

**UNIVERSIDAD DEL VALLE**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**SEDE CENTRAL COCHABAMBA**

**APLICACIÓN DE DETECCION DE NOTICIAS FALSAS CON TECNOLOGIA MACHINE LEARNING Y DATASET DEFINIDO PARA DISMINUIR INFORMACION EN LOS MEDIOS EN LA CIUDAD DE COCHABAMBA**

**ESTUDIANTES(S): PEDRO ANDRES CONDE VALDEZ (50% Contribución)**

**AXEL MATIAS MIRANDA CLAVEL (50% Contribución)**

**TUTOR: ING. OSCAR CONTRERAS CARRASCO**

Cochabamba - Bolivia

2022

**APLICACIÓN DE DETECCION DE NOTICIAS FALSAS CON TECNOLOGIA MACHINE LEARNING Y DATASET DEFINIDO PARA DISMINUIR INFORMACION EN LOS MEDIOS EN LA CIUDAD DE COCHABAMBA**

1. **INTRODUCCIÓN**

En Bolivia estamos retrasados en avance tecnológico como es evidente, esto afecta a los sectores del periodismo y los medios ya que la desinformación de algunos de estos tiende a afectar de manera negativa a la ciudad de Cochabamba

En el caso boliviano, desde las Elecciones Generales de 2009, ya se observaba el uso de medios digitales para las campañas electorales, principalmente en forma de blogs, y en las de 2014 se pasó la campaña a las de redes sociales, pero con un uso unidireccional (PNUD, 2014). Sin embargo, no es hasta el Referendo Constitucional de 2016 donde se ve una mayor presencia de las redes en la política electoral, tanto así que los actores políticos de uno y otro bando atribuyeron los resultados a las campañas desplegadas en redes sociales. Ya para las Elecciones Generales de 2019 se esperaba una mayor presencia de desinformación deliberada y noticias falsas por parte de los actores políticos.

Esa crisis posee muchas facetas y factores. Uno de esos factores, centrales por su índice de contemporaneidad, fue la desinformación que ha circulado en las redes sociales digitales. Pero ¿cómo se ha desarrollado la producción de desinformación alrededor del conflicto?, ¿qué temáticas predominaron en los contenidos falsos?, ¿qué tendencias políticas poseen?, ¿cómo han circulado en redes sociales?, ¿existe alguna estrategia de desinformación predominante? Nuestro objetivo aquí es analizarlas tendencias generales en producción y circulación de desinformación alrededor del conflicto electoral de 2019 en Bolivia.

El proyecto tiene como misión crear una IA capaz de detectar noticias ya sean falsas o verdaderas. intuitiva y fácil de usar, dirigido a todo público, con ella podrás ver las noticias que te brinda el Data set y verificar si una noticia es Falsa o Verdadera (Colocar la tecnología y lo que usaremos para esto)

1. **PROBLEMÁTICA PARA RESOLVER**

Una característica única de las noticias en las redes sociales es que cualquier persona puede registrarse como editor de noticias sin ningún tipo de restricción (por ejemplo, cualquiera puede crear una página de Facebook que diga ser un periódico o una organización de medios de comunicación).

Como la extensa difusión de noticias falsas puede tener un impacto negativo grave en los individuos y en la sociedad, la falta de estrategias de verificación de hechos escalables es especialmente preocupante.

El problema como bien se introdujo anteriormente son la cantidad de noticias falsas que abundan en estos tiempos. Se interesa en crear una IA con Machine Learning, por medio de un Data set escogido para el entrenamiento de dicha Inteligencia Artificial, para que en un futuro pueda funcionar propiamente recopilando información de todos los sitios web y así verificar la veracidad de las noticias.

1. **OBJETIVOS**

* 1. **OBJETIVO GENERAL**

Realizar un Entrenamiento de una Inteligencia Artificial con Machine Learning usando Python y librerías de Python tales como numpy, pysparks, seaborn, matplot, etc para poder verificar la veracidad de noticias escogidas en un Data Set seleccionado para este propósito.

* 1. **OBJETIVOS ESPECÍFICOS**
* Realizar una Página Web en el cual se pueda verificar las noticias de la IA por medio del Data set especifico
* Python como lenguaje para realizar el recopila miento y procesamiento de datos
* Desarrollar con Librerías de Python para machine learning para el procesamiento de información y almacenamiento de información
* Uso de almacenamiento s3 de aws para subir la información procesada
* Desarrollar una aplicación web para la visualización de la información
* Desarrollar el backend con Python para la extracción y procesamiento de información

1. **DESARROLLO DEL PROYECTO**

En este caso nos encontramos con contenido web que puede obtenerse mediante [web scrapping](https://es.wikipedia.org/wiki/Web_scraping), y a cada noticia se la puede etiquetar en base a tantos criterios como se desee. Una de las condiciones para que los clasificadores de noticias falsas logren un buen desempeño es tener suficientes datos etiquetados. Sin embargo, obtener etiquetas en las que confiar requiere mucho tiempo y trabajo de catalogación.

Los modelos de clasificación más utilizados en la detección de noticias falsas son [Support Vector Machine](https://scikit-learn.org/stable/modules/svm.html) (SVM) y [Naive Bayes Classifier](https://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo) (NBC).

También se utilizan la [Regresión Logística](https://data-speaks.luca-d3.com/2019/02/predictor-de-infidelidad-ii-que-es-la.html) (LR) y los modelos basados [Árboles de Decisiones](https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n).

El Deep Learning también tiene cabida. Muchos tipos de modelos de redes neuronales, como los perceptrones multicapa, también funcionan para la detección de noticias falsas.

Las [Redes Neuronales Recurrentes](https://colah.github.io/posts/2015-08-Understanding-LSTMs/) (RNN) son muy populares en el procesamiento de lenguaje natural, especialmente en la memoria a largo plazo a largo plazo. Long Short Term Memory (LSTM).

Primero debemos tener un data set para este proyecto para poder entrenar, el data set obtendremos de internet.

Luego pasamos a importar las librerías de Python como: numpy, matplot, pandas, seaborn, sklearn y pysparks.

También una creación de un Docker para albergar la aplicación web y clúster en la nube en aws.

Primero se avanzó con la primera parte que es el backend en Python, la recopilación y extracción de información y el procesamiento de datos.

Luego se pasará con la carga de esa información en un repositorio para posteriormente subirla o guárdala en un almacenamiento de base de datos o repositorio con los datos procesados.

Finalmente se tiene que mostrar o visualizar dicha información en una aplicación web que posiblemente se esté usando flask.

Existen diversos servicios para la detección de Fake News basados en:

* Verificación de dominios de origen como fuente legítima o poco sesgada
* Análisis de patrones en el contenido de la noticia
* Búsqueda de keywords para mejorar la búsqueda, categorización y manejo de información

1. **Producto final**

**Ventana principal**



**Modulo Registro Usuarios**

En esta parte el usuario nuevo podrá registrarse al sitio web

Para eso se debió crear una base de datos y acceder a la conexión con ella

También se creó la tabla user para que se guarden la información del usuario

#configuracion para acceder al servidor y base de datos

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'Univalle'

app.config['MYSQL\_DB'] = 'newsDb'

#funcion para registrar un nuevo usuario

@app.route('/add', *methods*=['POST'])

*def* add\_user():

if request.method == 'POST':

name = request.form['name']

email = request.form['email']

password = request.form['password']. codificar('utf-8')

hash\_password = bcrypt. hashpw(contraseña, bcrypt. gensalt())

cursor = mysql.connection.cursor()

cursor.execute('INSERT INTO user (name, email, password) VALUES (%s, %s, %s)',

(name, email, hash\_password))

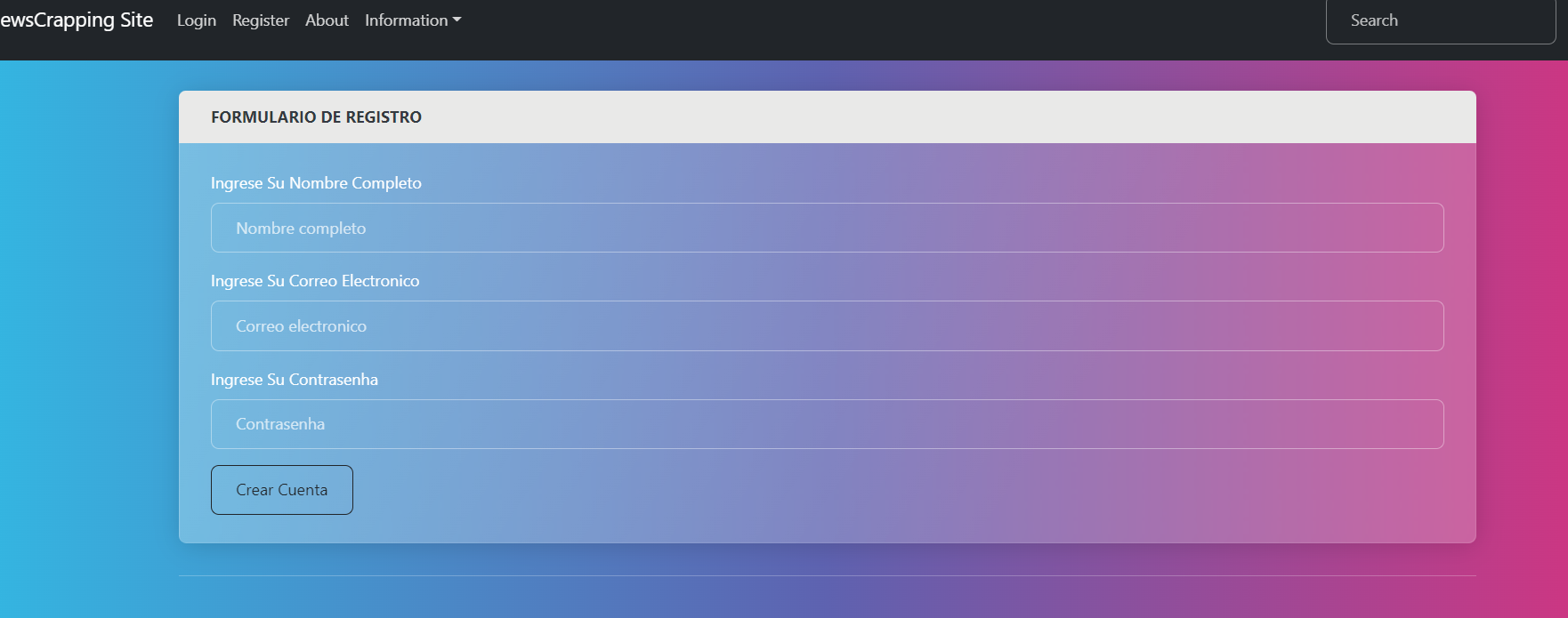
mysql.connection.commit()

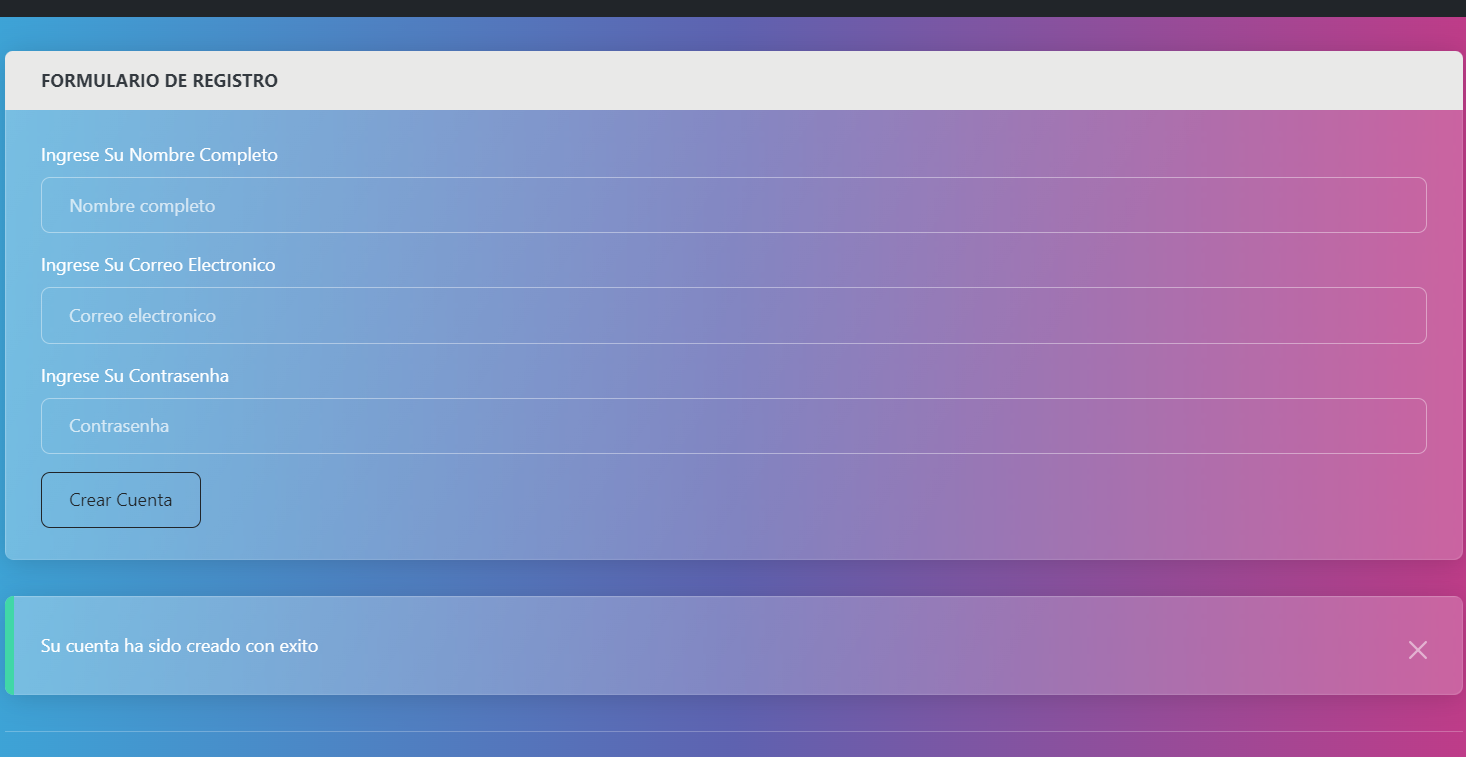
        flash('Su cuenta ha sido creado con exito', "info")

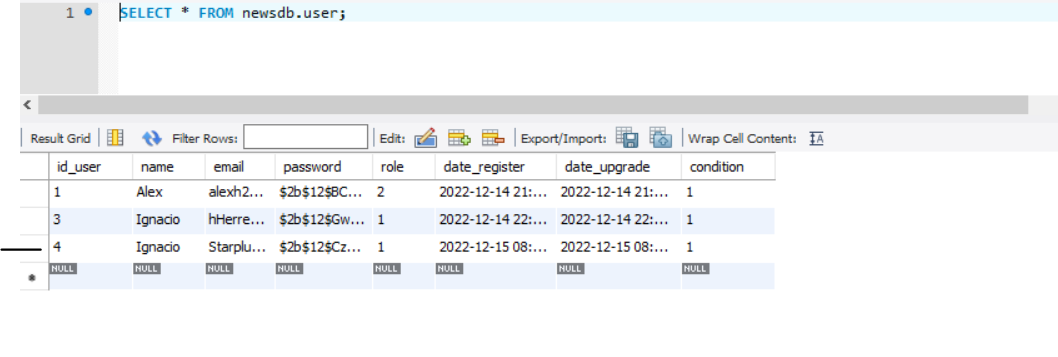
session['name'] = name

session['email'] = password

return redirect(url\_for('users\_list'))







**Modulo Inicio de Sesión**

En esta parte el usuario podrá acceder a iniciar sesión con su cuenta

Para que pueda ocurrir este evento accedemos al usuario con una cosulta select para agarrar el email y luego verificar si esta registrado y También su contraseña

Se uso manejo de sesión para cada usuario

#funcion para iniciar sesion

@app.route('/login', *methods*=['GET','POST'])

*def* login():

if request.method=="POST":

email = request.form['email']

password = request.form['password']. codificar('utf-8')

cursor = mysql.connection.cursor(MySQLdb. cursores. DictCursor)

cursor.execute('SELECT \* FROM user WHERE email=%s',(email,))

usuario = cursor.fetchone()

cursor.close()

Si len(user) > 0:

Si bcrypt. hashpw(password, user['password'].encode('utf-8')) == user['password'].encode('utf-8'):

session['email'] = usuario['email']

session['role'] = usuario['role']

session['name'] = usuario['name']

return redirect(url\_for('home'))

De lo contrario:

flash ('Enviar por correo electrónico la contraseña incorrecta', 'info')

           return "Error usuario o contrasenha"

De lo contrario:

return redirect(url\_for('users\_list'))

El usuario Tambien podra cerrar su cuenta mediante sesiones

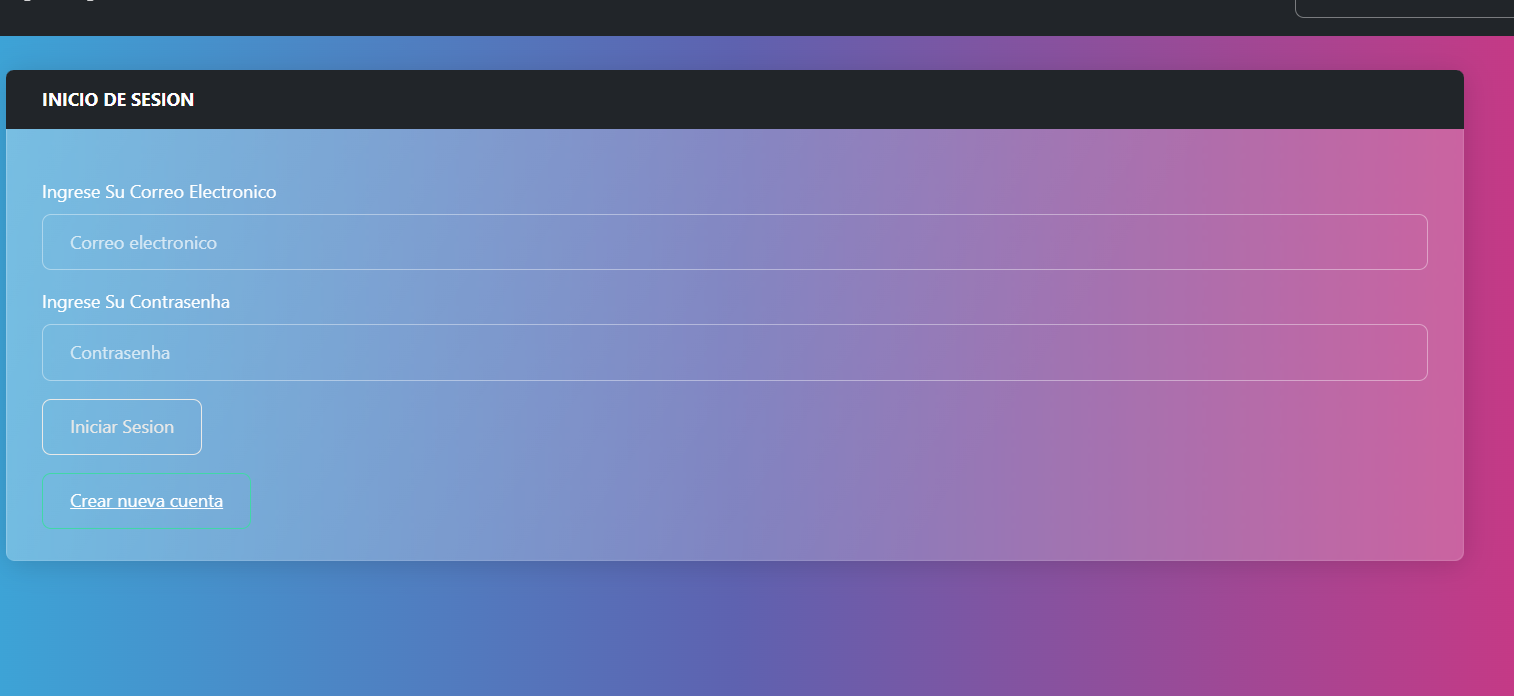
#funcion para cerrar sesion

@app.route('/logout')

*def* logout():

sesión. clear()

return redirect(url\_for('home'))





**Modulo Gestión de Usuarios**

En esta parte el administrador podrá realizar consultas como eliminar y mostrar

Para que esto ocurra le enviara una vista donde esta la lista de registros y podrá eliminar

Se realizo dos consultas una select para mostrar los registros y otra delete para eliminar el registro especificado

#mostrar las lista de registros

@app.route('/list', *methods*=['GET'])

*def* users\_list():

cursor = mysql.connection.cursor(MySQLdb. cursores. DictCursor)

cursor.execute('SELECT \* FROM user')

data = cursor.fetchall()

return render\_template('view\_register.html', *users*=data)

#eliminar registro existente de la tabla usuario

@app.route('/delete/<id>', *methods*=['POST', 'GET'])

*def* delete\_user(*id*):

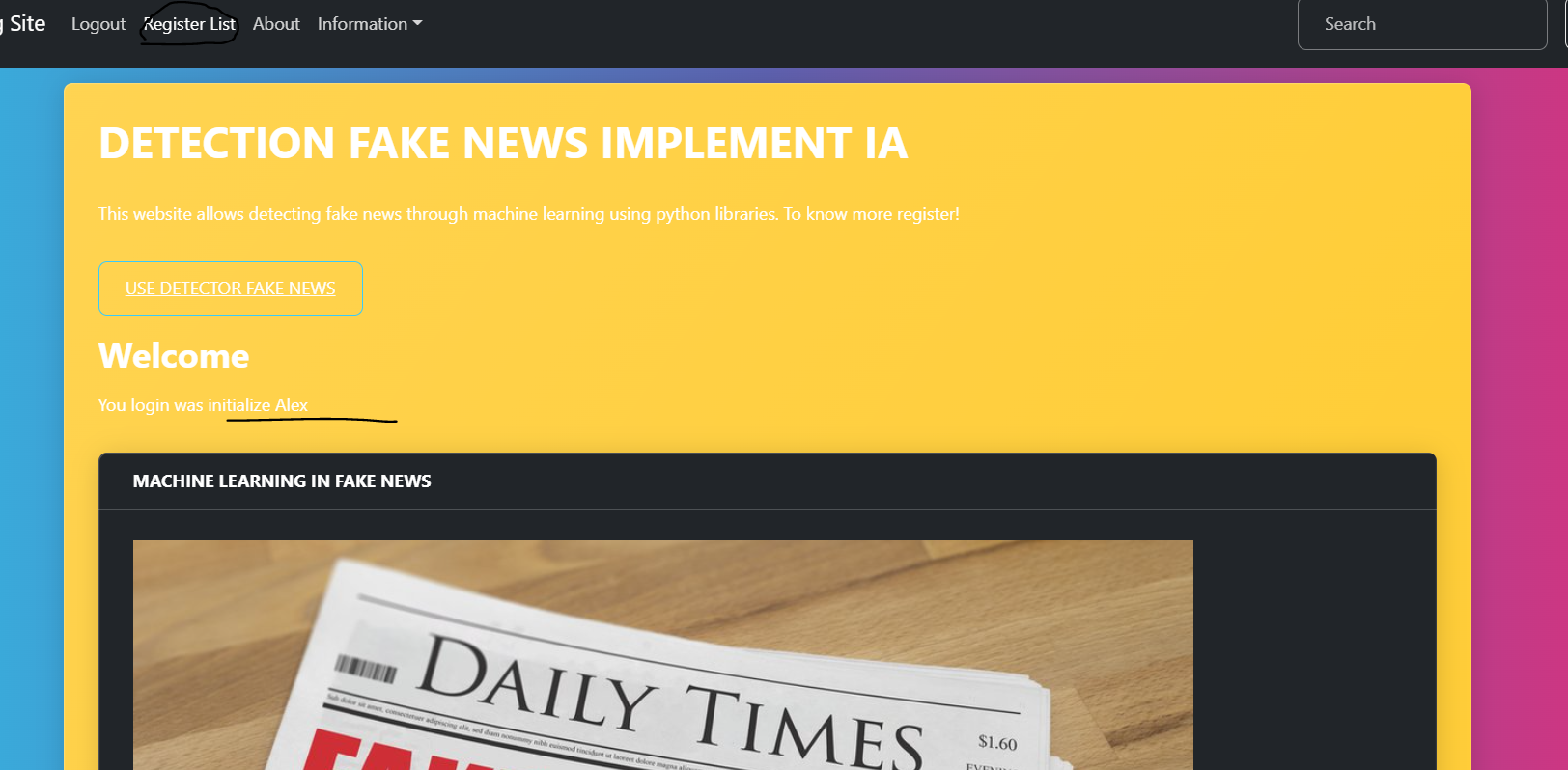
cursor = mysql.connection.cursor()

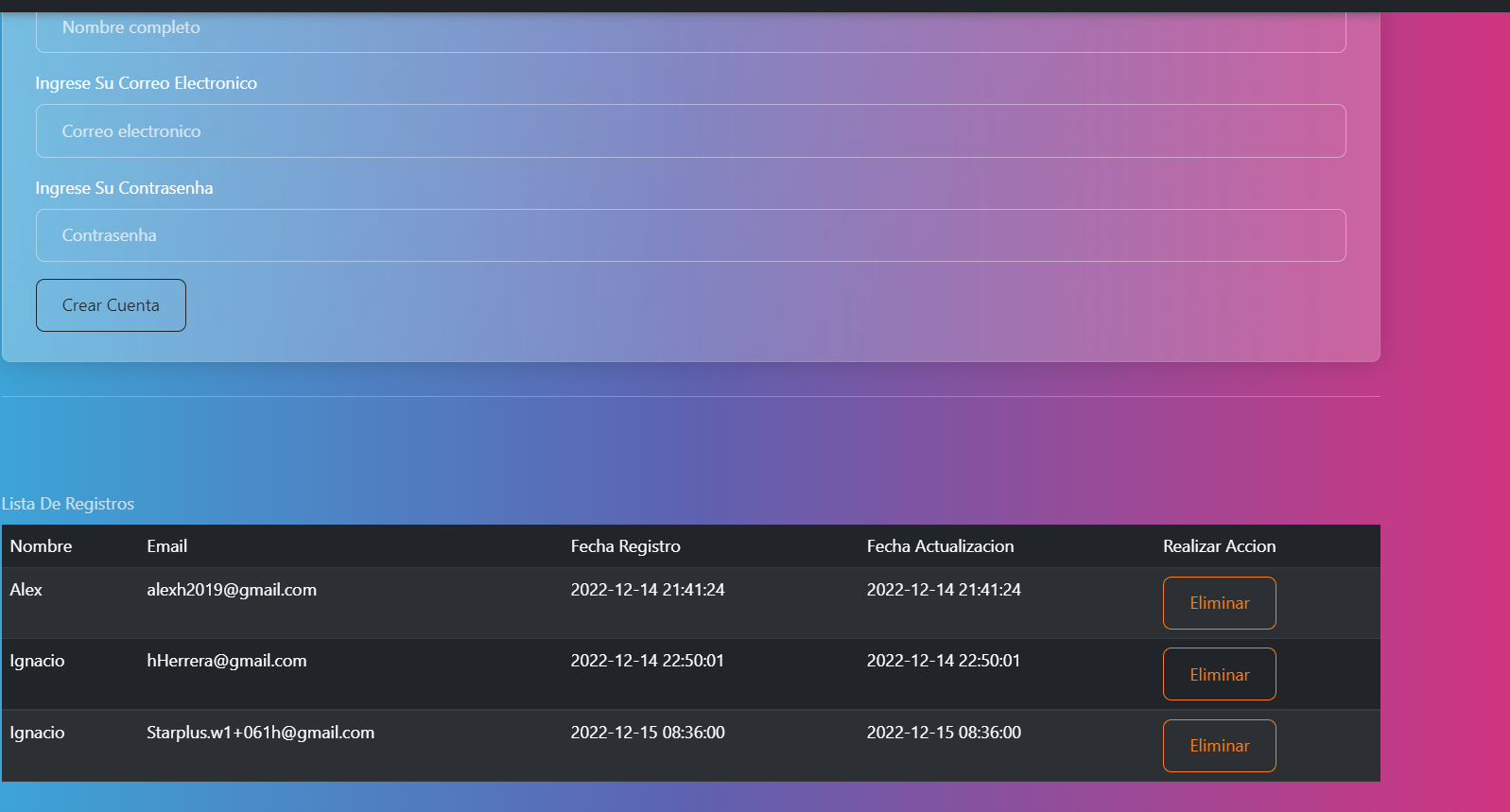
cursor.execute('DELETE FROM user WHERE id\_user = %s', *(id*))

mysql.connection.commit()

    flash('Registro eliminado')

return redirect(url\_for('users\_list'))





**Modulo Detección de Noticias Falsas**

En esta parte el usuario registrado podrá acceder a la vista para detectar la noticia falsa o real

Para que esto ocurra se crearon varios metodos para la extraccion de informacion y procesamiento de informacion de archivos csv que seran nuestro dataset usamos librerias sklearn, pandas, pickle, etc

Entrenamos nuestro modelo para que pueda reconocer que noticia es falsa o real

Luego usamos scrapping para que pueda detectar la noticia

#uso de machine learning en python

tfvect = TfidfVectorizer(*stop\_words*='inglés', *max\_df*=0.7)

#cargar el modelo para el dataset

loaded\_model = pepinillo. load(open('model.pkl', 'rb'))

#datset usados para el entrenamiento de la app

verdadero = pd. read\_csv('Verdadero.csv')

falso = pd. read\_csv('Falso.csv')

Datos = PD. read\_csv('noticias.csv')

#etiquetar las noticias

true['target'] = 'true'

fake['target'] = 'fake'

x = datos['texto']

y = datos['etiqueta']

#variables para entrenar el dataset

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, *test\_size*=0.2, *random\_state*=0)

#funcion para concatenar dos archivos

*def* concat\_file(*file\_true*, *file\_fake*):

    #concatenar los archivos true y fake en un solo dataset

Conjunto de datos = PD. concat([*file\_true*, *file\_fake*]). reset\_index(*drop* = True)

dataset['text'] = dataset['text']. apply(*lambda* x: x.lower())

Devolver conjunto de datos

#funcion para ordenacion y shuffle

*def* shuffling\_file(*dataset*):

datos = aleatorio(*conjunto de datos*)

data = data.reset\_index(*drop*=True)

datos de devolución

#funcion para limipar datos del dataset

*def* clear\_file(*dataset*):

clear\_data = *dataset.drop*(["date"],*axis*=1,inplace=True)

clear\_data = *dataset.drop*(["title"],*axis*=1,inplace=True)

Volver clear\_data

#funcion para detectar la noticia usando prediccion

*def* detect\_news(*noticias*):

tfid\_x\_train = tfvect. fit\_transform(x\_train)

tfid\_x\_test = tfvect. transformar(x\_test)

input\_data = [*noticias*]

vectorized\_input\_data = tfvect. transformar(input\_data)

predicción = loaded\_model.predict(vectorized\_input\_data)

imprimir(tfid\_x\_test)

imprimir(tfid\_x\_train)

Predicción de retorno

#ruta para la vista de deteccion de noticias

@app.route('/detector')

*def* detector\_view():

return render\_template('view\_detector.html')

#funcion para enviar la deteccion de la noticia ingresada

@app.route('/predict', *methods*=['POST'])

*def* predict():

if request.method == 'POST':

message = request.form['message']

pred = detect\_news(mensaje)

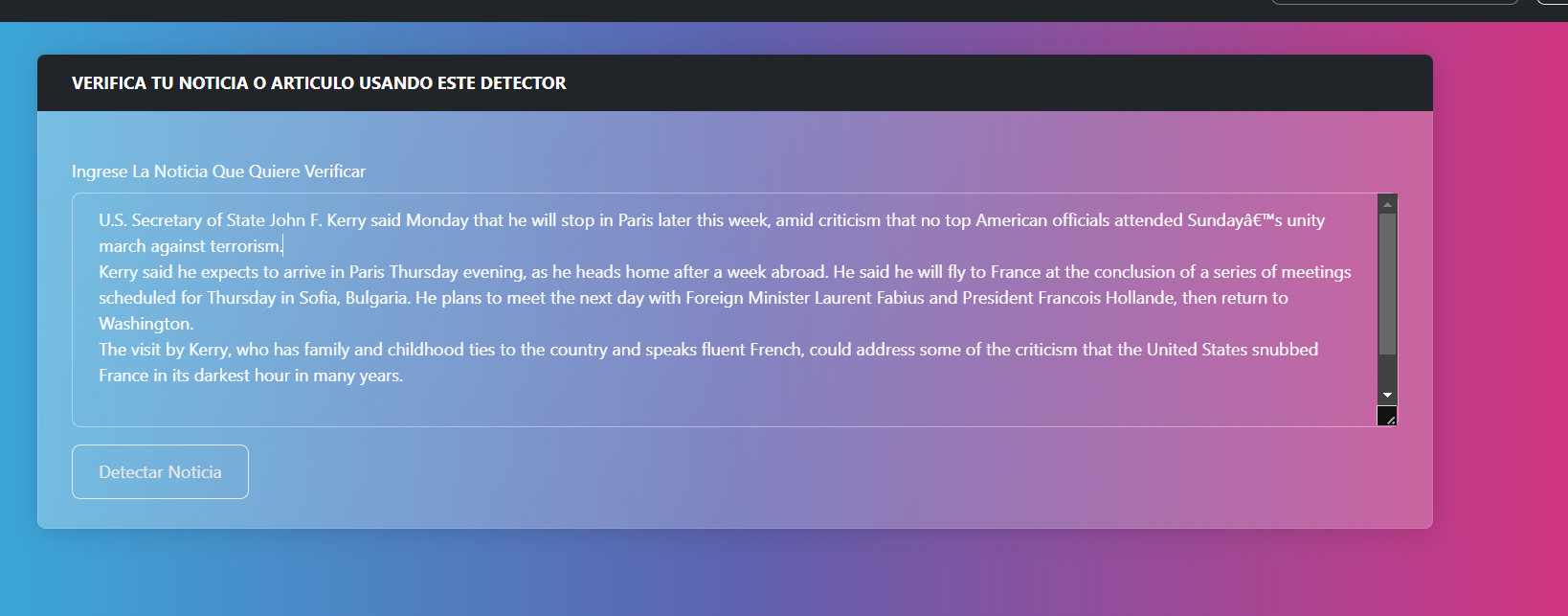
flash(pred, "info")

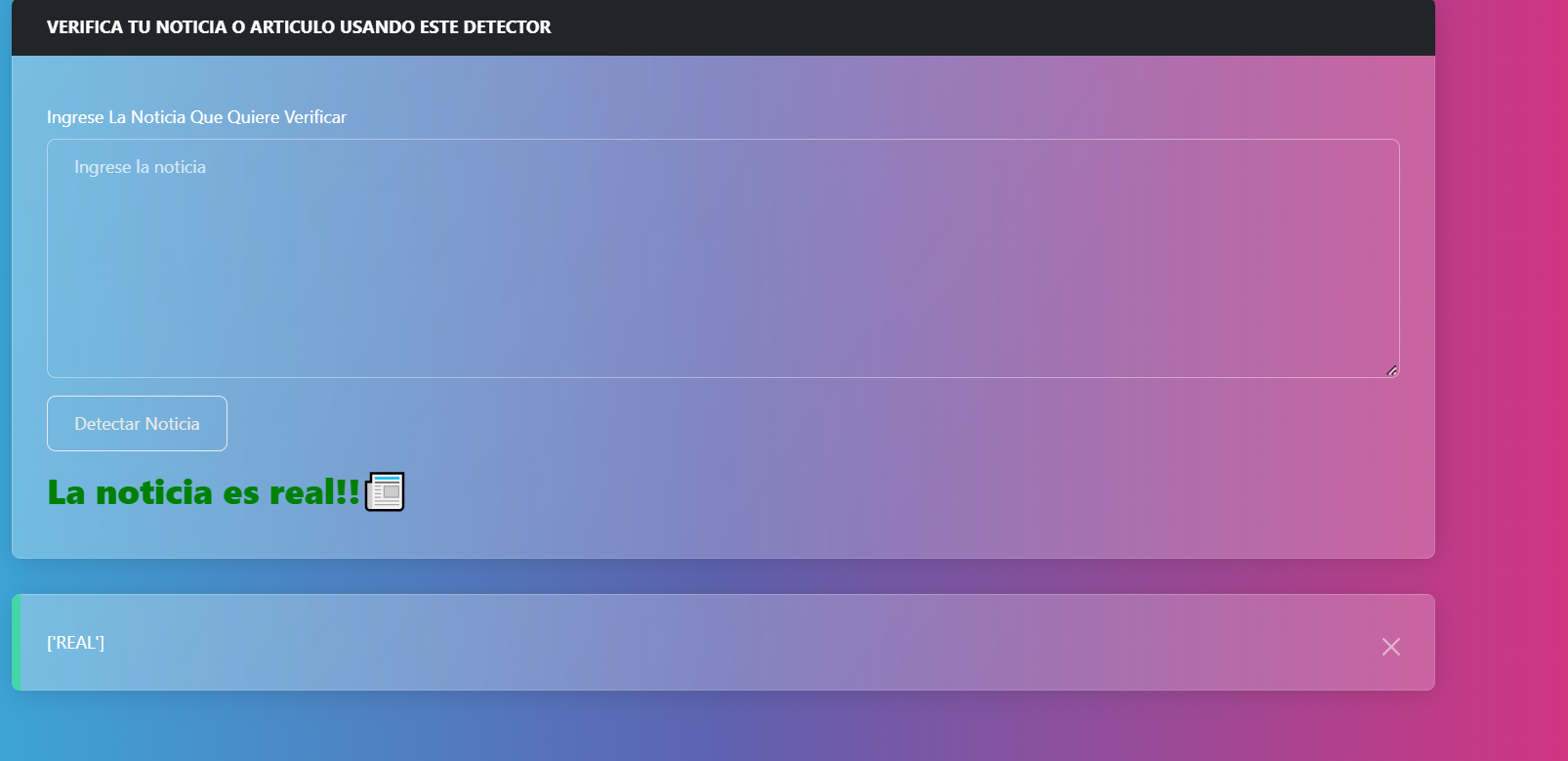
return render\_template('view\_detector.html', *prediction*=pred)

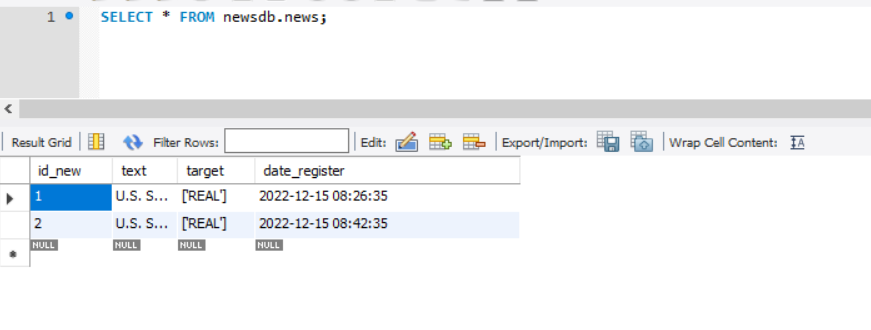
De lo contrario:

        flash("Vuelve a intentar mas tarde!" , "info")

        return render\_template('view\_detector.html', *prediction*="Esta en proceso!" )

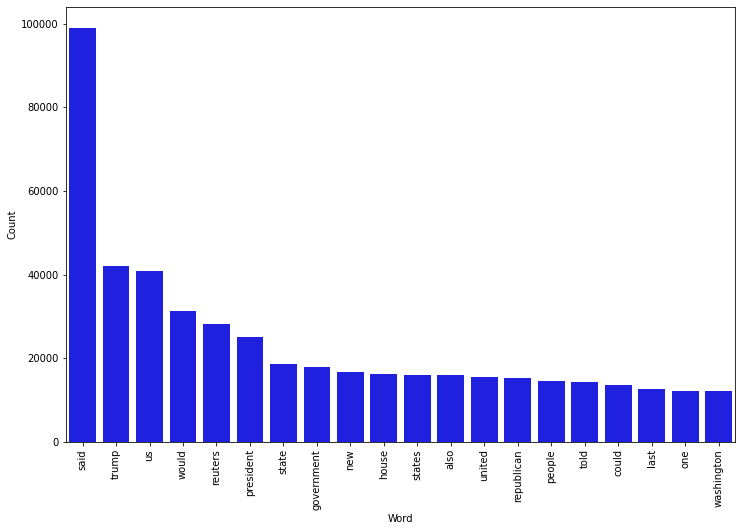


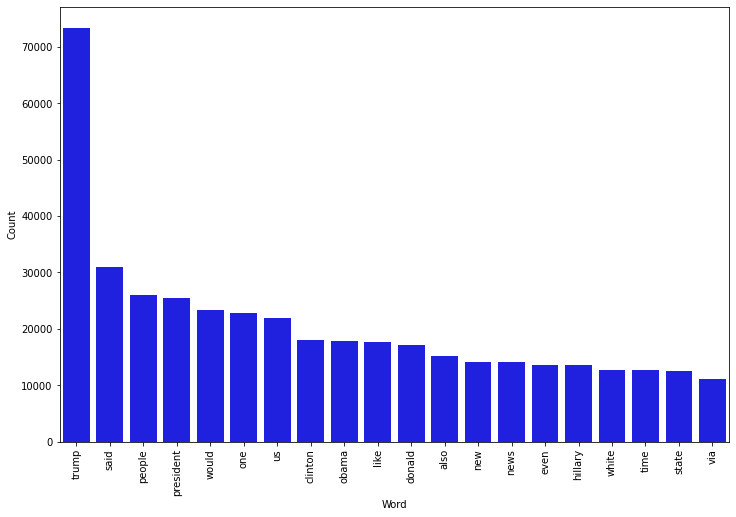




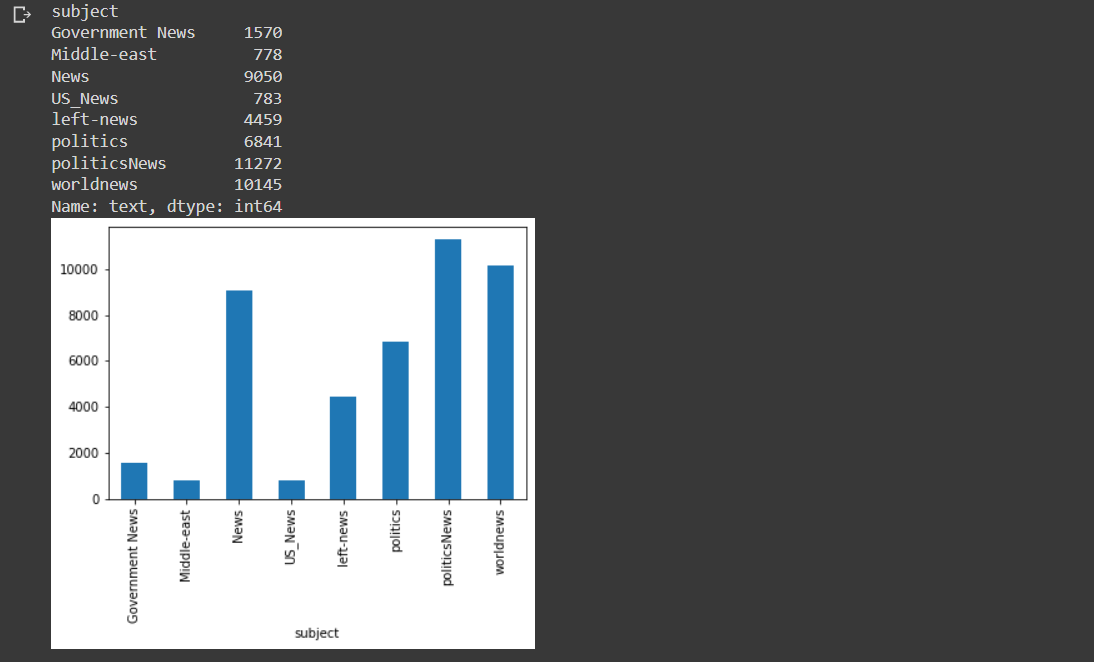
1. **Graficos y comparaciones del dataset**

Cantidad de palabras usadas en las noticias falsas

Cantidad de palabras usadas en las noticias reales



Cantidad de artículos por tema



Comparación de noticias falsas vs noticias reales

