

# Avance Proyecto Final

Integrantes:

Sabrina Cárdenas 216306

Camilo Acosta 323454

# INTRODUCCIÓN

# **Descripción del Proyecto**

**Se utilizó una base de datos de un hospital público para desarrollar un modelo de predicción efectivo que pueda identificar a los pacientes con alta probabilidad de sufrir un accidente cerebrovascular.**

**El objetivo es abordar uno de los problemas de salud más graves a nivel mundial. En el proyecto previamente se realizó una limpieza de datos adecuada, se analizarán las variables predictivas y se implementarán algoritmos de predicción adecuados. Con su posterior optimización y evaluación.**

# PUNTO 5

## Algoritmos para Entrenamiento y Optimización de Hiperparámetros

# Optimización Hiperparámetros

¿Qué métrica usar para optimizar hiperparámetros?



# Optimización Hiperparametros

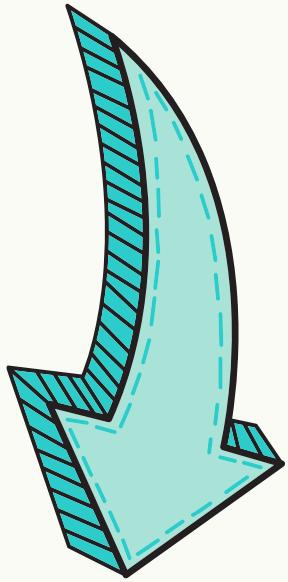
## Matriz de Confusión

1 POSITIVO Tendrá el accidente  
0 NEGATIVO No tendrá el accidente

		ACTUAL
		1
PREDICTED	1	TP
	0	FP
		0
		TN
		FN

# Optimización Hiperparámetros

Minimizar FN



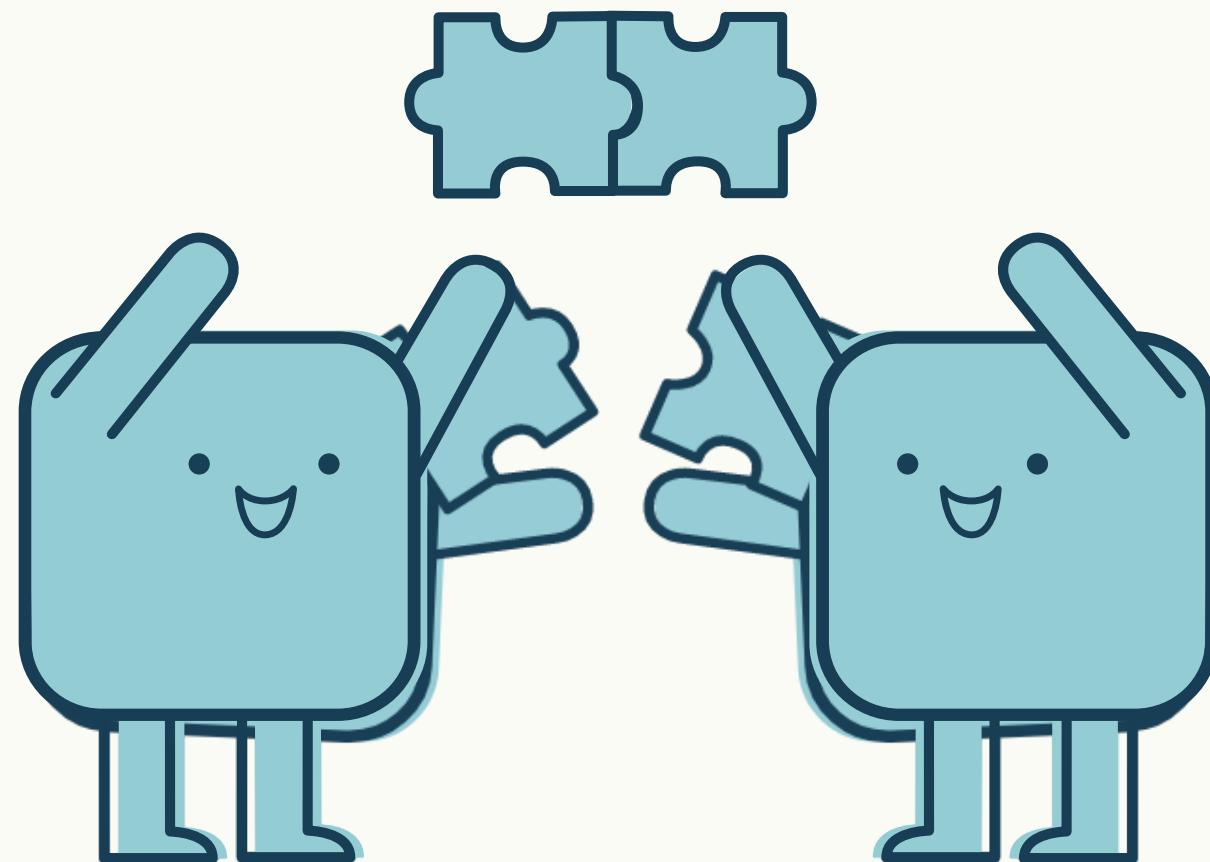
Sensibilidad



# Algoritmo Ensamble

Modelo 1 DecisionTreeClassifier

Modelo 2 RandomForestClassifier



- Combina múltiples árboles de decisión independientes para crear un modelo más robusto y preciso.
- Cada árbol se entrena en una muestra aleatoria de los datos de entrenamiento
- Reduce el sobreajuste

# Algoritmo Ensamble

## Hiperparametros

### Modelo 1 DecisionTreeClassifier

```
criterion='gini', splitter='best', max_depth=None,  
min_samples_split=2, min_samples_leaf=1,  
min_weight_fraction_leaf=0.0, max_features=None,  
random_state=None, max_leaf_nodes=None,  
min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0
```

Referencias:

Hornik, K. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.  
[https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

# Algoritmo Ensamble

## Hiperparametros

### Modelo 2 RandomForestClassifier

```
n_estimators=100, *, criterion='gini', max_depth=None,  
min_samples_split=2, min_samples_leaf=1,  
min_weight_fraction_leaf=0.0, max_features='sqrt',  
max_leaf_nodes=None, min_impurity_decrease=0.0,  
bootstrap=True, oob_score=False, n_jobs=None,  
random_state=None, verbose=0, warm_start=False,  
class_weight=None, ccp_alpha=0.0, max_samples=None
```

Referencias:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn-ensemble-randomforestclassifier>

# Algoritmo Ensamble

## Hiperparametros

1. **criterion='gini'**: Criterio para medir la impureza de un nodo. Menos sensible a los cambios en las probabilidades de clase
2. **El parámetro splitter**, que determina la estrategia de selección del atributo en cada nodo, tiene como valor 'best'
3. **El parámetro max\_depth** controla la profundidad máxima del árbol. Cuando se establece en None, el árbol se expande hasta que todas las hojas sean puras

Referencias:

Hornik, K. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.  
[https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

# Algoritmo No visto en Clase: GradientBoostingClassifier

- Resolver problemas de clasificación
- Ensambaje: árboles de decisión débiles para crear un modelo fuerte y más preciso.
- Cada árbol intenta corregir los errores del árbol anterior



# Algoritmo No visto en Clase: GradientBoostingClassifier

## Hiperparametros

```
loss='log_loss', learning_rate=0.1, n_estimators=100, subsample=1.0,  
criterion='friedman_mse', min_samples_split=2,  
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3,  
min_impurity_decrease=0.0, init=None, random_state=None,  
max_features=None, verbose=0, max_leaf_nodes=None,  
warm_start=False, validation_fraction=0.1, n_iter_no_change=None,  
tol=0.0001, ccp_alpha=0.0
```

Referencias:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn-ensemble-gradientboostingclassifier>

# Algoritmo No visto en Clase: GradientBoostingClassifier Hiperparametros

**learning\_rate: 0.1** El parámetro learning\_rate controla la tasa de aprendizaje en el proceso de potenciación del gradiente . 0.1 indica una tasa de aprendizaje moderada.

**n\_estimators** Cantidad de árboles de decisión en el modelo de Gradient Boosting. El valor es 100 árboles, Un numero mas alto da sobreajuste.

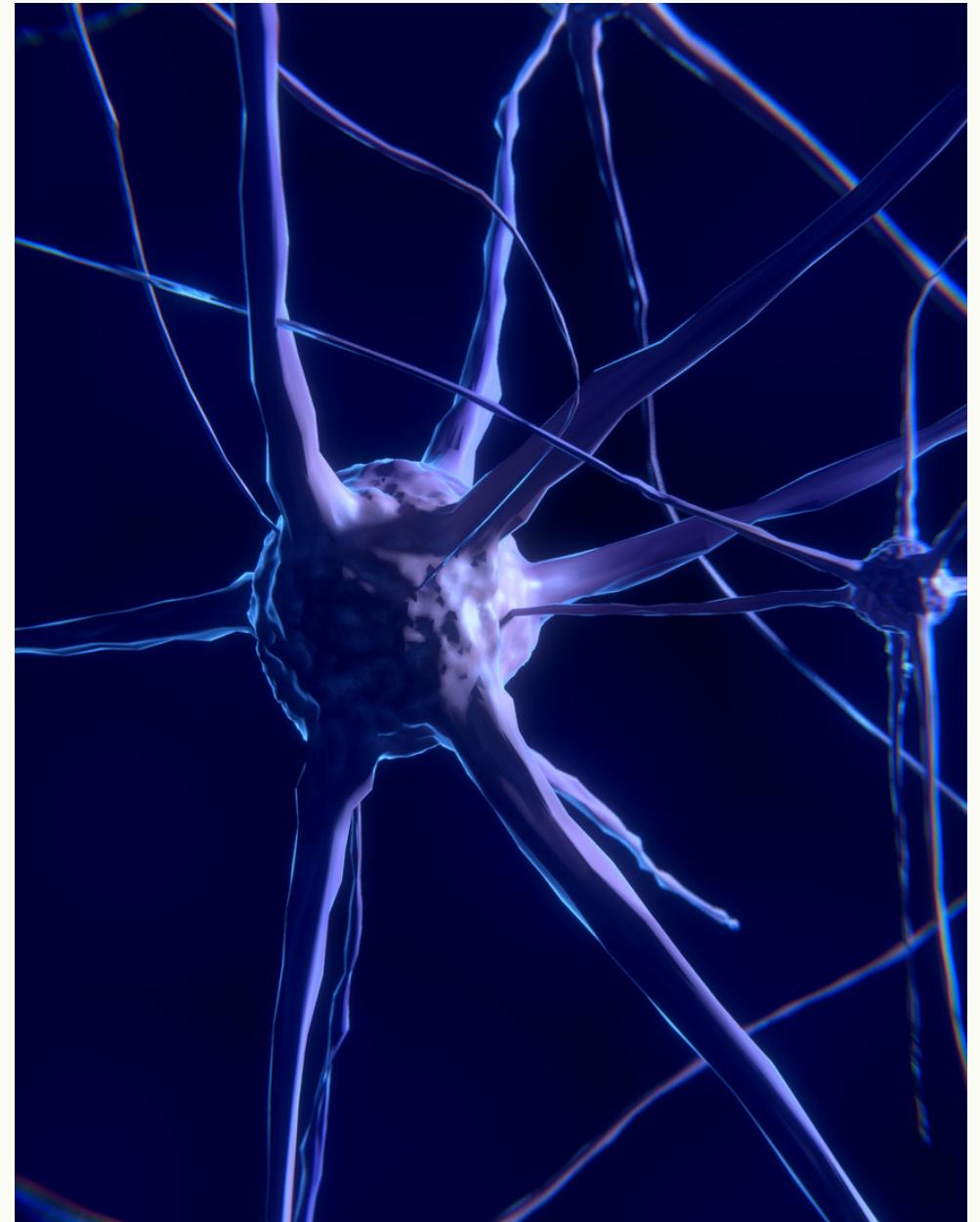
**El parámetro subsample** controla la proporción de muestras utilizadas para entrenar cada árbol en el proceso de Gradient Boosting. 1.0, permite que cada árbol tenga acceso a toda la información en el conjunto de entrenamiento.

Referencias:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn-ensemble-gradientboostingclassifier>

# Redes Neuronales Artificiales (ANN)

- Un modelo de perceptrón multicapa (MLP, por sus siglas en inglés)
- Resolver problemas de clasificación y regresión.
- Cada capa está conectada a la siguiente.
- Recibe una combinación lineal de las salidas de las neuronas en la capa anterior, función de activación.



Referencias:

Hornik, K. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.  
[https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)

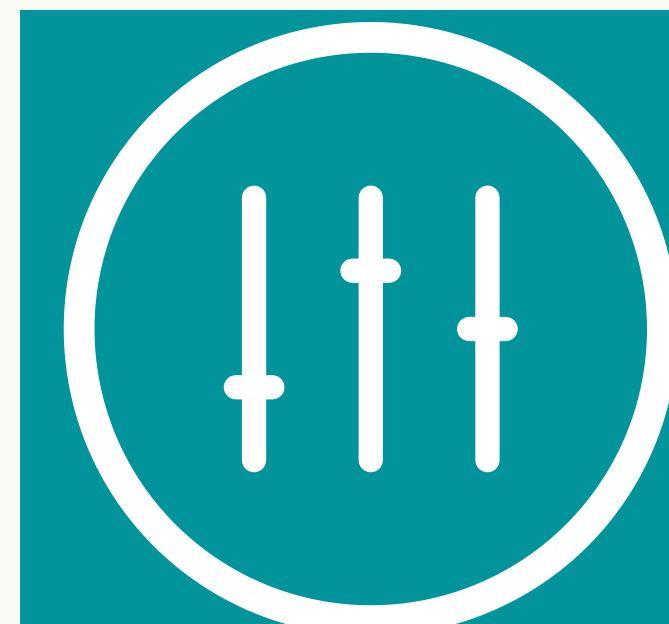
# Redes Neuronales Artificiales (ANN)

## ¿Qué hiperparámetros tiene?

- 'activation'
- 'alpha'
- 'batch\_size'
- 'beta\_1'
- 'beta\_2'
- 'early\_stopping'
- 'epsilon'
- 'hidden\_layer\_sizes'
- 'learning\_rate'
- 'learning\_rate\_init'
- 'max\_fun'
- 'max\_iter'
- 'momentum'
- 'n\_iter\_no\_change'
- 'nesterovs\_momentum'
- 'power\_t'
- 'random\_state'
- 'shuffle'
- 'shuffle':
- 'solver':
- 'tol':
- 'validation\_fraction':
- 'verbose':
- 'warm\_start':

Referencias:

Artificial\_Neural\_Network\_Hyperparameters\_Optimiza.pdf



# Redes Neuronales Artificiales (ANN)

## ¿Cuáles optimizar?

**Artificial Neural Network Hyperparameters Optimization: A Survey**  
<https://doi.org/10.3991/ijoe.v18i15.34399>



### Referencia

- [73] H. Harafani, I. Suryani, Ispandi, and N. Lutfiyana, "Neural network parameters optimization with genetic algorithm to improve liver disease estimation," J. Phys. Conf. Ser., vol. 1641, no. 1, 2020, <https://doi.org/10.1088/1742-6596/1641/1/012034>
- [77] P. Kumar, S. Batra, and B. Raman, "Deep neural network hyper-parameter tuning through twofold genetic approach," Soft Comput., vol. 25, no. 13, pp. 8747–8771, 2021, <https://doi.org/10.1007/s00500-021-05770-w>
- [82] P. Kaur, A. Singh, and I. Chana, "BSense: A parallel Bayesian hyperparameter optimized Stacked ensemble model for breast cancer survival prediction," J. Comput. Sci., vol. 60, p. 101570, 2022, <https://doi.org/10.1016/j.jocs.2022.101570>

# Redes Neuronales Artificiales (ANN)

¿Qué soluciona?

- [73]
  - Mínimos Locales
- [77]
  - Overfitting
- [82]
  - Heterogeneidad DB



# Redes Neuronales Artificiales (ANN)

## Estado del Arte

- [73]
  - Learning rate 'learning\_rate'
  - Momentum Coeficient 'momentum'
- [77]
- Activation function 'activation'
- Number of nodes 'hidden\_layer\_sizes'
- Number of layers 'hidden\_layer\_sizes'
- [77]
- Network Optimizer 'solver'
- [82]
- Hiden Layers 'hidden\_layer\_sizes'
- Epochs 'max\_iter'
- Number of iterations 'max\_iter'



# Redes Neuronales Artificiales (ANN)

## ¿Qué hiperparámetros tiene?

- 'activation'
- 'alpha'
- 'batch\_size'
- 'beta\_1'
- 'beta\_2'
- 'early\_stopping'
- 'epsilon'
- 'hidden\_layer\_sizes'
- 'learning\_rate'
- 'learning\_rate\_init'
- 'max\_fun'
- 'max\_iter'
- 'momentum'
- 'n\_iter\_no\_change'
- 'nesterovs\_momentum'
- 'power\_t'
- 'random\_state'
- 'shuffle'
- 'shuffle':
- 'solver'
- 'tol':
- 'validation\_fraction':
- 'verbose':
- 'warm\_start':

Referencias:

Artificial\_Neural\_Network\_Hyperparameters\_Optimiza.pdf

# Redes Neuronales Artificiales (ANN)

## Hiperparametros a Optimizar

- [73]
  - 'learning\_rate\_init' (0.01, 0.003)
  - 'momentum' (0.7, 0.8)
- [77]
  - 'activation' (tanh, relu)
  - Number of nodes 'hidden\_layer\_sizes' (64, 128) -> (16,32)
  - Network Optimizer 'solver' (adam, sdg)



# Optimización de Hiperparámetros

## GridSearchCV

- Búsqueda exhaustiva en una cuadrícula de hiperparámetros
- Entrena y evalúa un modelo para cada combinación de valores de los hiperparámetros en la cuadrícula
- Revisa rendimiento en términos de una métrica de evaluación específica **RECALL**



# Selección de Hiperparámetros

## Según Bibliografía

- **El parámetro alpha** controla el término de regularización L2. El valor predeterminado es 0.0001, pequeña de regularización. Un valor demasiado alto = modelo subajustado y no permite que el modelo capture patrones en los datos.
- **El parámetro max\_fun** controla el número máximo de llamadas a la función. El valor predeterminado es 15000, Un valor más alto de max\_fun permite mejorar la precisión del modelo al permitir una convergencia más completa hacia una solución óptima
- **shuffle: True.** Controla si las muestras se barajan después de cada iteración. Mejora convergencia y evita mínimos locales

- Referencia: Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research

# Redes Neuronales Artificiales (ANN)

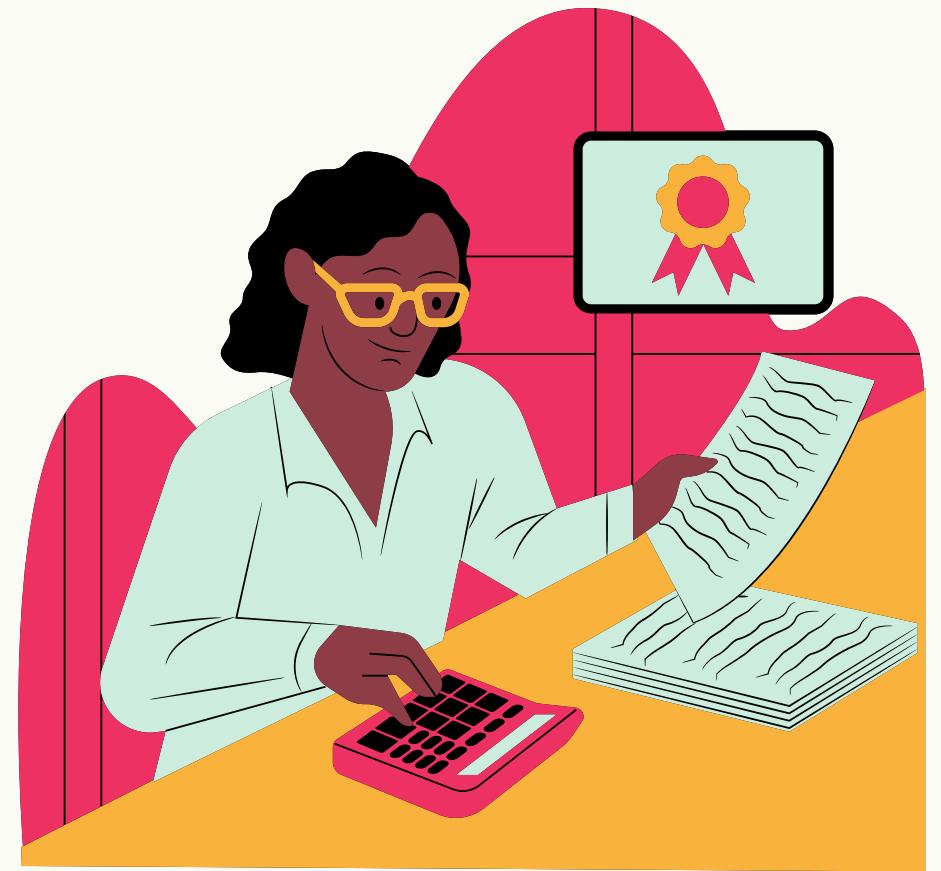
## Resultados

- 'learning\_rate\_init' (0.003)
- 'momentum' (0.7)
- 'activation' (tanh)
- Number of nodes 'hidden\_layer\_sizes' (32)
- Network Optimizer 'solver' (adam)



# Regresión Logística

- Utiliza una función logística para predecir la probabilidad de pertenecer a una clase.
- Ajusta un modelo lineal a los datos y transforma la salida mediante la función logística.

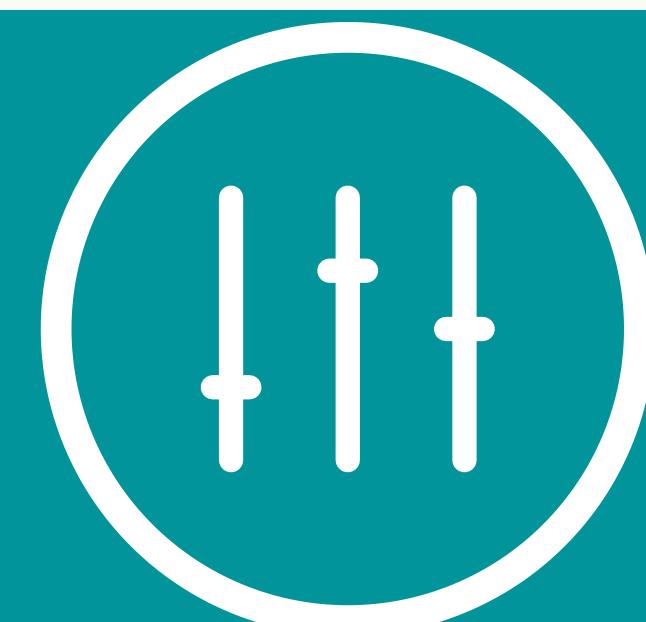


# Regresión Logística

¿Qué hiperparámetros tiene?

```
penalty='l2', *, dual=False, tol=0.0001, C=1.0,  
fit_intercept=True, intercept_scaling=1, class_weight=None,  
random_state=None, solver='lbfgs', max_iter=100,  
multi_class='auto', verbose=0, warm_start=False,  
n_jobs=None, l1_ratio=None
```

Referencias:  
[Ahttps://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)



# Regresión Logistica

¿Cuáles optimizar?

Logistic Regression Hyperparameter Optimization for Cancer Classification.

[https://www.researchgate.net/publication/358043825\\_Logistic\\_Regression\\_Hyperparameter\\_Optimization\\_for\\_Cancer\\_Classification](https://www.researchgate.net/publication/358043825_Logistic_Regression_Hyperparameter_Optimization_for_Cancer_Classification)



# Regresión Logistica

## Estado del Arte

- Tipo de Penalidad 'penalty' ( $l_1, l_2$ )
- $l_1\_ratio$  ' $l_1\_ratio$ ' (0,1)
- Alpha 'C' (0.00001, 0,01)



# Regresión Logistica

## Hiperparametros Según Bibliografia

**El parámetro random\_state** controla la semilla utilizada para la generación de números aleatorios El valor predeterminado es 42, Establecer una semilla específica permite obtener resultados reproducibles en diferentes ejecuciones del modelo

**El parámetro solver** Optimiza FO. El valor predeterminado es 'lbfgs' (Limited-memory Broyden-Fletcher-Goldfarb-Shanno), aproximaciones de segundo orden para actualizar los parámetros del modelo.

Referencias:

Nocedal, J., & Wright, S. (2006). Numerical Optimization. Springer Series in Operations Research and Financial Engineering

# Regresión Logistica

## Resultados

- Tipo de Penalidad 'penalty' ( $l_2$ )
- $l_1\_ratio$  ' $l_1\_ratio$ ' (0)
- Alpha 'C' (0,01)



**Nota:** Por definición si sale  $l_2$  sale 0



# PUNTO 6

## Evaluacion de los modelos

# Ensamble

## Matriz de Confusión



Validation		
	Prediction	
Actual	0	1
0	686	24
1	23	2

Test		
	Prediction	
Actual	0	1
0	670	31
1	29	5

Train		
	Prediction	
Actual	0	1
0	3279	0
1	0	32

# Ensamble

## Métricas

	Train	Test	Val
• Sensibilidad	<b>1.0</b>	<b>0.1470</b>	<b>0.08</b>
• Especificidad	<b>1.0</b>	<b>0.9557</b>	<b>0.9661</b>
• Exactitud	<b>1.0</b>	<b>0.9183</b>	<b>0.9360</b>
• Precisión	<b>1.0</b>	<b>0.1388</b>	<b>0.0769</b>
• AUC	<b>1.0</b>	<b>0.8127</b>	<b>0.7276</b>

# Ensamble

## Analisis AUC-ROC

- AUC ROC > 0.5 tiene habilidad de clasificación.
- AUC ROC > 0.7 a 0.8 se considera aceptable,
- AUC ROC de 0.8 a 0.9 se considera bueno.
- AUC ROC superior a 0.9 se considera excelente.
- AUC ROC del modelo está por debajo de la línea diagonal (0.5), esto indica que el modelo está haciendo peor que una predicción aleatoria.

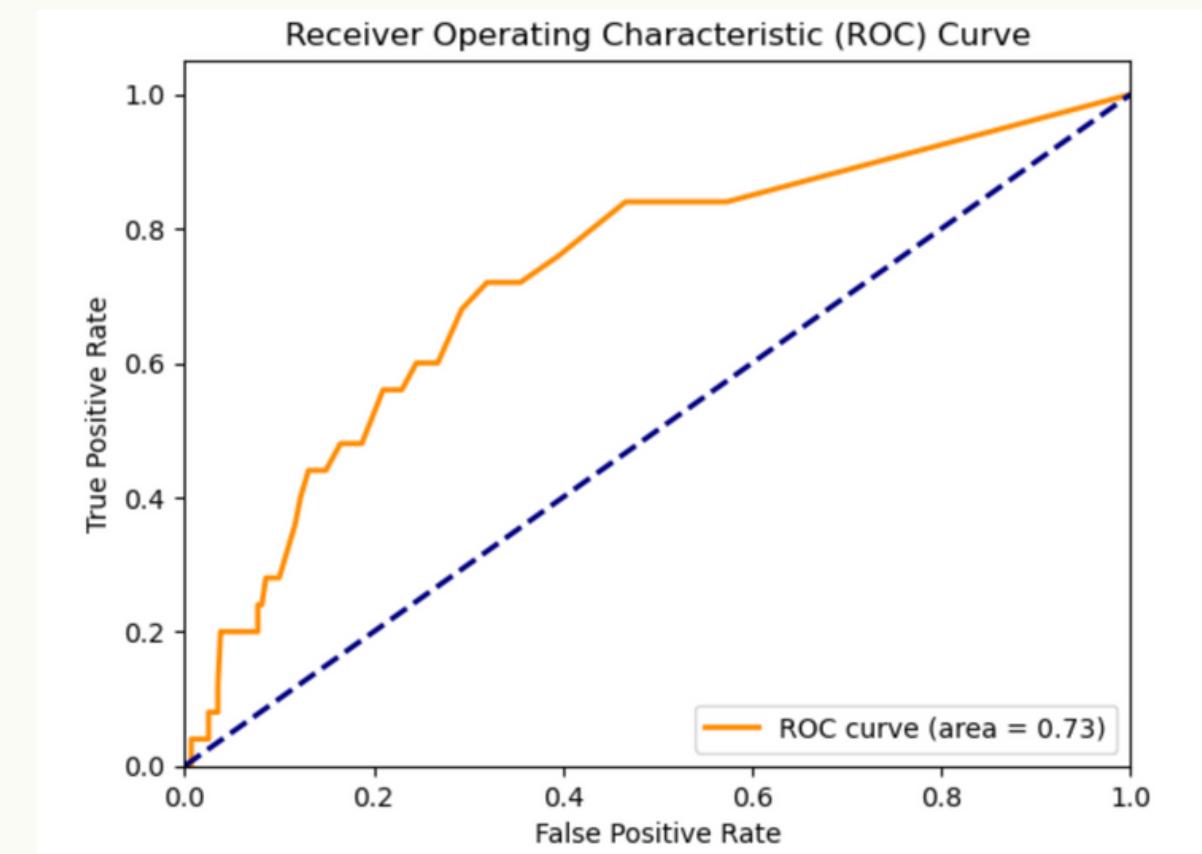
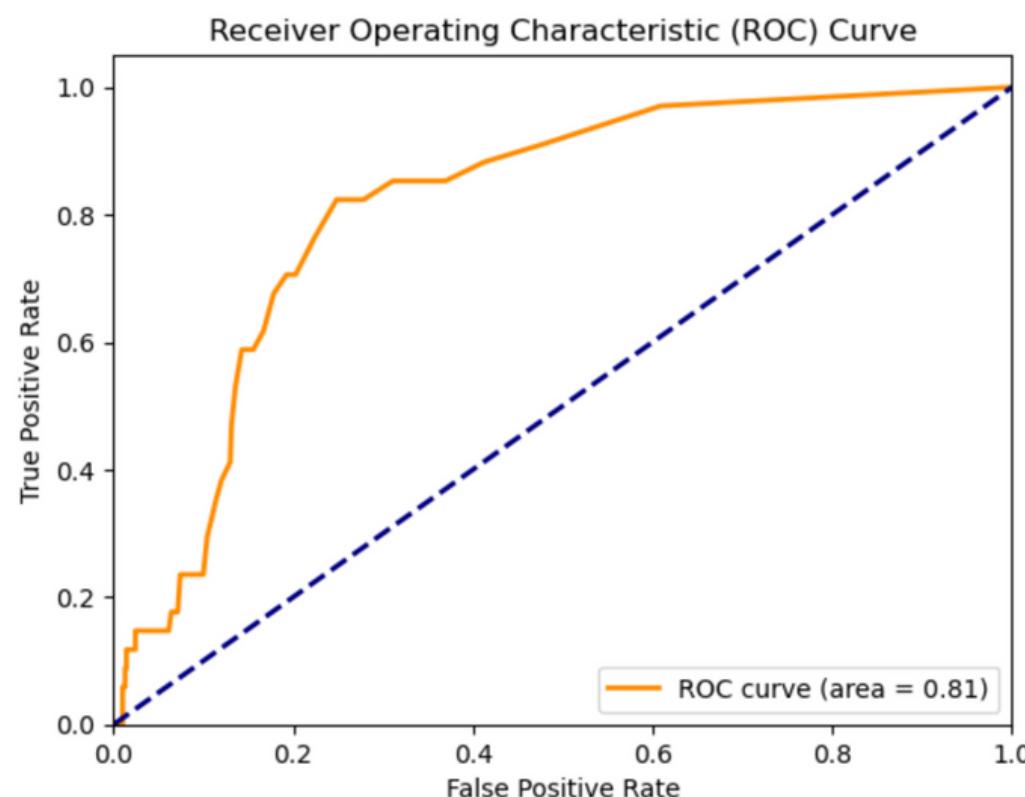
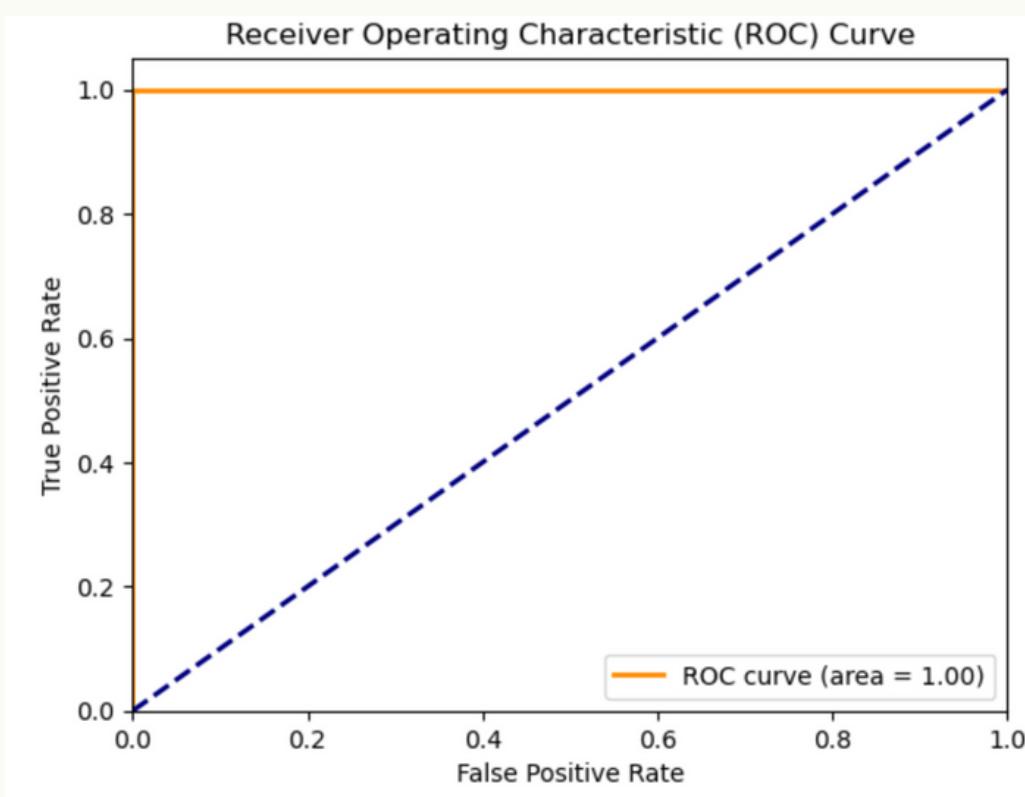
# Ensamble

ROC

Train 1

Test 0.81

Val 0.73



# GradientBoostingClassifier

## Matriz de Confusión



Validation		
	Prediction	
Actual	0	1
0	578	132
1	12	13

Test		
	Prediction	
Actual	0	1
0	570	131
1	11	23

Train		
	Prediction	
Actual	0	1
0	2732	547
1	81	3198

# GradientBoostingClassifier

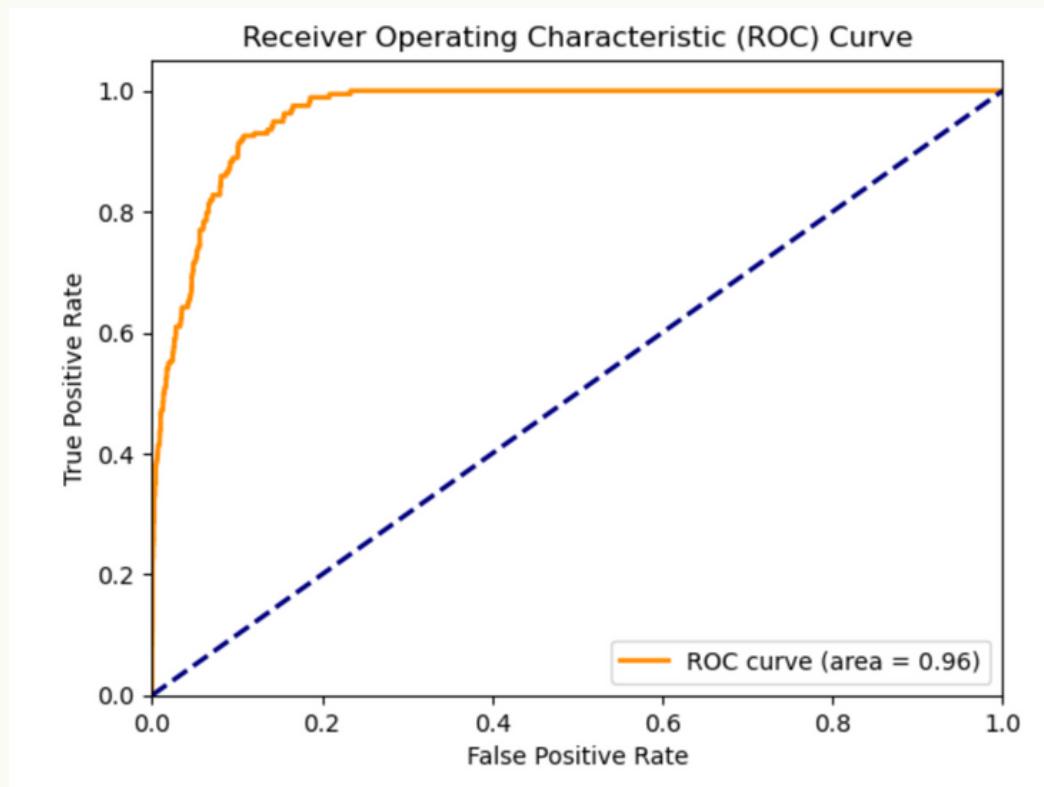
## Métricas

	Train	Test	Val
• Sensibilidad	<b>0.9752</b>	<b>0.6764</b>	<b>0.8140</b>
• Especificidad	<b>0.8331</b>	<b>0.8131</b>	<b>0.52</b>
• Exactitud	<b>0.9042</b>	<b>0.8068</b>	<b>0.8040</b>
• Precisión	<b>0.8539</b>	<b>0.1493</b>	<b>0.0896</b>
• AUC	<b>0.9633</b>	<b>0.8505</b>	<b>0.7494</b>

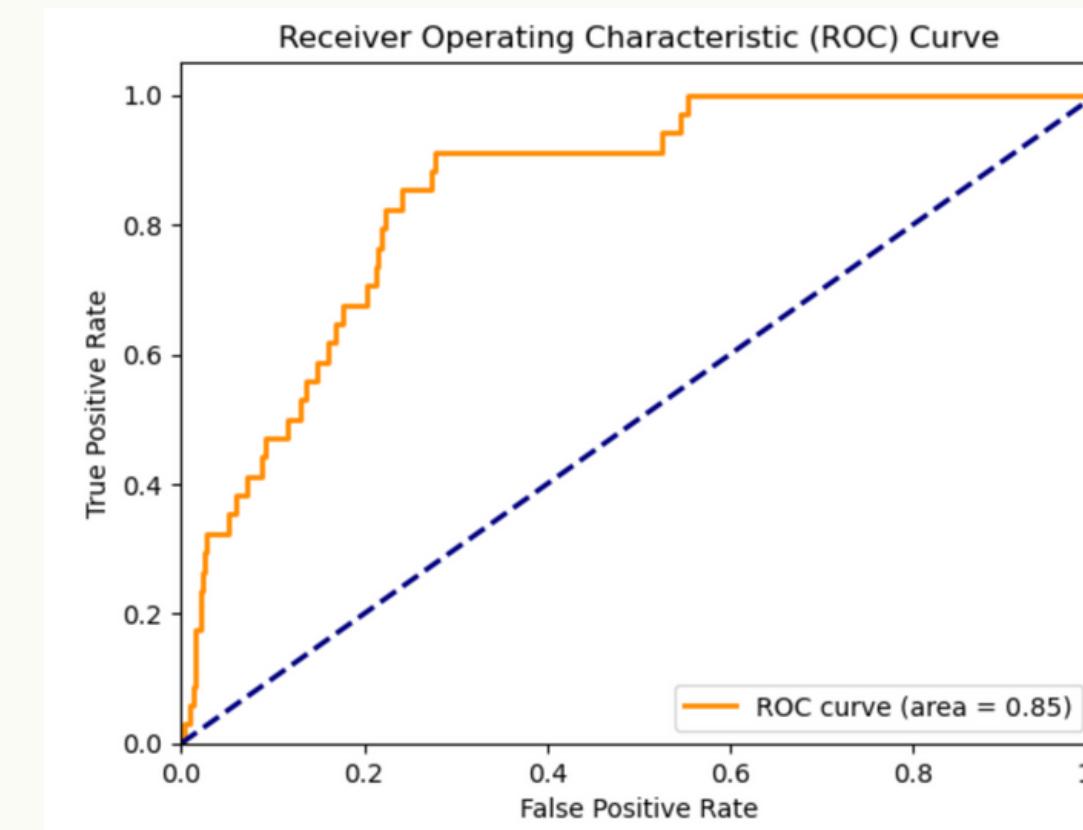
# GradientBoostingClassifier

ROC

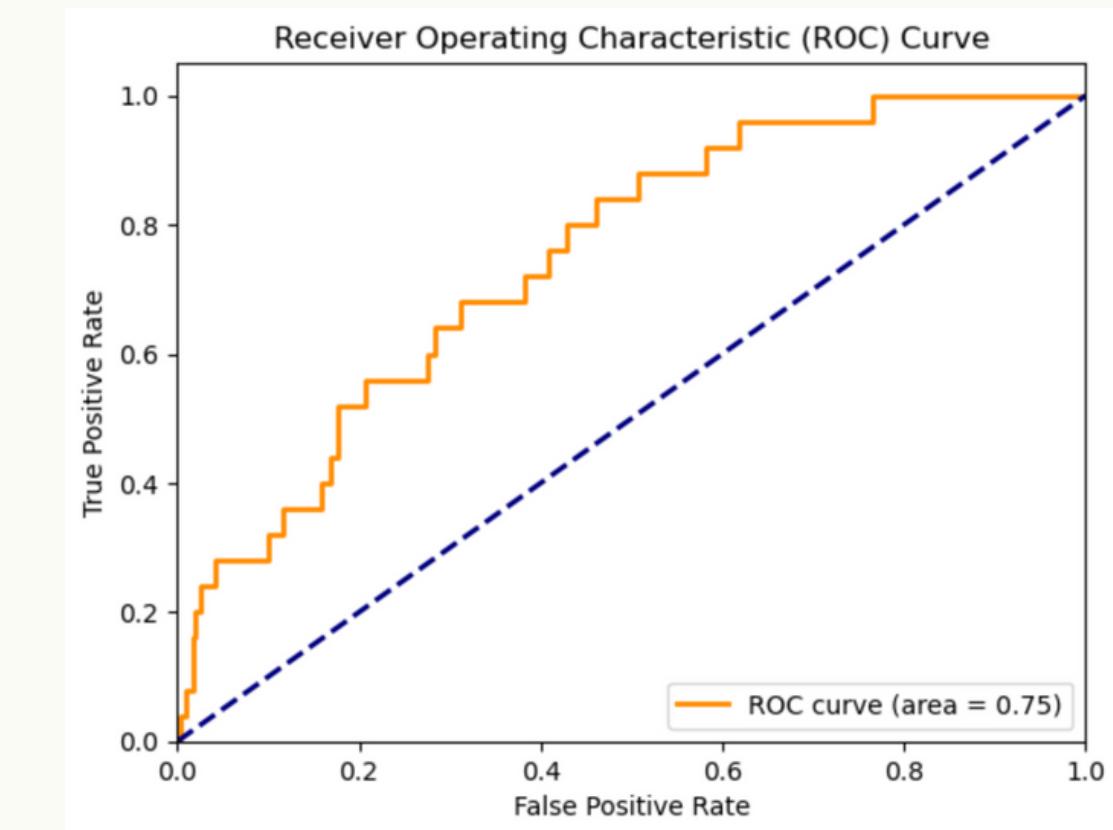
Train 0.96



Test 0.85



Val 0.73



# Redes Neuronales

## Matriz de Confusión



Validation		
	Prediction	
Actual	0	1
0	603	107
1	18	7

Test		
	Prediction	
Actual	0	1
0	590	111
1	16	18

Train		
	Prediction	
Actual	0	1
0	2846	433
1	163	3116

# Redes Neuronales

## Métricas

	Train	Test	Val
• Sensibilidad	0.9502	0.5294	0.28
• Especificidad	0.8679	0.8416	0.8492
• Exactitud	0.9091	0.8272	0.8299
• Precisión	0.8779	0.1395	0.0614
• AUC	0.9547	0.7791	0.7597

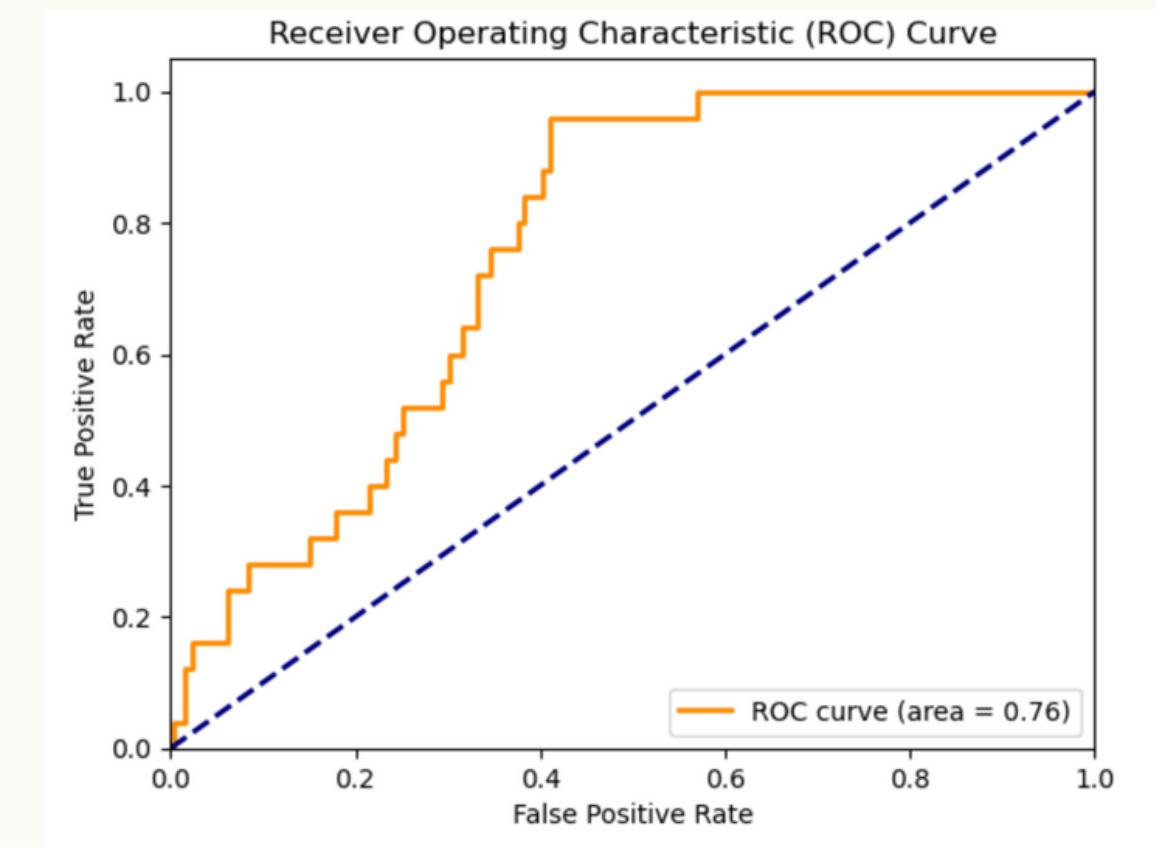
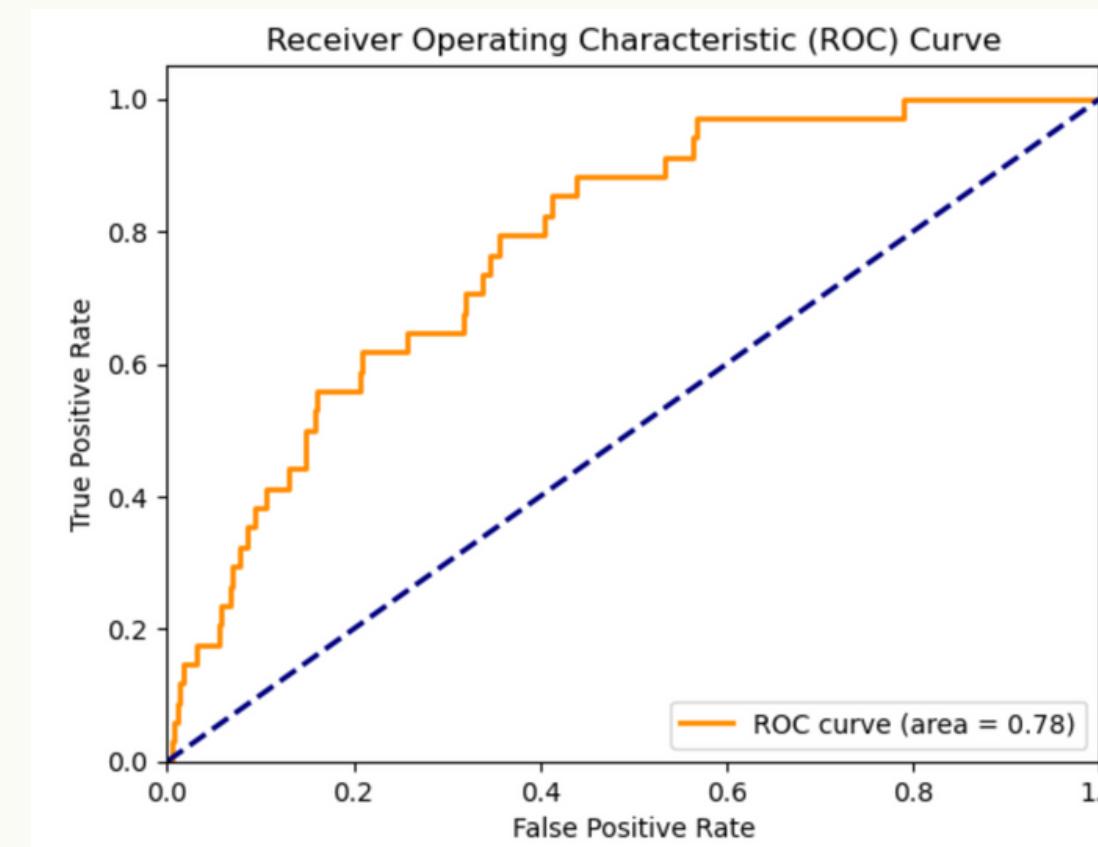
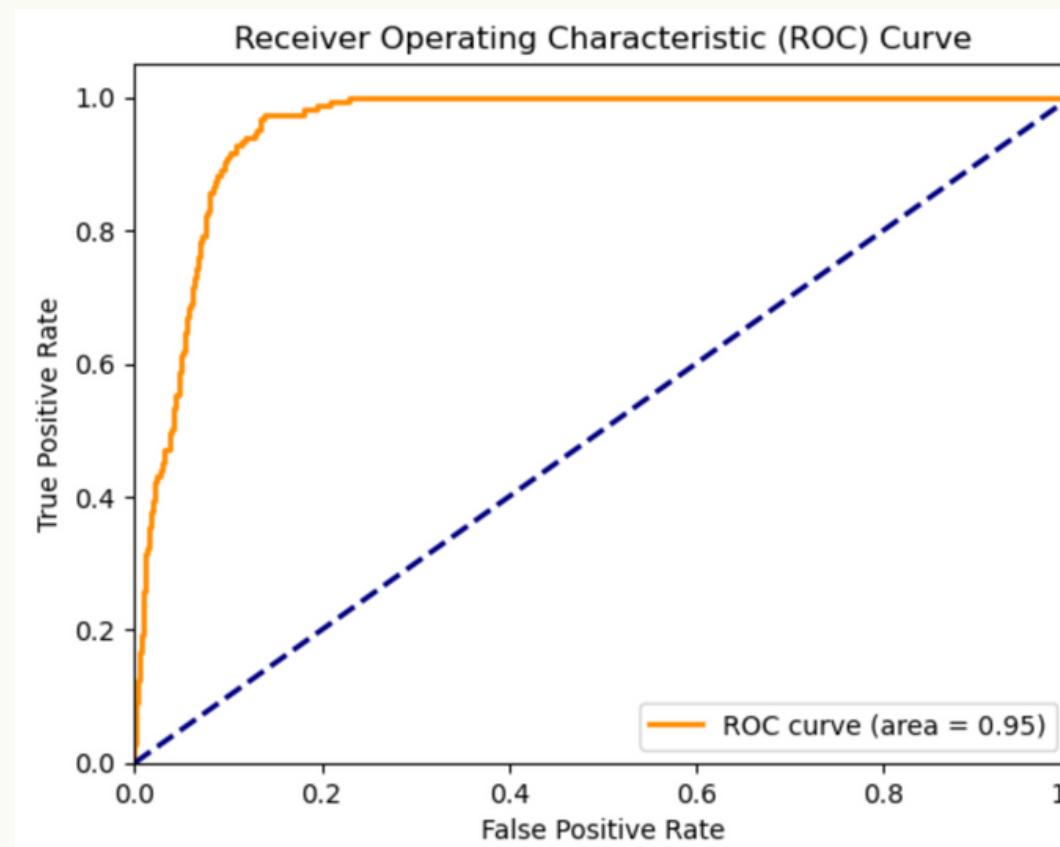
# Redes Neuronales

ROC

Train 0.95

Test 0.78

Val 0.76



# Regresión Logistica

## Matriz de Confusión



Validation		
	Prediction	
Actual	0	1
0	524	186
1	7	18



Test		
	Prediction	
Actual	0	1
0	511	190
1	5	29



Train		
	Prediction	
Actual	0	1
0	2408	871
1	553	2726



# Regresión Logistica

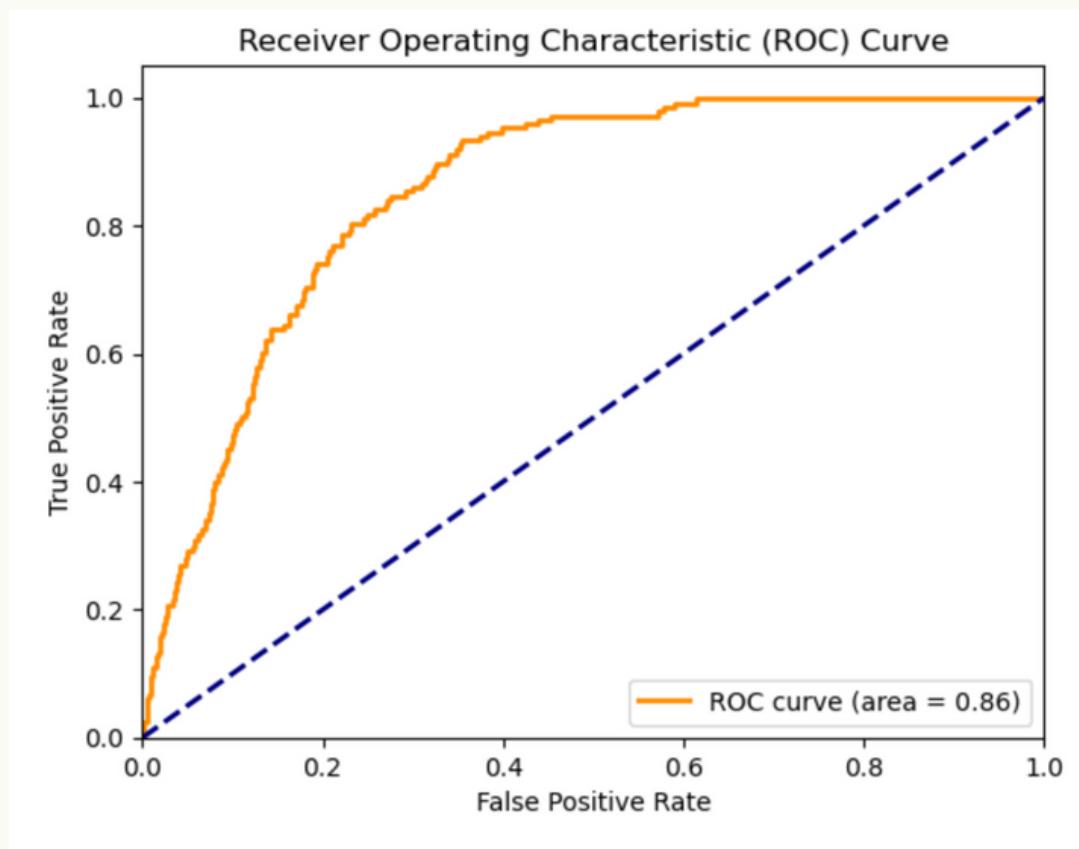
## Métricas

	Train	Test	Val	
• Sensibilidad	0.8313	0.8529	0.72	✓
• Especificidad	0.7343	0.7289	0.7380	
• Exactitud	0.7828	0.7346	0.7374	
• Precisión	0.7578	0.1324	0.0882	
• AUC	0.8550	0.8526	0.8032	

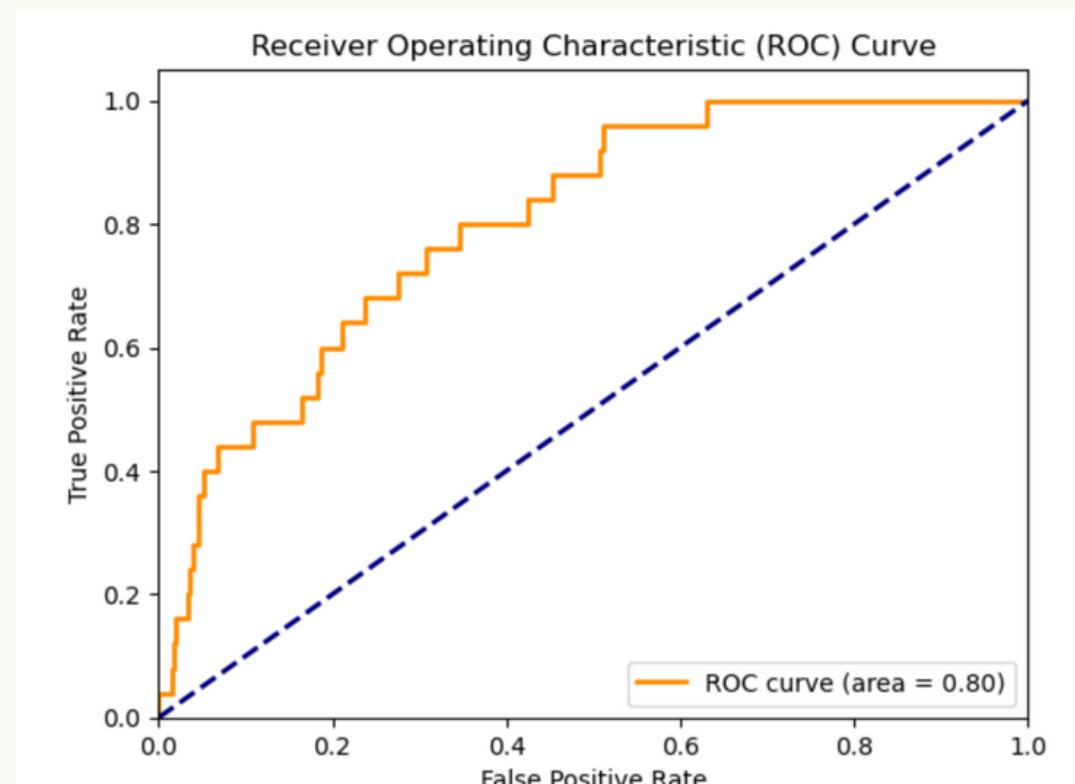
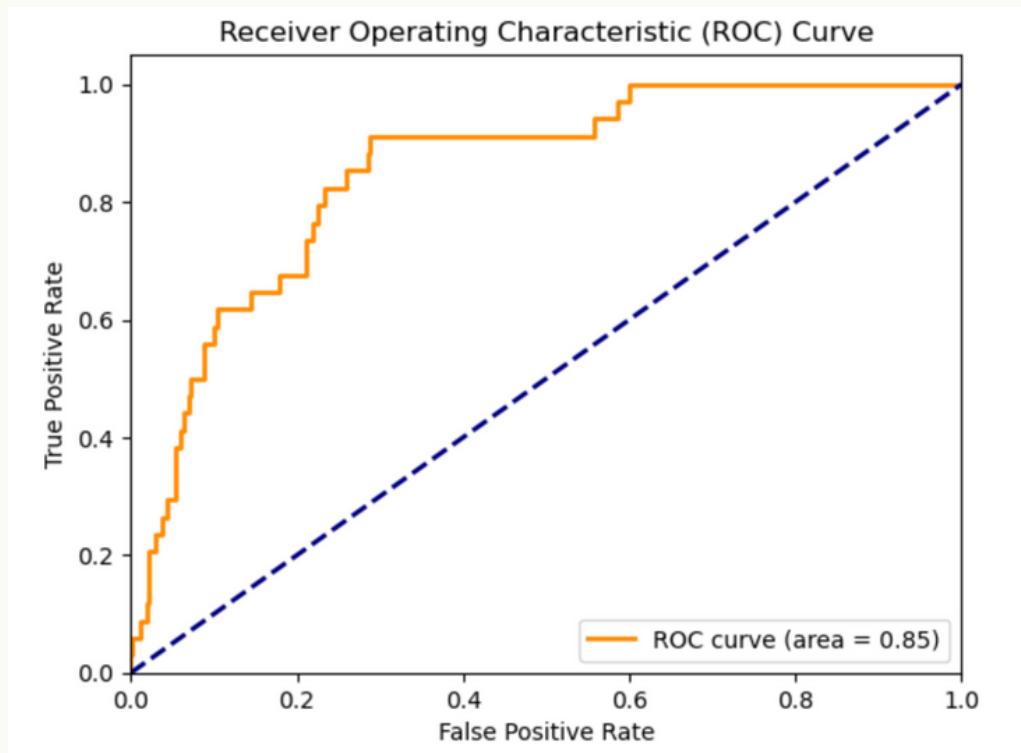
# Regresión Logistica

ROC

Train 0.86



Test 0.85 ✓



# La métrica de evaluación más adecuada

## RECALL SENSIBILIDAD



# El mejor algoritmo

- Regresión Logística
- Es aquel donde se tiene mayor Sensibilidad

Train 0.8529

Val 0.7200

- Menos Falsos Negativos

Train 5

Val 7



# Random forest para seleccionar las variables predictivas

`model.feature_importances_`

## ¿Qué es una Importancia?

Se refiere a la capacidad para explicar la variabilidad en la variable objetivo.



# Random forest para seleccionar las variables predictivas

## Variable

- Age
- Avg\_glucose\_level
- Bmi
- Hypertension
- Married



# PUNTO 7

## Calcule el costo de implementación del modelo

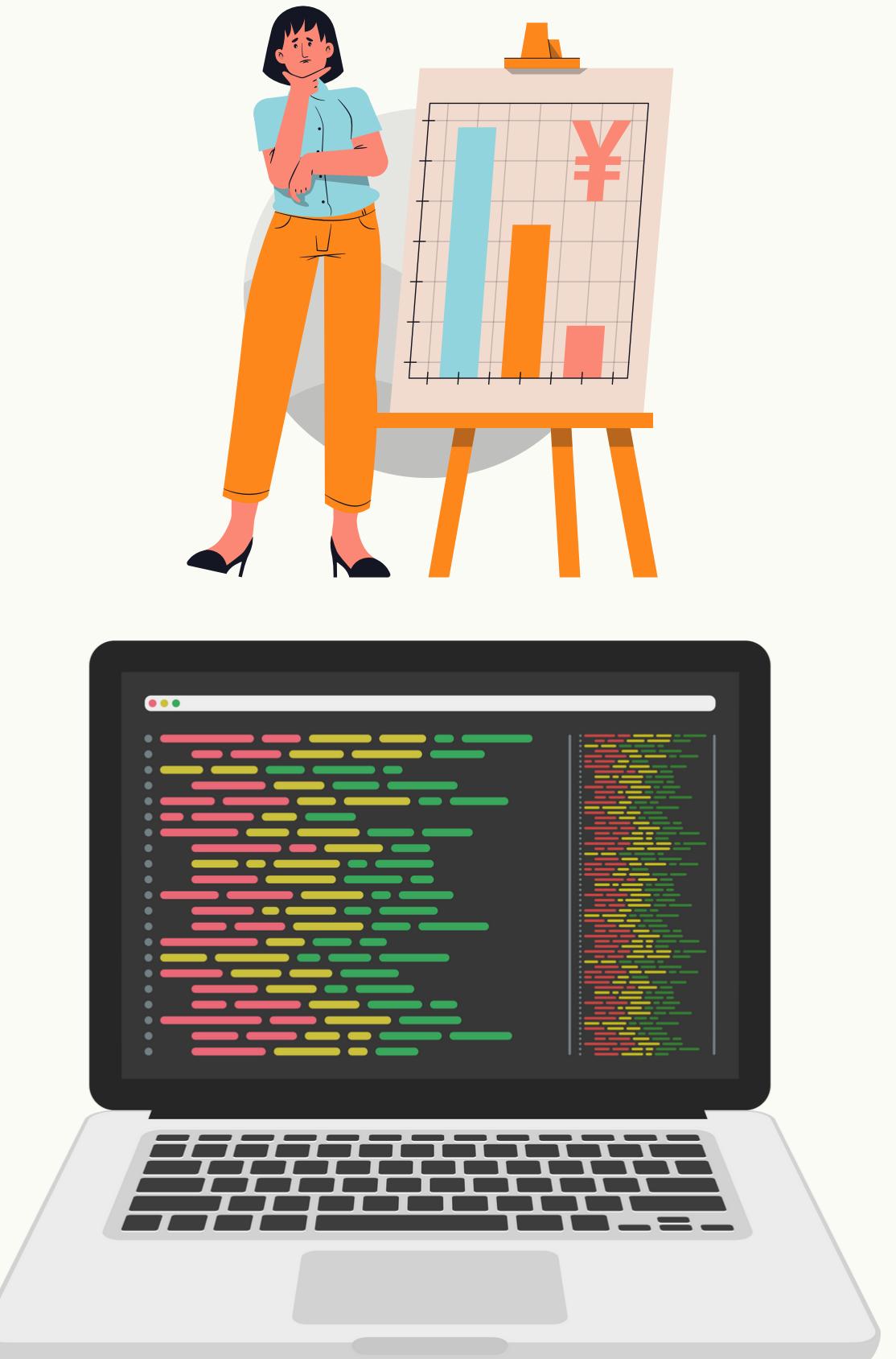
# Importancia de la implementación

- Disponibilidad de datos consistentes.
- Diagnósticos precisos y toma de decisiones informadas.
- Mejora en la planificación de los recursos.
- Actualización de políticas de salud de atención médica en general.



# Costos asociados a Empresas

- **Personal:** Desarrollo y mantenimiento del modelo.
  - Analista de datos: \$1434.
- **Software:** Visualización, manejo de bases de datos y software de análisis estadístico.
  - Python: código abierto.
- **Hardware:** Soportar el modelo y los datos.
  - Computadora: \$1500.



# Costos asociados a Hospitales

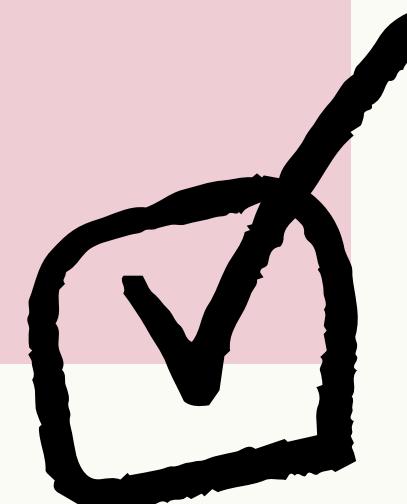
- Infraestructura de red: Tamaño y velocidad de transferencia de la red.
  - Básico: \$2000.
  - Tamaño mediano: \$500000.
- Almacenamiento: Capacidad requerida y tipo de almacenamiento utilizado.
  - Nube: \$0.03 a \$0.15 por GB al mes.



# Costos asociados a Hospitales

## Ventajas

- Mejora la toma de decisiones clínicas y administrativas.
- Detección temprana de enfermedades.
- Optimización de recursos
- Implementación temprana trae grandes beneficios.



## Desventajas

- Grandes costos de inversión a corto plazo.
- Cambio en la estructura administrativa de información.
- Necesidad de tiempo para acostumbrarse a entender el funcionamiento del modelo.



# PUNTO 8

## GitHub

# GitHub

<https://github.com/acosmilo/ProyectoFinalAnalitica>

The screenshot shows a GitHub repository page for 'acosmilo / ProyectoFinalAnalitica'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. The commit history shows an initial commit from 'acosmilo' 5 days ago. The 'About' section contains a description in Spanish: 'Este es el código del proyecto final de la clase de Analítica de Datos en la USFQ.' It also lists files like 'LICENSE' and 'README.md', and metrics such as 0 stars, 1 watching, and 0 forks.

acosmilo / **ProyectoFinalAnalitica** Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code About

acosmilo Initial commit 71889f3 5 days ago 1 commit

LICENSE Initial commit 5 days ago

README.md Initial commit 5 days ago

README.md

**ProyectoFinalAnalitica**

Este es el código del proyecto final de la clase de Analítica de Datos en la USFQ.

About

Este es el código del proyecto final de la clase de Analítica de Datos en la USFQ.

Readme

MPL-2.0 license

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

# CONCLUSIONES

# Conclusiones

- Importancia de métrica Recall,
- Regresión logistica, enfocandonos en maximizar el RECALL, como resultado se obtuvo los mejores valores en esta metrica con estos dos modelos en comparación a otros modelos.
- Sabiendo los valores de AUC ROC se puede afirmar que todos los modelos tanto para el set de validación como el de prueba tiene habilidad de clasificación.

# REFERENCIAS

- <https://repositorio.utp.edu.co/server/api/core/bitstreams/767003ef-6a5a-4b19-8d13-0c3a25b8f128/content>
- [https://repositorio.uam.es/bitstream/handle/10486/697900/abella\\_miravet\\_blanca\\_tfg.pdf?sequence=1](https://repositorio.uam.es/bitstream/handle/10486/697900/abella_miravet_blanca_tfg.pdf?sequence=1)
- [https://repositorio.uam.es/bitstream/handle/10486/697900/abella\\_miravet\\_blanca\\_tfg.pdf?sequence=1](https://repositorio.uam.es/bitstream/handle/10486/697900/abella_miravet_blanca_tfg.pdf?sequence=1)
- [https://gateway.ipfs.io/ipfs/bafykbzacebj6eleeilrlkjh7dlqn2cmb5yurhfqktb24dubkhwmdvmtypftxu?filename=Sebastian%20Raschka%20-%20Python%20Machine%20Learning-Packt%20Publishing%20\(2015\).pdf](https://gateway.ipfs.io/ipfs/bafykbzacebj6eleeilrlkjh7dlqn2cmb5yurhfqktb24dubkhwmdvmtypftxu?filename=Sebastian%20Raschka%20-%20Python%20Machine%20Learning-Packt%20Publishing%20(2015).pdf)
- <https://www.mayoclinic.org/es-es/diseases-conditions/prediabetes/diagnosis-treatment/drc-20355284>
- <https://www.medigraphic.com/pdfs/conapeme/pm-2012/pm124g.pdf>

- Artificial\_Neural\_Network\_Hyperparameters\_Optimiza.pdf
- <https://iopscience.iop.org/article/10.1088/1742-6596/1641/1/012034/pdf>
- <https://sci-hub.se/10.1007/s00500-021-05770-w>
- <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn-tree-decisiontreeclassifier>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn-ensemble-randomforestclassifier>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn-ensemble-gradientboostingclassifier>
- [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)

- [https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)
- [https://www.researchgate.net/publication/358043825\\_Logistic\\_Regression\\_Hyperparameter\\_Optimization\\_for\\_Cancer\\_Classification](https://www.researchgate.net/publication/358043825_Logistic_Regression_Hyperparameter_Optimization_for_Cancer_Classification)

**GRACIAS**