vez compilado nuestro modelo, le hacemos clasificar todo nuestro conjunto de entradas X utilizando el método “predict(X)” y revisamos algunas de sus salidas y vemos que coincide con las salidas reales de nuestro archivo csv.

```
predictions = model.predict(X)  
print(predictions)
```

## 11. Y confirmamos cuan bueno fue nuestro modelo utilizando model.score() que nos devuelve la precisión media de las predicciones, en nuestro caso del 86.94%.

```
model.score(X,y)  
0.8694390715667312
```

## 12. Validación de nuestro modelo

Para ello se dividirá el dataset en forma aleatoria en 80% entrenamiento y 20% prueba para la validación.

```
validation_size = 0.20
```

```
seed = 7  
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, y, test_size=validation_size,  
random_state=seed)
```

## Resultados y discusión

Y ahora hacemos las predicciones -en realidad clasificación- utilizando nuestro “cross validation set”, es decir del subconjunto que habíamos apartado. En este caso vemos que los aciertos fueron del 88% que es un resultado aceptable.

```
predictions = model.predict(X_validation)  
print(accuracy_score(Y_validation, predictions))  
0.8820116054158608
```

Dentro de los resultados tenemos la matriz de confusión la cual tenemos que codificar de la siguiente forma:

```
print(confusion_matrix(Y_validation, predictions))
```

y esta nos da como resultado:

```
[[867 14]  
 [108 45]]
```

Donde muestra cuántos resultados equivocados tuvo de cada clase (los que no están en la diagonal), en nuestro caso predijo que 108 pasos negativos cuando estos eran positivos y predijo que 14 eran positivos cuando en realidad eran positivos.

También podemos ver el reporte de clasificación con nuestro conjunto de Validación. En nuestro caso vemos que se utilizaron como “soporte” 881 registros negativos y 153 positivos. La valoración que de aquí nos conviene tener en cuenta es la de F1-score, que tiene en cuenta la precisión y *recall*. El promedio de F1 es de 87% lo cual no está nada mal.

```
print(classification_report(Y_validation, predictions))  
precision recall f1-score support
```

0.0	0.89	0.98	0.93	881
1.0	0.76	0.29	0.42	153

<i>accuracy</i>		0.88	1034	
<i>macro avg</i>	0.83	0.64	0.68	1034
<i>weighted avg</i>	0.87	0.88	0.86	1034

Para poder comprobar se colocaron datos de dos filas, una con resultado negativo y otra con resultado positivo, obteniendo la respuesta satisfactoria.

*#fila 590 valor respuesta 0*

```
X_new =
pd.DataFrame({'sex':[1], 'age_years':[27], 'systolic_bp':[71], 'diastolic_bp':[37], 'weight_kg':[51], 'height_cm':[150], 'smoking':[0], 'physical_activity':[0], 'region':[2]})
model.predict(X_new)

array([0.] )
```

*#fila 11 valor respuesta 1*

```
X_new =
pd.DataFrame({'sex':[0], 'age_years':[58], 'systolic_bp':[180], 'diastolic_bp':[86], 'weight_kg':[61], 'height_cm':[162], 'smoking':[1], 'physical_activity':[1], 'region':[2]})
model.predict(X_new)

array([1.] )
```

## Conclusiones

En conclusión, se puede decir que el presente trabajo tiene un acierto del 88% en sus predicciones de hipertensión arterial a través de un sistema de regresión logística, por lo que puede predecir si una persona tiene o es propensa a sufrir de hipertensión. El uso de regresión logística, se usa normalmente cuando se quiere estimar la relación existente entre una variable dependiente, y un conjunto de variables independientes métricas o no métricas.

## Referencias

- [1] Organización Mundial de la Salud. <https://www.who.int/es/news-room/fact-sheets/detail/hypertension>
- [2] Ministerio de Salud el 18 de mayo de 2021 - 2:55 p. <https://www.gob.pe/institucion/minsa/noticias/493681->

[minsa-estima-que-pacientes-con-hipertension-arterial-aumentarian-en-20-durante-la-pandemia](#)

[3] Dr.Raul Gamboa, Revista Peruana de Cardiología: Octubre - Diciembre 1993

[https://sisbib.unmsm.edu.pe/bvrevistas/cardiologia/v19\\_n2/la%20hiper.htm](https://sisbib.unmsm.edu.pe/bvrevistas/cardiologia/v19_n2/la%20hiper.htm)

[4]Celia Mercedes Salcedo Poma, Capitulo 2- Modelo de Regresión Logística

[https://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/Salcedo\\_pc/enPDF/Cap2.PDF](https://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Basic/Salcedo_pc/enPDF/Cap2.PDF)

[5] Jose Martinez Heras-Guía rápida de IArtificial.net <https://www.iartificial.net/guia-rapida-iartificial-net/>

[6] Universidad de Alcalá, “Scikit-learn, herramienta básica para el data science en Python” 2021.

<https://www.master-data-scientist.com/scikit-learn-data-science/>

[7] Interactive Chaos, 2019. <https://interactivechaos.com/es/python/function/sklearnmodelselectiontraintestsplit>

[8] Scikit Learn, “Metrics and scoring: quantifying the quality of predictions” [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html)

[9] Aprende con Alf, “La librería Matplotlib” <https://aprendeconalf.es/docencia/python/manual/matplotlib/>

[10]María Elvira Ferre Jaén.FEIR 45: Regresión logística. Jueves 04 abril 2019,23:30:47

<https://gauss.inf.um.es/feir/45/>

[11] Failoc-Rojas Virgilio, Dataset externo almacenado zenodo.org. Criterios de cambio hipertensión Perú, datos

<https://zenodo.org/record/4567767#.YOsR0hKhPZ>