PROYECTO INFORME FINAL

PROYECTO DE ANALÍTICA DE DATOS

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

PRESENTADO POR:
SAMUEL GIL ARBOLEDA

UNIVERSIDAD DE ANTIOQUIA

DOCENTE:
RAÚL RAMOS POLLÁN

2022-1

1. INTRODUCCIÓN

1.1 PROBLEMA PREDICTIVO

En este trabajo se aborda el problema de clasificación, que tiene como objetivo predecir si un sitio web es considerado malicioso (phishing) o no, teniendo en cuenta características en la URL que normalmente tienen estos sitios. Este problema es de vital importancia debido a que estos sitios son difíciles de reconocer para gente que no tiene conocimientos en tecnología, por lo cual es necesario resolverlo mediante técnicas de aprendizaje de máquina debido a que reconocerlas manualmente podría conllevar mucho tiempo o ser un trabajo imposible, ya que la cantidad de páginas web existentes son prácticamente infinitas.

1.2 DATASET

El dataset a utilizar proviene de un repositorio de datos de machine learning llamado UCI (https://archive.ics.uci.edu/ml/datasets/phishing%20websites#). La base de datos de este proyecto es “Phishing Websites Data Set” [5], cuenta con 11055 muestras, y 31 variables, las cuales 30 son para explicar las características de una url de un sitio web, y la otra es la variable de salida “Result” la cual se encarga de decirnos si un sitio web es phishing o no.

Las variables presentes en el dataset pueden contener 2 o 3 de los siguientes valores: -1, 0 ó 1 , lo cual significa que cumplen con la condición propuesta en la variable, es sospechosa, y no la cumple, respectivamente. Además no hay datos faltantes en ninguna característica.

#	Nombre variable	Codificación
0	having_IP_Address	(categorical - signed numeric) : { -1,1 }
1	URL_Length	(categorical - signed numeric) : { 1,0,-1 }
2	Shortining_Service	(categorical - signed numeric) : { 1,-1 }
3	having_At_Symbol	(categorical - signed numeric) : { 1,-1 }
4	double_slash_redirecting	(categorical - signed numeric) : { -1,1 }
5	PrefixSuffix-	(categorical - signed numeric) : { -1,1 }
6	having_Sub_Domain	(categorical - signed numeric) : { -1,0,1 }
7	SSLfinal_State	(categorical - signed numeric) : { -1,1,0 }
8	Domain_registeration_length	(categorical - signed numeric) : { -1,1 }
9	Favicon	(categorical - signed numeric) : { 1,-1 }

10	port	(categorical - signed numeric) : { 1,-1 }
11	HTTPS_token	(categorical - signed numeric) : { -1,1 }
12	RequestURL	(categorical - signed numeric) : { 1,-1 }
13	URL_of_Anchor	(categorical - signed numeric) : { -1,0,1 }
14	Links_in_tags	(categorical - signed numeric) : { 1,-1,0 }
15	ServerFormHandler	(categorical - signed numeric) : { -1,1,0 }
16	Submitting_to_email	(categorical - signed numeric) : { -1,1 }
17	AbnormalURL	(categorical - signed numeric) : { -1,1 }
18	Redirect	(categorical - signed numeric) : { 0,1 }
19	on_mouseover	(categorical - signed numeric) : { 1,-1 }
20	RightClick	(categorical - signed numeric) : { 1,-1 }
21	popUpWidnow	(categorical - signed numeric) : { 1,-1 }
22	Iframe	(categorical - signed numeric) : { 1,-1 }
23	AgeofDomain	(categorical - signed numeric) : { -1,1 }
24	DNSRecord	(categorical - signed numeric) : { -1,1 }
25	web_traffic	(categorical - signed numeric) : { -1,0,1 }
26	PageRank	(categorical - signed numeric) : { -1,1 }
27	GoogleIndex	(categorical - signed numeric) : { 1,-1 }
28	LinksPointingToPage	(categorical - signed numeric) : { 1,0,-1 }
29	Statistical_report	(categorical - signed numeric) : { -1,1 }
30	Result	(categorical - signed numeric) : { -1,1 }

Fig 1. Columnas del dataset

1.3 MÉTRICAS

Para evaluar el sistema se van a usar las siguientes métricas de evaluación: accuracy y f1 score. Siendo accuracy la medida principal.

Por otra parte, en cuanto a la métrica de negocio, se tiene interés en que las predicciones sean lo suficientemente confiables para saber si un sitio web es de pishing o no. Con esta información un navegador de internet podría evitar que sus usuarios entren a estos sitios maliciosos solo leyendo la página a la cual se dirigen.

1.4 DESEMPEÑO ESPERADO

Lo que se esperaría de un modelo de este tipo es obtener una predicción con bastante desempeño (más de un 80% de precisión) , porque no sería viable tener muchos falsos positivos, ya que bloquear constantemente las páginas de un usuario de un navegador podría hacer que este deje de usarlo. En un ambiente productivo sería usado como filtro para evitar que los usuarios entren a las páginas que sean sospechosas.

2. EXPLORACIÓN DESCRIPTIVA DEL DATASET

La base de datos de este proyecto es “Phishing Websites Data Set”[1], cuenta con 11055 muestras, y 31 variables, las cuales 30 son para explicar las características de una url de un sitio web, y la otra es la variable de salida “Result” la cual se encarga de decirnos si un sitio web es phishing o no. La distribución de las clases es de 6157 muestras para la clase 1 (no phishing) y 4898 muestras para la clase -1 (phishing), por lo cual es un problema balanceado al no tener una diferencia significativa entre el número de muestras de una clase con respecto a la otra.

Para evaluar el sistema se usaron las siguientes medidas de evaluación: accuracy y f1 score. Siendo accuracy la medida principal.

Se procedió entonces en una primera instancia a realizar la respectiva exploración y limpieza de los datos del dataset:

Out[ ]:

	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	F
0	-1	1	1	1	-1	-1	-1	-1	-1	
1	1	1	1	1	1	-1	0	1	-1	
2	1	0	1	1	1	-1	-1	-1	-1	
3	1	0	1	1	1	-1	-1	-1	1	
4	1	0	-1	1	1	-1	1	1	-1	

5 rows × 31 columns

Dimensiones del dataset:

In [ ]:
```
data.shape
```

Out[ ]:
```
(11055, 31)
```

Fig 2. Dataframe del dataset

En las imágenes que se muestran a continuación, se evidencia que no hay presencia de datos nulos o faltantes. **Nota:** Si bien para el proyecto se pedía una un porcentaje específico de datos nulos, por temas de tiempo no se pudieron simular correctamente.

```
In [ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11055 entries, 0 to 11054
Data columns (total 31 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   having_IP_Address            11055 non-null  int64
 1   URL_Length                   11055 non-null  int64
 2   Shortining_Service           11055 non-null  int64
 3   having_At_Symbol             11055 non-null  int64
 4   double_slash_redirecting     11055 non-null  int64
 5   Prefix_Suffix                11055 non-null  int64
 6   having_Sub_Domain            11055 non-null  int64
 7   SSLfinal_State               11055 non-null  int64
 8   Domain_registeration_length  11055 non-null  int64
 9   Favicon                      11055 non-null  int64
 10  port                         11055 non-null  int64
 11  HTTPS_token                  11055 non-null  int64
 12  Request_URL                  11055 non-null  int64
 13  URL_of_Anchor                11055 non-null  int64
 14  Links_in_tags                11055 non-null  int64
 15  SFH                          11055 non-null  int64
 16  Submitting_to_email          11055 non-null  int64
 17  Abnormal_URL                 11055 non-null  int64
 18  Redirect                     11055 non-null  int64
 19  on_mouseover                 11055 non-null  int64
 20  RightClick                   11055 non-null  int64
 21  popUpWidnow                  11055 non-null  int64
 22  Iframe                       11055 non-null  int64
 23  age_of_domain                11055 non-null  int64
 24  DNSRecord                    11055 non-null  int64
 25  web_traffic                  11055 non-null  int64
 26  Page_Rank                    11055 non-null  int64
 27  Google_Index                 11055 non-null  int64
 28  Links_pointing_to_page       11055 non-null  int64
 29  Statistical_report           11055 non-null  int64
 30  Result                       11055 non-null  int64
dtypes: int64(31)
memory usage: 2.6 MB
```

Fig 3. Datos nulos

```
Out[ ]: having_IP_Address              0
        URL_Length                    0
        Shortining_Service            0
        having_At_Symbol              0
        double_slash_redirecting      0
        Prefix_Suffix                 0
        having_Sub_Domain             0
        SSLfinal_State                0
        Domain_registeration_length   0
        Favicon                       0
        port                          0
        HTTPS_token                   0
        Request_URL                   0
        URL_of_Anchor                 0
        Links_in_tags                 0
        SFH                           0
        Submitting_to_email           0
        Abnormal_URL                  0
        Redirect                      0
        on_mouseover                  0
        RightClick                    0
        popUpWidnow                   0
        Iframe                        0
        age_of_domain                 0
        DNSRecord                     0
        web_traffic                   0
        Page_Rank                     0
        Google_Index                  0
        Links_pointing_to_page        0
        Statistical_report            0
        Result                        0
        dtype: int64
```

Fig 4. Datos faltantes

La explicación de cada columna se encuentra en la documentación del dataset en el siguiente link: https://archive.ics.uci.edu/ml/machine-learning-databases/00327/Phishing%20Websites%20Features.docx, es bastante extensa, por lo que no se añade a este

documento.
El nombre de las columnas presentes en el dataset son las siguientes:

```
In [ ]: data.columns

Out[ ]: Index(['having_IP_Address', 'URL_Length', 'Shortining_Service',
       'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
       'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
       'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
       'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL',
       'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe',
       'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
       'Google_Index', 'Links_pointing_to_page', 'Statistical_report',
       'Result'],
      dtype='object')
```

Fig 5. Columnas del dataset

Adicionalmente, se decidió eliminar la columna 'ID' ya que esta no es una variable que sea relevante para el análisis.

```
In [ ]: del data["id"]
        data.head()
```

Fig 6. Eliminación columna ‘ID’

2.1 NORMALIZACIÓN, BALANCEO DE LOS DATOS y TRAIN TEST SPLIT

En la exploración del dataset, se evidenció que no existe un desbalance en los datos con un total de 11055 muestras, 6157 muestras para la clase 1 (no phishing) y 4898 muestras para la clase -1 (phishing).

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa68b6e71d0>
```

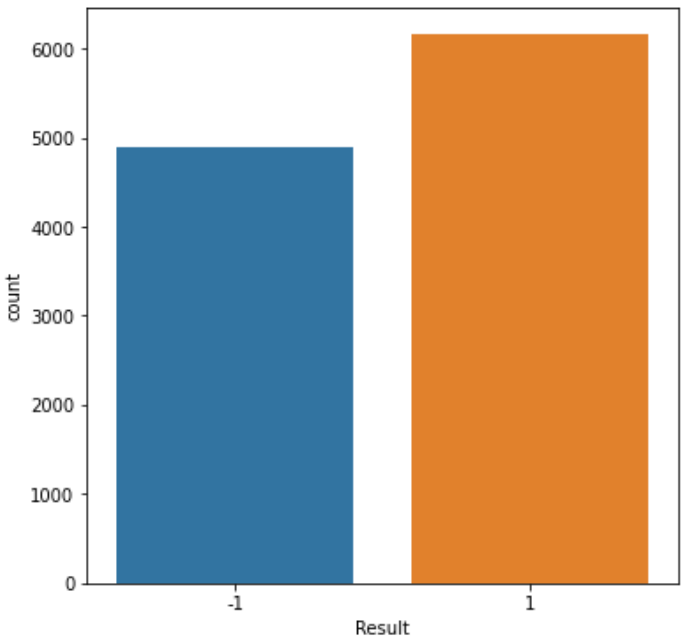

Fig 7. balance en los datos

Se dividió el dataset en dos bloques destinados al entrenamiento de los datos y validación del modelo:

Separacion de X y Y:

In [ ]:
```
X = data.drop('Result', axis=1).values
Y = data['Result'].values
print (X.shape , Y.shape)
#1
```

```
(11055, 30) (11055,)
```

Separacion de test y train: (Para probar codigos y disminuir tiempo de ejecucion)

In [ ]:
```
#Para pruebas con modelos complicados
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.01, random_state=0)
print (X_test.shape , Y_test.shape)
```

```
(111, 30) (111,)
```

Fig 8. Train Test Split

REFERENCIAS

1. UCI Machine Learning Repository: Phishing Websites Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites .