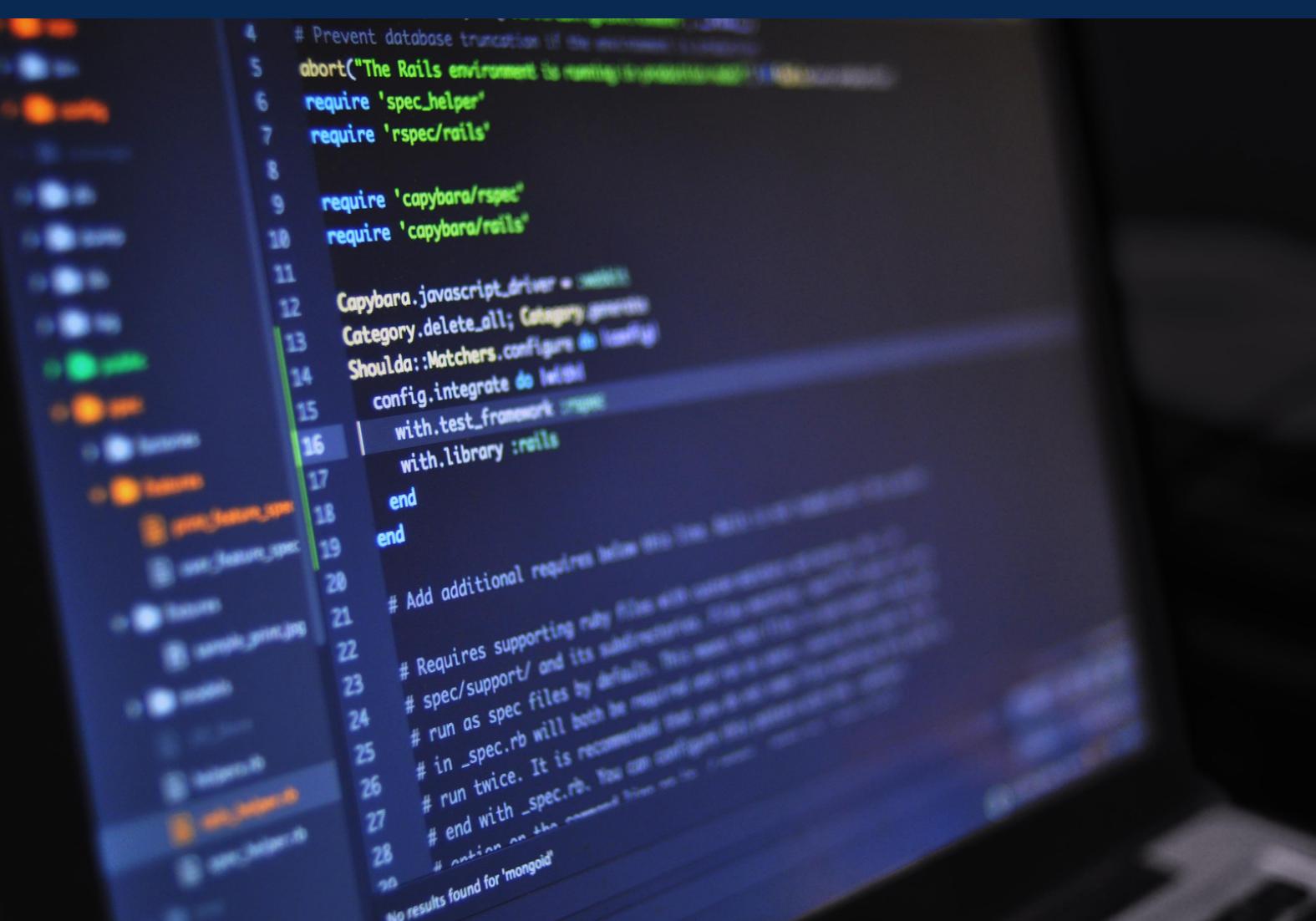


MODELO ANTI-PHISHING

DANIEL CARRERA
PROYECTO MACHINE LEARNING

INDICE DE CONTENIDOS



```
4 # Prevent database truncation if the environment is test.
5 abort("The Rails environment is running in production mode!
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create!(name: "Default")
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line to append them to all models in the application below.
22 # Requires supporting ruby files with custom matchers and helpers with:
23 # require 'path/to/matchers.rb'
24 # run as spec/ and its subdirectories. This can be used to share common fixtures
25 # in _spec.rb will both be required after the parent require statement above.
26 # run twice. It is recommended you do this in your spec/spec_helper.rb file
27 # end with _spec.rb. You can configure the parent require statement by adding
28 # require 'spec_helper' to the bottom of your _spec.rb
29
30 # no results found for 'mongoid'
```

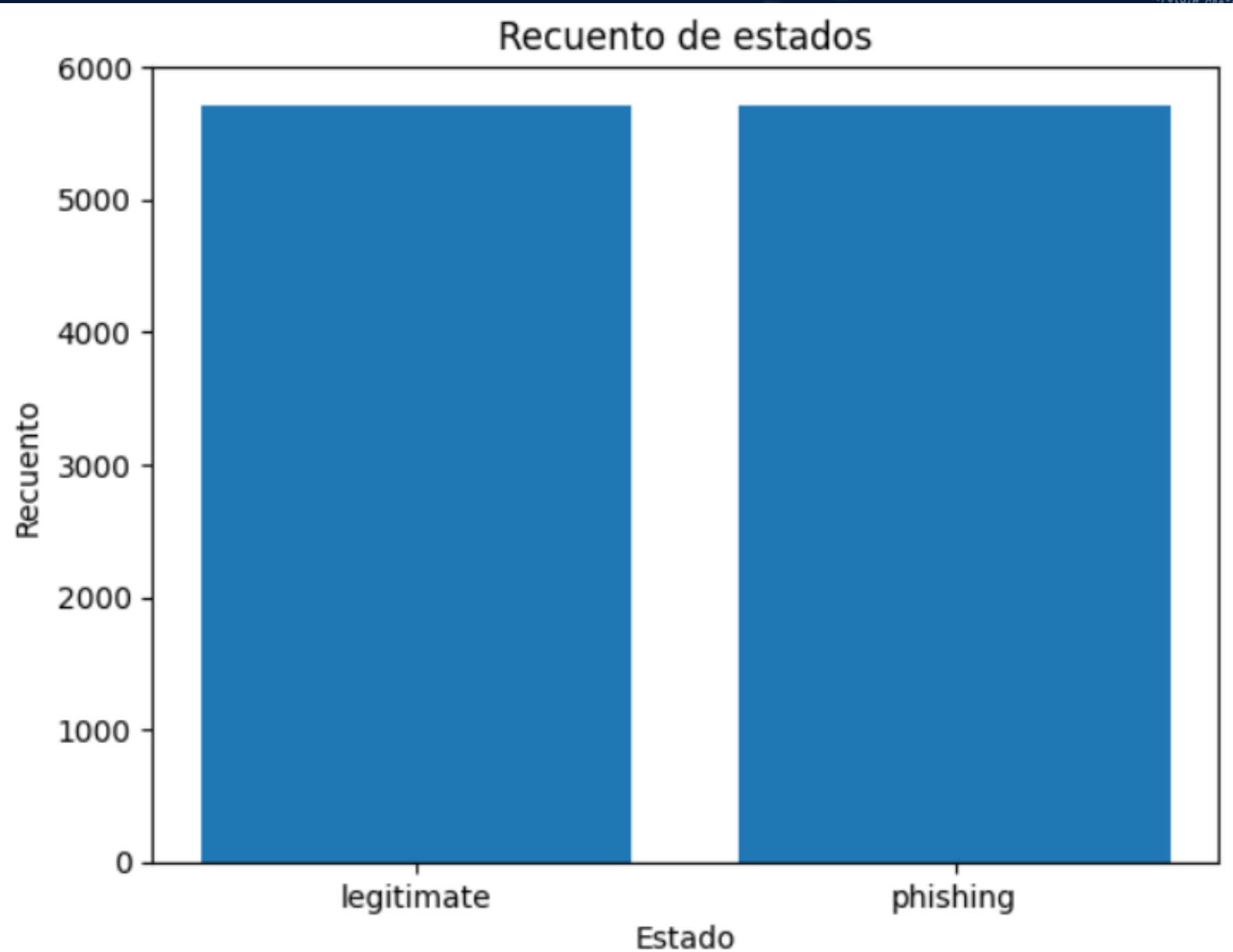
1. ANÁLISIS DEL DATASET
2. COMPARACIÓN DE MODELOS
3. APLICACIÓN DEL MODELO ELEGIDO
4. PROYECTO A FUTURO

MODELO ANTI-PHISHING

1/ ANÁLISIS DEL DATASET

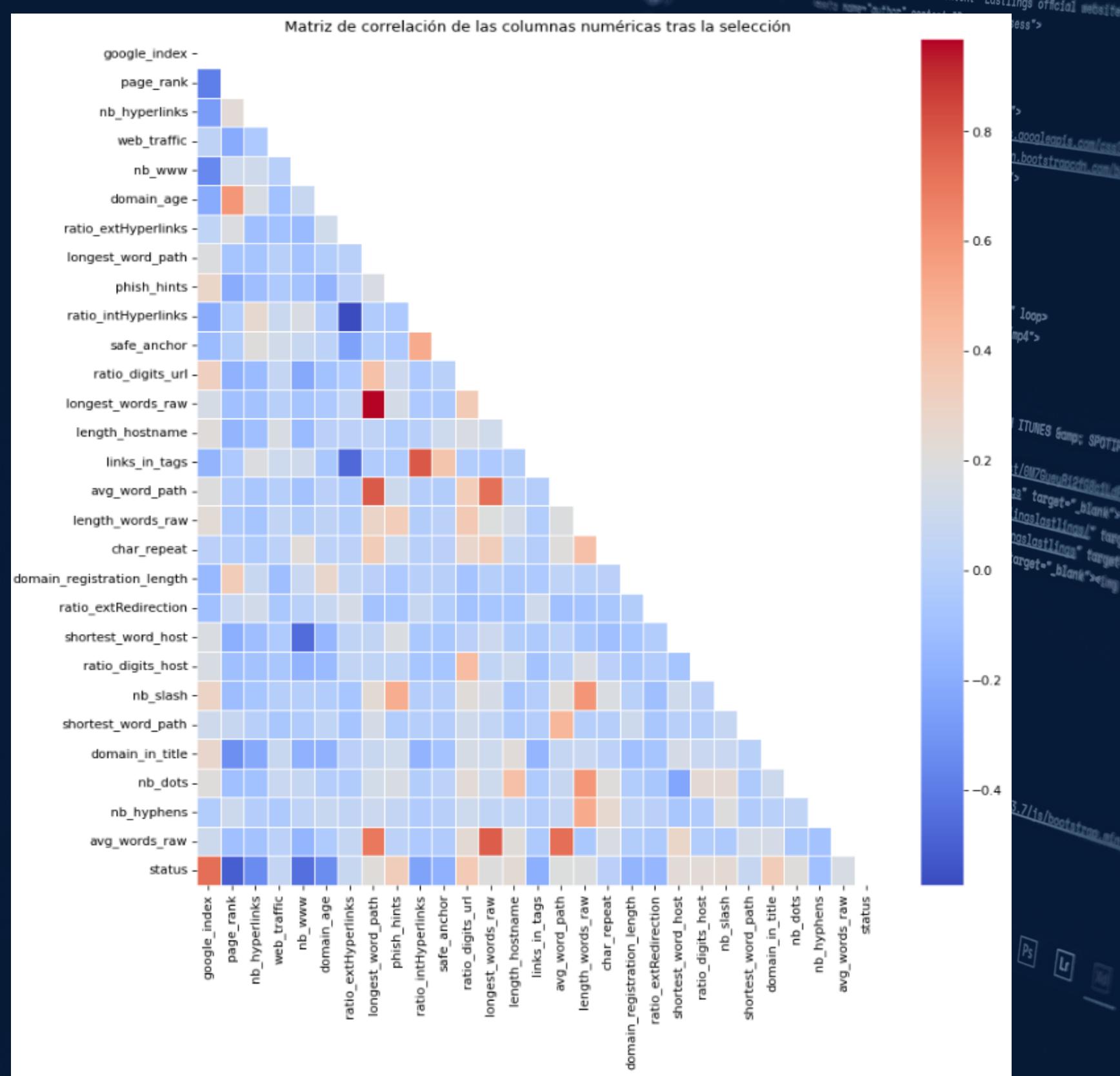
1/ ANÁLISIS DEL DATASET

- 11430 url del dataframe
- Sin presencia de NaN.
- 89 columnas, 2 categóricas y 87 numericas.
- Cabe destacar que en la vida real, hay mucha más proporción de enlaces que son legítimos que enlaces que son fishing, por lo que, a pesar de que para nuestro modelo, el hecho de que estén equilibrados es bastante bueno, puede que este un poco desvirtualizado.
- La columna objetivo será la de 'Status'.



1/ ANÁLISIS DEL DATASET

- - Cuanto mayor sea la longitud de la url, más probable es que el enlace sea falso.
- - Cuanto mayor sea la longitud promedio de las palabras de la URL sin caracteres especiales, más probable de que el enlace sea falso.
- - Cuanto mayor sea la longitud promedio de las palabras en la ruta de la URL sin caracteres especiales, más probable de que el enlace sea falso.
- - Cuanto mayor sea el número de hipervínculos de la página web, aumentan exponencialmente las posibilidades de que sea un enlace legítimo.
- - Cuanto mayor sea el número de enlaces en las etiquetas HTML de la página web, más probable que sea legítimo.
- - Cuanto mayor sea la proporción de archivos multimedia internos en la página web, más probable que sea legítimo.
- - Cuanto mayor sea la proporción de archivos multimedia externos en la página web, más probable que sea legítimo.
- - Si los enlaces de la página web utilizan atributos "rel" para evitar la apertura de nuevas pestañas, más probable que sea legítimo.





2/ COMPARACIÓN DE MODELOS

1 / LINEAL
REGRESSION

2 / KNN
CLASSIFIER

3 / DECISION
TREE
CLASSIFIER

4 / SUPPORT
VECTOR
CLASSIFIER

5 / BAGGING
CLASSIFIERS

6 / RANDOM
FOREST
CLASSIFIER

7 / ADABOOST
CLASSIFIER

8 / GRADIENT
BOOSTING
CLASSIFIER

7 / XGBOOST CLASSIFIER

2 / COMPARACIÓN DE MODELOS

- Recall Score
- GridSearchCV para la optimización de hiperparámetros.
- Bagging con LR, KNN y SVC.
- AdaBoost con DTC.

Modelo	Parámetros óptimos	Score (Recall)
LogisticRegression	max_iter=10000, penalty='l2', C=0.1, random_state=42	0.708450
KNeighborsClassifier	n_neighbors=7, p=0.1, weights='distance'	0.886614
DecisionTreeClassifier	criterion='entropy', max_depth=None, max_features='auto', min_samples_leaf=1, min_samples_split=2, random_state=42	0.922373
SVC	C=1, gamma='scale', kernel='rbf', random_state=42	0.885740
BaggingClassifier_LR	base_estimator=LogisticRegression(max_iter=10000, penalty='l2', C=0.1, random_state=42), max_features=0.5, max_samples=0.7, n_estimators=10	0.811814
BaggingClassifier_KNN	base_estimator=KNeighborsClassifier(n_neighbors=7, p=1, weights='distance'), max_features=0.5, max_samples=0.7, n_estimators=10	0.901873
BaggingClassifier_SVC	base_estimator=SVC(C=1, gamma='scale', kernel='rbf', random_state=42), max_features=0.5, max_samples=0.5, n_estimators=10	0.887267
RandomForestClassifier	max_depth=None, max_features='log2', min_samples_leaf=1, min_samples_split=2, n_estimators=200, random_state=42	0.961185
AdaBoostClassifier	base_estimator=DecisionTreeClassifier(max_depth=5, min_samples_leaf=1, min_samples_split=2, random_state=42), learning_rate=0.1, n_estimators=200	0.950720
GradientBoostingClassifier	learning_rate=0.5, max_depth=None, max_features='log2', min_samples_leaf=1, min_samples_split=2, n_estimators=200	0.962929
XGBClassifier	colsample_bytree=0.5, learning_rate=0.1, max_depth=None, min_child_weight=1, n_estimators=200, reg_alpha=0.1, reg_lambda=0.1, subsample=0.9	0.965328

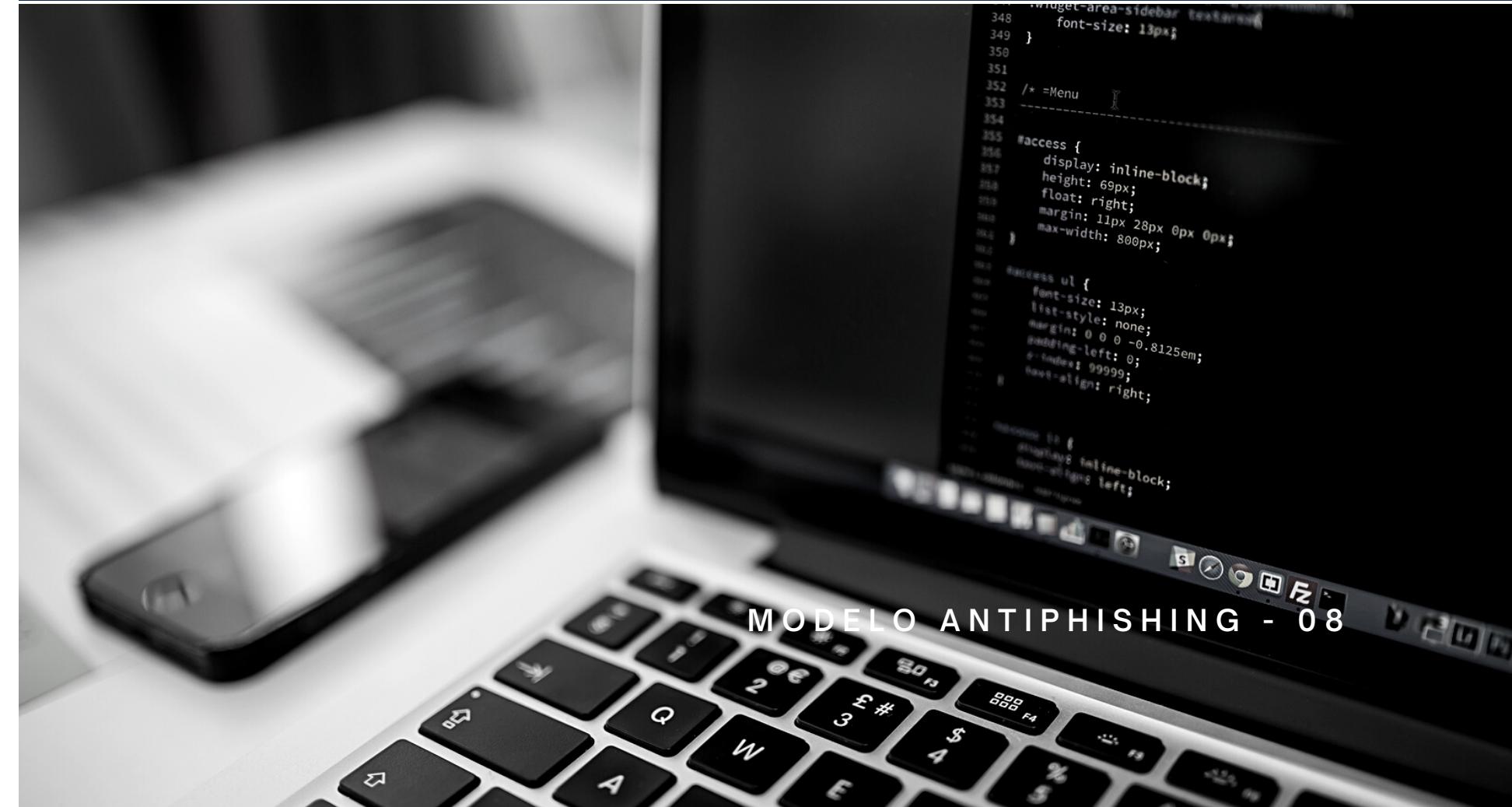
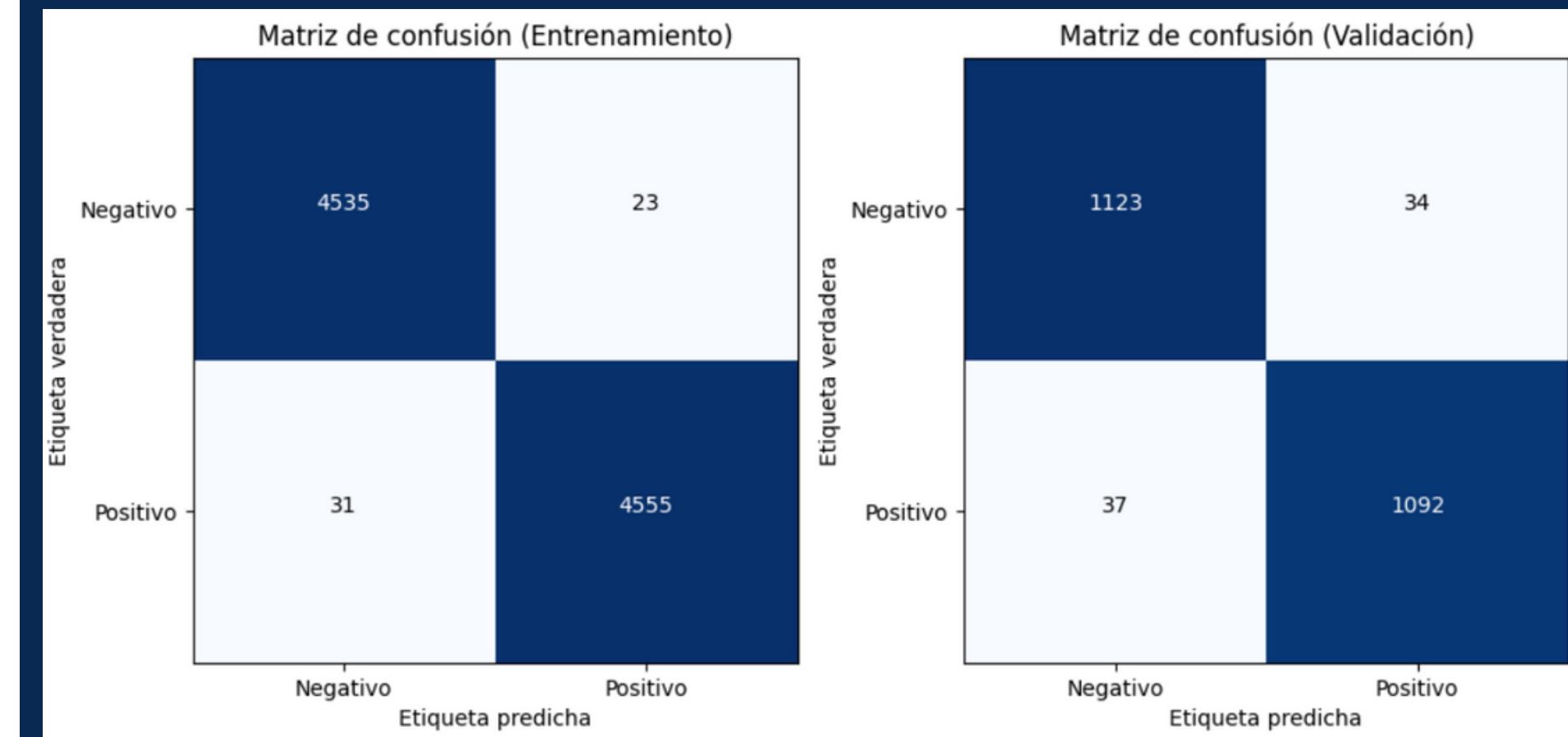


3 / APLICACIÓN DEL MODELO ELEGIDO

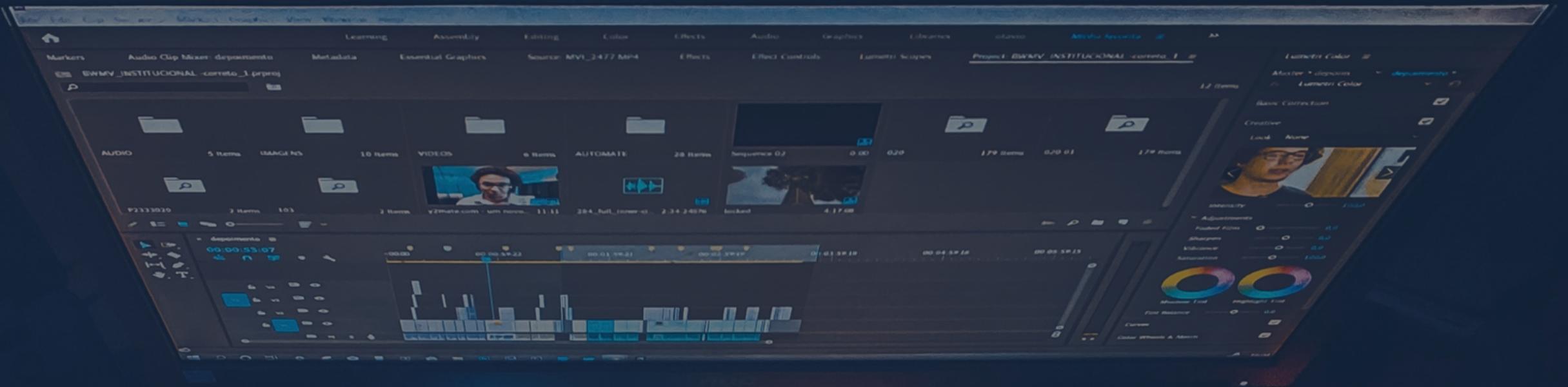
3 / APLICACIÓN DEL MODELO ELEGIDO

XGB CLASSIFIER()

- colsample_bytree=0.5
- learning_rate=0.1
- max_depth=None
- min_child_weight=1
- n_estimators=200
- reg_alpha=0.1
- reg_lambda=0.1
- subsample=0.9



Lucas Brito - sopro
Rodrigo Cartier - back vocal
Producelo -
Felipe Ranieri
Daniel Dutra
Iago Suarez
Matheus Montoto
Mix/Master -
Arthur Luna
Arranjo -
Lk
Filmagem -
markind3
Edição -
markind3 e Lk



4 / PROYECTO A FUTURO



MUCHAS GRACIAS

DANIEL CARRERA
PROYECTO MACHINE LEARNING