ANÁLISIS Y PREDICCIÓN DE FRAUDE ELECTRÓNICO MEDIANTE REDES NEURONALES DE CLASIFICACIÓN

Aplicación de Red Neuronal tipo MLP Classifier

Jeison Robles Arias

roblesjeison@gmail.com

Resumen

Este trabajo explora las funcionalidades de la librería de código abierto SK LEARN en el análisis y procesamiento de datos relacionados a transacciones bancarias electrónicas con la finalidad de crear un algoritmo de aprendizaje electrónico basado en redes neuronales que sea capaz de aprender el comportamiento de las transacciones fraudulentas o no, basado en el análisis de hasta 24 dimensiones de datos. El trabajo se desarrolla en Python 3 y utilizando Spyder 4 de anaconda.

Palabras Clave: SK Learn, Aprendizaje Electrónico supervisado, redes neuronales, fraude electrónico.

Abstract

This work explores the functionalities of the SK LEARN open source library in the analysis and processing of data related to electronic banking transactions in order to create an electronic learning algorithm based on neural networks that is capable of learning the behavior of fraudulent transactions. or not, based on the analysis of up to 24 data dimensions. The work is carried out in Python 3 and using Spyder 4 from anaconda.

Keywords: SK Learn, Machine learning, neural networks, electronic fraud.

I. INTRODUCCIÓN

Las transacciones bancarias mediante mecanismos electrónicos son ampliamente utilizadas en casi cualquier parte del mundo y por tanto son un aperitivo importante para organizaciones criminales que se dedican a cometer distintos tipos de fraude electrónico, aprovechando la virtualidad y tomando ancla en ingeniería social y portillos dentro de los procesos de las organizaciones para la sustracción de información y activos monetarios.

Por esta razón es importante que las áreas de inteligencia de las distintas entidades, entre ellas la banca, actúen de forma preventiva, aplicando modelos de aprendizaje electrónico y estadístico que permitan no solo identificar cuando y como un fraude ocurrió, sino que puedan predecir cuando, a quien y donde ocurrirán estos fraudes y con base en estos análisis se puedan tomar mecanismos de acción efectivos en la mitigación de estos alertamientos.

II. METODOLOGÍA

Durante este trabajo se explora un set de datos (base de datos) de una entidad aleatoria que brinda cadenas extensas de datos de transacciones bancarias que han sido ejecutadas durante cierta cantidad de tiempo y en las cuales se ha identificado cada transacción como transacción fraudulenta (etiquetada con un numero 1) y transacción no fraudulenta (etiquetada con el numero 0), para el caso de estudio se utiliza un registro de poco mas de un millón de transacciones con la finalidad de aumentar la precisión de las predicciones realizadas. La base de datos utilizada para el ejemplo es conectada desde kaggle.com, que es un repositorio de datos de calidad para el análisis de modelos de machine learning.

Figura 1. Logo de kaggle.

Se procede a realizar un análisis y explicación de los datos contenidos en el set de datos y posteriormente se realiza un análisis de pre procesamiento de información para balancear los pesos de datos, etiquetado de datos y una etapa de análisis de componentes primarios para determinar las dimensiones de datos mas influyentes y que se utilizaran en a creación de una red neuronal.

La red neuronal utilizada para este ejemplo se crea mediante el uso de la librería de SK Learn y el tipo de red es el Clasificador Perceptron Multi Capa, el cual por el tipo de datos suele ser de amplio rendimiento.

Data set seleccionado:

Como se menciono anteriormente, se tiene una base de datos en un archivo csv con 1 048 576 registros de transacciones etiquetados con una columna de target “is_fraud”, la cual es la salida resultante de dicha transacción previamente identificada por los procesos de dicha organización.

Para este caso la etiqueta de datos se distribuye de la siguiente forma>

Dato de salida	Resultado
0	Transacción no fraudulenta
1	Transacción fraudulenta

Tabla 1. Target de fraudes.

Las librerías utilizadas durante el desarrollo de este proyecto son las siguientes:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import LabelEncoder
```

```
import train_test_split
```

```
import StandardScaler

import MLPClassifier
```

Figura 1. Librerías utilizadas en Python 3.

Todas siendo previamente instaladas y comprobadas en el ambiente “Spyder”

Los datos a trabajar se detallan en el siguiente listado, que basicamente es informacion de hora,fecha, lugar, ciudad, codigo postal, punto de origen, nombre, genero, edad,estado, entre otros. Se presenta un resumen de los datos:

```
trans_date_trans_time
cc_num
merchant
category
amt
first
last
gender
street
city
state
zip
lat
long
city_pop
job
dob
trans_num
unix_time
merch_lat
merch_long
is_fraud
```

Figura 2. Variables del data set

Pre procesamiento

Como se menciono en la seccion anterior, se tiene una etapa de preprocesamiento de datos, con la finalidad de ejecutar acciones basicas que permitan un mejor manejo de los datos y una selección de caracteristicas optima para dicho ejercicio.

Se trabajo un proceso de escalamiento de datos basados en la media y su desviacion estandar para evitar errores en los calculos de correlacion.

Adicionalmente se maneja un analisis de caracteristicas, calculando la correlacion de los datos con respecto a la salida de si una transaccion fue o no fraudulenta. Gracias a este analisis fue posible identificar las variables con una correlacion idonea

Dicho analisis de correlacion se extrae y presenta en la siguiente imagen:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{\operatorname{Cov}(X,Y)}{\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}}$$

```
ANALIZANDO CORRELACION DE DATOS...
is_fraud      1.000000
amt           0.250768
category      0.019971
Mes           0.013969
Hora          0.012416
Dia           0.009354
Unnamed: 0    0.009041
unix_time     0.007936
zip           0.001844
city_pop      0.001824
merch_long    0.001453
long          0.001426
lat           0.000983
job           0.000660
merchant      0.000631
state         0.000309
merch_lat     0.000149
city          0.000130
```

```
print(Fraude.corr()['is_fraud'].abs().sort_values(ascending=False))
```

Figura 3. Calculo de correlación de datos.

Luego de este análisis se decide utilizar las siguientes variables, catalogadas como mas influyentes en el resultado (target)

X son las variables de entrada y "**y**" es la variable de salida. El resto de variables se dejan para futuros análisis pero no se contemplan en el diseño de la red neuronal.

```
X = Fraude[["amt","category","Mes","Hora","unix_time","Dia"]]
y = Fraude["is_fraud"]
```

Figura 4. Selección de características para los sets de prueba

Extraccion de datos de prueba y datos de entrenamiento

Los datos son divididos en datos entrenamiento para que la red neuronal pueda aprender el comportamiento y datos de prueba, con los cuales se puede probar la eficiencia de la prediccion ejecutada por el modelo.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y)
```

Figura 5. Codigo de train test split

Esto se logra con la clase Model Selection de SK Learn, el cual contiene un divisor de datos aleatorios para prueba y para entrenamiento.

Creación de la red neuronales

La red neuronal que se selecciono es la de Clasificador Perceptron Multi Capa, cuya arquitectura se muestra a continuación:

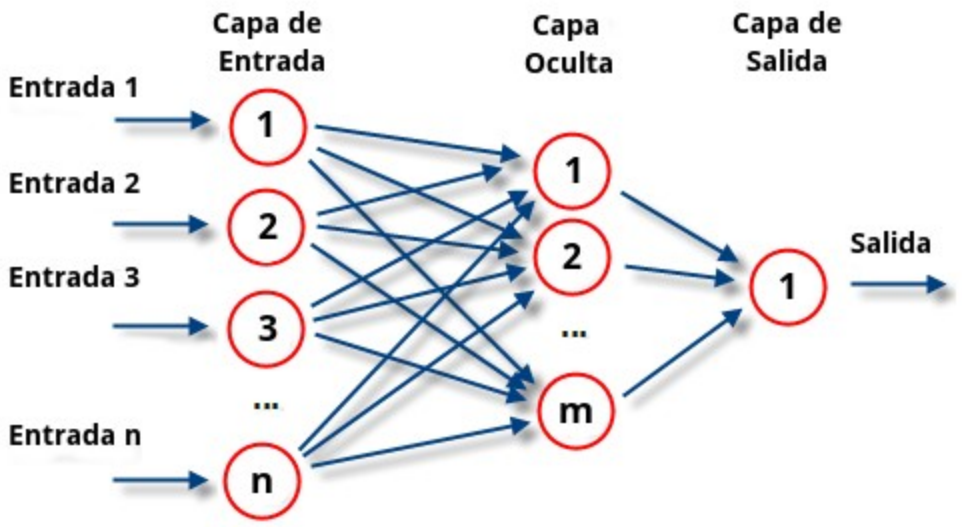

Figura 6. Arquitectura de Perceptron Multi Capa

La misma posee capa de entrada con las variables ya comentadas, las capas ocultas con hasta diez neuronas y tres capas y una capa de salida con una sola neurona de muestra de la clasificación.

```
from sklearn.neural_network import MLPClassifier
```

Figura 7. MLP Classifier

III. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos en general son moldeables de acuerdo a las necesidades del negocio en el que se desee operar, por ejemplo, para aplicaciones de prevención de fraudes, deberá estudiarse a detalle la operativa y el ambiente de trabajo de la corporación.

En general la red neuronal tarda alrededor de 30 minutos en entrenarse y para su estabilización requiere cerca de 200 iteraciones.

La precisión de la red alcanza con este análisis y detalle, un 89% de eficiencia para el calculo de fraudes y el 100% para calcular transacciones no fraudulentas.

Se coloco un resumen a continuación donde se detalla el resultado directamente de Spyder.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	260661
1	0.89	0.69	0.78	1483
accuracy			1.00	262144
macro avg	0.94	0.84	0.89	262144
weighted avg	1.00	1.00	1.00	262144

Figura 8. Resumen de resultados.

IV. CONCLUSIONES

Los modelos de machine learning son ampliamente utilizados para la identificacion de patrones y en el caso estudiado se alcanzo una eficiencia del 89% para la prediccion del fraude mediante el uso de una red de clasificacion de perceptron multicapa.

La informacion correctamente categorizada es sumamente valiosa para la correcta prediccion de datos.

Un modelo de redes neuronales robusto puede apoyar a prevenir el fraude antes de que se cometa o en el momento que el mismo se esta ejecutando.

V. REFERENCIAS

https://es.wikipedia.org/wiki/Coeficiente_de_correlaci%C3%B3n_de_Pearson

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

https://runebook.dev/es/docs/scikit_learn/modules/generated/sklearn.feature_selection.chi2

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html