International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I10P207
Volume 70 Issue 10, 54-60, October 2022

Original Article

Application of Machine Learning for the Prediction of Strokes in Peru

Hernan Matta-Solis[1], Rosa Perez-Siguas[1], Eduardo Matta-Solis[1], Lourdes Matta-Zamudio[1], Segundo Millones-Gomez[2], Jehovanni Fabricio Velarde-Molina[3]

[1] TIC Research Center: eHealth & eEducation, Instituto Peruano de Salud Familiar, Lima -Perú.
[2] Instituto de Medicina Legal y Ciencias Forenses (IML), Lima-Perú.
[3] Escuela de posgrado Newman, Tacna-Perú.

Received: 24 July 2022 Revised: 22 September 2022 Accepted: 06 October 2022 Published: 19 October 2022

Abstract *- Strokes are one of the most common causes of death or disability worldwide; several proposals have been put forward to reduce these accidents. The goal of the research is to create a machine learning model, which will help us predict the probability of how likely a person is to suffer or suffer a stroke. To do this, machine learning techniques were applied, as these have evolved exponentially over the years, and a dataset of stroke patients and stroke-free patients was used to train the model. As a result, our model obtained an accuracy of 77% for patients who could suffer from this disease, after which prevention can be done and thus achieve a decrease in the mortality rate from strokes.*

Keywords *- Machine Learning, Logistic regression, Stroke, Prediction model.*

1. Introduction

Currently, World Health Organization (WHO) studies report that strokes are among the leading causes of death. Stroke is one of the main causes, so much so that by 2030 stroke will remain the second cause in the world with 12.2% of deaths, and it is also predicted that those affected between 35% and 52% die from haemorrhagic stroke within a month [1].

In Peru, this health problem aggravates citizens because of the cerebrovascular accident (stroke), in 2018 it had a total of 12835 were affected, specifically in adults aged 35 to 65 years, with a rate of infections of 95% in men, a total of 7066 cases and in women 5769 [2]. There was an increase in affected, and also in the pandemic has been detected sequelae of stroke associated with the SARS-COV-2 virus through a series of cases with severe and critical affected patients. Hence, it has a similarity with respect to its genome of 82%, so some infected were prone to develop ischemic stroke [3].

In recent years, the use of data mining algorithms in predictive medical analytics has increased due to serious research in related areas. In recent years, several researchers have postulated that it is possible to acquire clinical care support and predictive models from basic patient data. [4].

Machine learning (ML) is a subset of artificial intelligence that builds a mathematical model based on sample data, known as "training data," to make predictions or decisions without being explicitly programmed to perform the task. Regarding learning, a good definition given by Mitchell is: A computer program is said to learn from experience E with respect to some tasks T and performance measures P if its performance on tasks of T, measured by P, improves with experience E [5].

Making an accurate prediction of the onset of the disease can be of great clinical value to healthcare professionals. A highly effective data-driven predictive algorithm is desired to increase the efficiency of disease prevention and improve patient outcomes through early detection and treatment [13].

That is why, in the present research work, as an objective, a solution to this problem is proposed with the creation of a machine learning model, which will help us predict the probability that a person is so prone to suffer or suffer a stroke, and thus be able to let him know his condition and danger, and be able to take preventive measures against it.

The present research was structured as follows, defined in 5 sections, section II describes the methodology to be used, section III develops the case study, section IV details the results, and section V carry out the conclusions and works towards the future.

2. Methodology

The section details the steps taken to build a model for machine learning, as shown in Fig. 1.

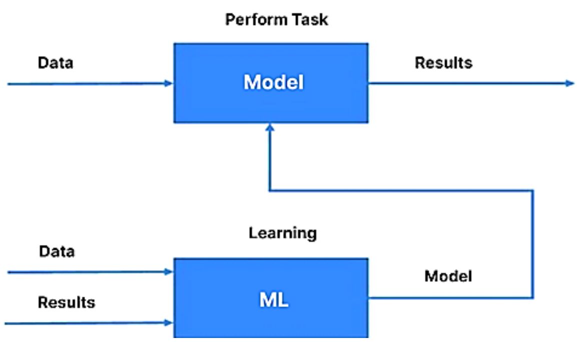


Fig. 1. Stage of a machine learning project

2.1. Data Collection and Obtaining

As a first step, data were collected from patients who had suffered a stroke and those who had not suffered it. The information was obtained from various sources such as web pages, blogs, spreadsheets, databases, and comma-separated values (CSV) files, among others, so that we can process this data and use it as training for our model.

2.2. Data Preparation

Publicly available datasets are usually not cleaned. Therefore, we will have to ensure that they are cleaned and, as a result, appropriate to build our model. Depending on this step, performance metrics are important, such as model accuracy and performance [7].

At this stage, the extraction, exploration, understanding and cleaning of our data will be carried out, which will be very important for the quality of the result. Every point is important, but most of the time in a machine learning project is spent cleaning up the data, preventing duplicate, null, or NaN (unavailable) fields from being found. This could result in a less accurate prediction.

3. Model Construction

In essence, machine learning is about learning the training samples given to solve one of two basic problems: regression (for continuous outputs) or classification (for discrete outputs). Classification is closely related to pattern recognition; its goal is to design a classifier that learns a set of "training" input data to perform the collection or classification of unknown samples. By regression, it means designing a regressor or predictor based on the machine learning results of a training dataset to predict unknown continuous samples [5].

4. Case Study

4.1. Data Collection and Obtaining

In the present work, we used a dataset extracted from the Kaggle platform called Stroke Prediction Dataset [14], which contains information about people without any disease and people who suffered a stroke. The dataset was published by Federico Soriano Palacios, a data scientist from the city of Madrid.

4.2. Data Preparation

As you can see in Fig. 2, the first step was to import the libraries to be used, libraries such as numpy, pandas, matplotlib, seaborn, and sklearn for the management of vectors and matrices, data manipulation, creation of graphs with less syntax and the construction of the model respectively.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

Fig. 2 Importing libraries

After this, our data was imported. To do this, Google Colaboratory or Google Colab was used to carry out our code in Python. One more import was made to bring our data from Google Drive, as shown in Fig. 3.

```
from google.colab import drive
drive.mount('/content/drive/')

Mounted at /content/drive/
```

Fig. 3 Access to Google Drive.

Before exploring our data, our CSV file was read, and our data was saved in a df variable with the pd.read_csv() function and then with the df. head() function the first rows of our DataFrame, as seen in fig. 4.

```
df = pd.read_csv('/content/drive/MyDrive/DataFrames/healthcare-dataset-stroke-data.csv')
df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Fig. 4 Reading the dataset

As seen in Fig. 4, Table 1. was obtained where each dataset's column was detailed.

Table 1. Description of Attributes

Description of the columns	
Id	unique identification
Gender	Male or Female
age	patient's age
hypertension	0 if the patient does not have hypertension, 1 if the patient has hypertension
heart disease	0 if the patient does not have any heart disease, 1 if the patient has heart disease
ever married	No or yes
work type	Children, Govt jov, never worked, Private or Self-employed
Residence type	Rural the Urban
avg glucose level	an average blood glucose level
BMI	body mass index
smoking status	I used to smoke" I never smoked; I smoked o Unknown
Stroke	1 if the patient had a stroke or 0 if not

Next, in Fig. 5, the data cleansing process began after having the file ready. As a first step, it was verified that there was no data duplication and NaN with the functions df.duplicated() and df.isna().sum() respectively.

As seen in Fig. 5, the output shows us that the BMI column contains 201 NaN or null values, missing data, which can interfere with our analysis and prediction of the model. One possible solution would be to remove the rows containing the missing values easily, but this would be a disadvantage for us as valuable data would also be lost to our model. The solution was to fill them with a value close to the median of the column with the function df.fillna(), and the result was saved in a new variable called df1, as seen in Fig. 6.

In Fig. 6, you can see that the values of type NaN were filled with a number close to the median of the column, with the function df.isna().sum(); another revision of the data was performed, as you will see in Fig. 7.

As shown in Fig. 8, with the function df1.gender.value counts (), we see that the gender column has 3 possible values.

It is observed that there are 2994 patients belonging to the female gender, 2115 to the male sex and 1 patient was registered as "other", which does not help us since we have a patient with incomplete data, so the record was deleted, using the df1.drop() function and then the result was checked using the df1.gender.value counts() Again, as can be seen in Fig. 9, the patient with the value "Other" no longer exists in our dataset.

```
dpl = df.duplicated().sum()
row, col = df.shape
print('Columns: {} Rows: {}'.format(col, row),
      'Duplicate rows: {}'.format(dpl),
      'Number of NaN values per column: ',
      df.isna().sum(),
      sep='\n')
```

```
Columns: 12 Rows: 5110
Duplicate rows: 0
Number of NaN values per column:
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

Fig. 5 Checking for lost and duplicate data.

```
df1 = df.fillna(value=df.mean())
print('Before...: \n{}'.format( df['bmi'].head() ),
      'After...: \n{}'.format( df1['bmi'].head() ),
      sep='\n')
```

```
Before...:
0    36.6
1     NaN
2    32.5
3    34.4
4    24.0
Name: bmi, dtype: float64
After...:
0    36.600000
1    28.893237
2    32.500000
3    34.400000
4    24.000000
Name: bmi, dtype: float64
```

Fig. 6 Fill in the missing values with the median

```
df1.isna().sum()
```

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64
```

Fig. 7 Checking NaN values

```
dt = df1.gender.value_counts()
dt.plot(kind='bar', rot=0, title="Total patients by gender")
print(df1.gender.value_counts())
```

```
Female    2994
Male      2115
Other        1
Name: gender, dtype: int64
```

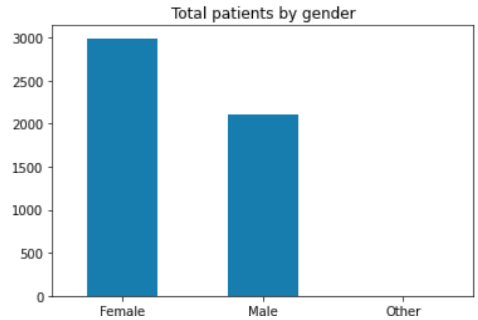


Fig. 8 Gender values

```
df1 = df1.drop(df1[df1['gender']=='Other'].index)
print(df1.gender.value_counts())
df1.gender.value_counts().\
plot(kind='bar', rot=0, title="Total patients by gender")
```

```
Female    2994
Male      2115
Name: gender, dtype: int64
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b2510edd0>
```

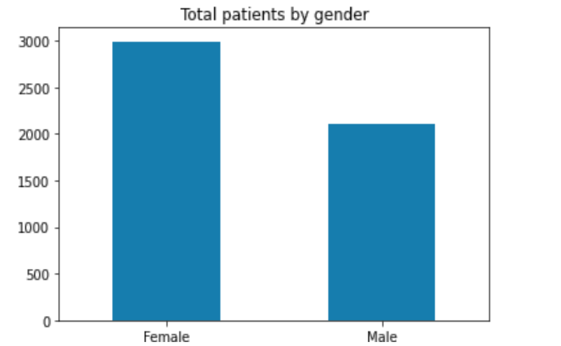


Fig. 9 Checking gender column values

```
enc = LabelEncoder()
df1.loc[:, ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']] = \
df1.loc[:, ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']].apply(enc.fit_transform)
df1.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	36.600000	1	1
1	51676	0	61.0	0	0	1	3	0	202.21	28.893237	2	1
2	31112	1	80.0	0	1	1	2	0	105.92	32.500000	2	1
3	60182	0	49.0	0	0	1	2	1	171.23	34.400000	3	1
4	1665	0	79.0	1	0	1	3	0	174.12	24.000000	2	1

Fig. 10 Data Encryption

```
plt.figure(figsize=(11,10))
sns.heatmap(df1.iloc[:,1:].corr(), cmap="Reds", vmax=1, vmin=-1, annot=True)
```

Fig. 11 Code that generates the heat map

Before we begin model construction and training, as shown in Fig. 4, our dataset contains object type values or type string text, and the model only supports entering numeric type values. It was done to encode these object-type values numerically, as shown in Fig. 10.

The next thing was to create a heat map of the correlation that exists between the variables, to know how dependent they are between them; as seen in Fig. 11, the function df1.corr() was used.

Pearson's correlation coefficient [9] can take values between +1 to -1; a positive value means that there is a dependency. If one goes up, the other does the same; in case it is a negative value, the dependence is inverse. In case it is 0, it would mean that there is no relationship between the variables, as shown in the following heat map in Fig. 12.

5. Model Construction

The model chosen was Logistic Regression, a method used to predict binary classes, which can be used to calculate the probability of an event occurring.

The first thing before creating a Logistic Regression model is.

[15] is to define the characteristics and the objective, where the characteristics are the values where the model will look for patterns among them and will result in the objective, which is the prediction of the model. They were defined in a variable called "X" for the characteristics and a variable "y" for the target. After this, oversampling and subsampling techniques were applied to our dataset as it was unbalanced, class 0 was much larger than class 1, and our prediction model would be affected due to this data imbalance. The subsampling approach eliminates instances of the majority class so that it can be balanced with the minority class. But

deleting overrepresented data points can lead to the loss of important data and jeopardize the classification process. The motivation for oversampling is that it is effective for very unbalanced data sets [11]. For this, we use the SMOTETomek class as a solution. Using the LogisticRegression() function of the sklearn library, our logistic regression model was created. The next thing was to standardize the data using the StandarScaler class of the sklearn. Preprocessing library, since the data must be of the same level or standard. Next, Fig. 13 shows each step explained above.

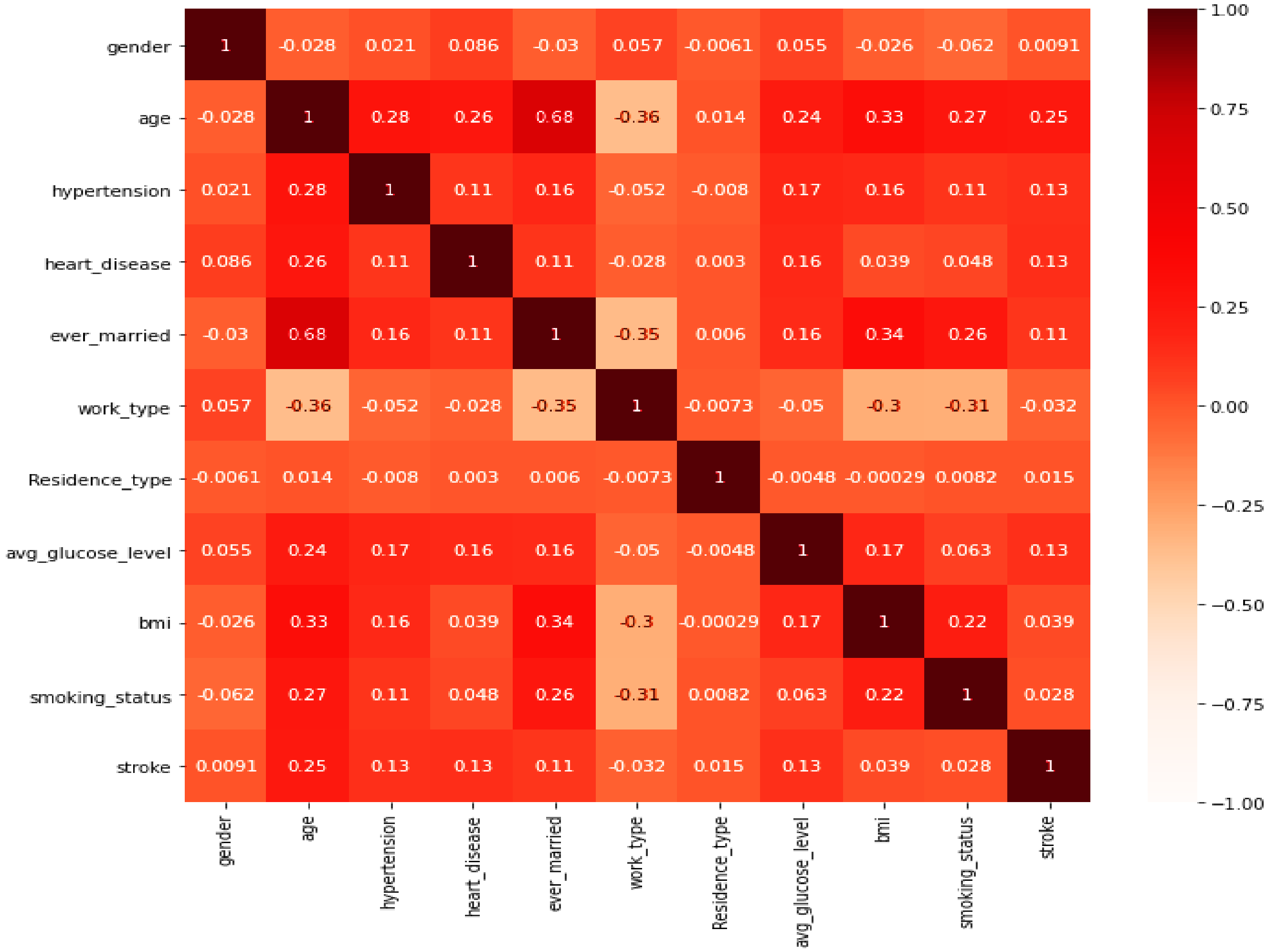


Fig. 12 Heat map of Pearson correlation coefficients

```
X = df1.iloc[:,1:-1]
y = df1.stroke
smote = SMOTETomek()
X_train_res, y_train_res = smote.fit_resample(X,y)
X_train, X_test, y_train, y_test = train_test_split\
(X_train_res, y_train_res, test_size=0.3, random_state=10)
model =  LogisticRegression()
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Fig. 13 Creation of our Logistic Regression model.

After creating our model, the next step was to train the model with our dataset with the model. fit() function.

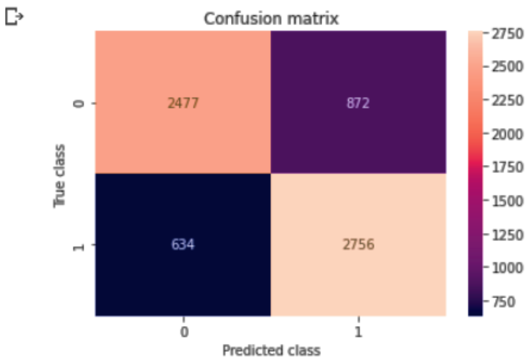
```
model = model.fit(X_train, y_train)
y_predTrain = model.predict(X_train)
model.score(X_train, y_train)
```

```
0.7765247069298116
```

Fig. 14 Training and model results

Fig. 14 shows that our model has a 77% accuracy with training data; in Fig. 15, it is seen through the heat map that 2477 true positives (VP), 872 false positives (FP), 634 false negatives (FN) and 2,576 true negatives (VN).

```
conf_matrix = confusion_matrix(y_train, y_predTrain)
sns.heatmap(data=conf_matrix, annot=True, fmt="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
print (classification_report(y_train, y_predTrain))
```




```
              precision    recall  f1-score   support

           0       0.80      0.74      0.77      3349
           1       0.76      0.81      0.79      3390

    accuracy                           0.78      6739
   macro avg       0.78      0.78      0.78      6739
weighted avg       0.78      0.78      0.78      6739
```

Fig. 15 Results Matrix confusion with training data

6. Results

The following are the expected results of this research, divided into a case study and methodology.

6.1. About the Case Study

As detailed above, this research paper deals with master's learning, which is one of the branches of Artificial Intelligence. Next, Table II shows a small description of the operation of some branches of Artificial Intelligence that exist. Fig. 15 shows a procedure very similar to that of Fig. 14, with the only difference being that this time we use new data that our model has never seen to know the result of the precision that will be obtained with the new data, as can be seen in Fig. 16.

```
y_predTest = model.predict(X_test)
model.score(X_test,y_test)
```

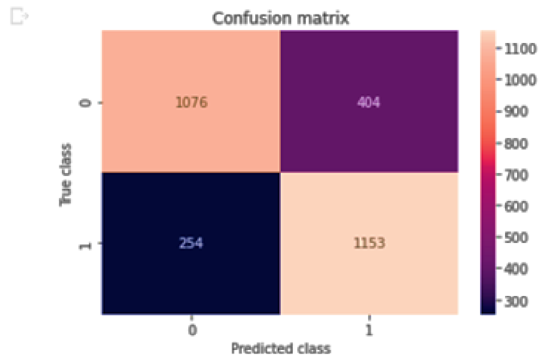
```
0.7784700588438906
```

Fig. 16 Prediction with new data.

As shown in Fig. 16, an accuracy of 77% was obtained with the new data. A more detailed view of the prediction is shown in Fig. 17, with a confusion matrix.

In Fig. 17, it is visualized through the heat map that 1076 true positives, 404 false positives, 254 false negatives and 1153 true negatives were achieved.

```
conf_matrix = confusion_matrix(y_test, y_predTest)
sns.heatmap(data=conf_matrix, annot=True, fmt="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
print (classification_report(y_test, y_predTest))
```




```
              precision    recall  f1-score   support

           0       0.81      0.73      0.77      1480
           1       0.74      0.82      0.78      1407

    accuracy                           0.77      2887
   macro avg       0.77      0.77      0.77      2887
weighted avg       0.78      0.77      0.77      2887
```

Fig. 17 Confusion matrix of results with new data.

6.2. Comparison between Deep Learning and Machine Learning

As you were able to witness, in the present work, we used one of the branches of artificial intelligence, which was Machine Learning, since the objective of the research work was to predict patients who could suffer a stroke. As you can see, a Machine Learning model has created a type of logistic regression continuation. In Table II, you will see a comparison between Deep Learning and Machine Learning.

It should be noted that better results were achieved with the applications of SMOTE techniques for data imbalance. In future work, we want to use more data records and higher quality that feed the model and thus improve the prediction. It should be noted that better results were achieved with the applications of SMOTE techniques for data imbalance. In future work, we want to use more data records and higher quality that feed the model and thus improve the prediction.

Table 2. Comparison between deep learning and machine learning

	Machine Learning	Deep Learning
Data format	Structured data	Unstructured data
Database	Manageable database data	More than one million data points
Training	It takes a human car	The system learns by itself
Algorithm	Variable algorithm	Neural Network Algorithms
Application	Simple Routine Tasks	Complex tasks

7. Conclusion

In conclusion, a machine learning model for stroke prediction was proposed. With the use of our model, an accuracy of 77% of our dataset was achieved. It should be noted that better results were achieved with the applications of SMOTE techniques for data imbalance. As we work in the future, we want to use more up-to-date, higher-quality data records to feed the model and thus improve prediction. I also recommend that more research be done using the different theories of artificial intelligence, as it has different branches. It is also suggested that new methodologies be sought for its application.

References

[1] Vieira P. M, Frizzo H. C. F, Machado M. P. R, da Silva Pires P, & Guimarães E. L, “Palliative Care in Stroke: A Nutritional Perspective Palliative Care in Stroke: A Nutritional Look Palliative Care in Stroke: A Nutritional Look IGOR Oliveira Loss1.

[2] Bernabé-Ortiz A, & Carrillo-Larco R. M, “Stroke Incidence Rate in Peru,” *Peruvian Journal of Experimental Medicine and Public Health*, vol. 38, pp. 399-405, 2021.

[3] Mariños E, Barreto-Acevedo E, & Espino P, “Ischemic Stroke Associated with COVID-19: First Case Report in Peru,” *Journal of Neuro-Psychiatry*, vol. 83, no. 2, pp. 127-133, 2020.

[4] H. Wu, S. Yang, Z. Huang, J. Él, y X. Wang, "Type 2 Diabetes Mellitus Prediction Model Based on Data Mining", *Informatics in Medicine Unlocked*, vol. 10, pp. 100–107, 2018.

[5] X. D. Zhang, "Machine Learning", *A Matrix Algebra Approach to Artificial Intelligence*, *Springer*, pp. 223-440, 2020.

[6] Srinivasan Suresh, "Prediction of Roadway Crashes using Logistic Regression in SAS," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 10, pp. 13-17, 2020. Crossref, https://doi.org/10.14445/23488387/IJCSE-V7I10P103

[7] S. Ray, K. Alshouiliy, A. Roy, A. AlGhamdi , D. P. Agrawal, "Chi-Squared Based Feature Selection for Stroke Prediction using Azureml", *2020 Intermountain Engineering, Technology and Computing (IETC)*, pp. 1–6, 2020.

[8] Vaibhav Gupta, Dr.Pallavi Murghai Goel, "Heart Disease Prediction Using ML," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 6, pp. 17-19, 2020. Crossref, https://doi.org/10.14445/23488387/IJCSE-V7I6P105

[9] A. E. Butler, S. R. Dargham, A. Abouseif, A. El Shewehy, and S. L. Atkin, “Vitamin D Deficiency Effects on Cardiovascular Parameters in Women with Polycystic Ovary Syndrome: A Retrospective, Cross-Sectional Study,” *Journal of Steroid Biochemistry and Molecular Biology*, vol. 211, 2021. [Online]. Available: www.scopus.com

[10] M. G. Worku, A. Annapoorani Anantharaman, "A Study of Logistic Regression and its Optimization Techniques Using Octave," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 10, pp. 23-28, 2019. Crossref, https://doi.org/10.14445/23488387/IJCSE-V6I10P105

[11] R. Das, S. K. Biswas, D. Devi y B. Sarma, "An Oversampling Technique by Integrating Reverse Nearest Neighbor in Smote: Reverse-Smote", *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1239-1244, 2020.

[12] Rajesh Kumar Prasad, "Prediction Model for DI Diesel Engine: Combustion," *SSRG International Journal of Mechanical Engineering*, vol. 8, no. 1, pp. 1-7, 2021. Crossref, https://doi.org/10.14445/23488360/IJME-V8I1P101

[13] C. Y. Hung, W. C. Chen, P. T. Lai, C. H. Lin, C. C. Lee, "Comparison of Deep Neural Networks and Other Machine Learning Algorithms for Stroke Prediction in a Large-Scale Population-Based Electronic Medical Claims Database", *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3110–3113, 2017.

[14] F. Soriano, “Stroke Prediction Dataset,” *Informatics in Medicine Unlocked*, 2021.

[15] M. G. Worku, A.B. Teshale, y G. A. Tesema, "Determinants of Under-Five Mortality in the High Mortality Regions of Ethiopia: Mixed-Effect Logistic Regression Analysis", *Archives of Public Health*, vol. 79, no. 1, 2021. [Online]. Available: www.scopus.com.