Cómo utilizar Streamlit y Python para crear una aplicación de ciencia de datos

1. Introducción a Streamlit

Streamlit es una biblioteca de Python de código abierto para crear y compartir aplicaciones web para proyectos de ciencia de datos y aprendizaje automático. La biblioteca puede ayudarle a crear e implementar su solución de ciencia de datos en unos minutos con unas pocas líneas de código.

Streamlit puede integrarse perfectamente con otras bibliotecas de Python populares utilizadas en ciencia de datos, como NumPy, Pandas, Matplotlib, Scikit-learn y muchas más.

Nota: Streamlit usa React como marco de interfaz para representar los datos en la pantalla.

2. Instalación y configuración

Streamlit requiere la versión Python >= 3.7 en su máquina.

Para instalar streamlit, debe ejecutar el siguiente comando en la terminal.

```
pip install streamlit
```

```
streamlit --version
```

Después de instalar streamlit con éxito, puede probar la biblioteca ejecutando el siguiente comando en la terminal.

```
streamlit hello
```

3. Desarrollar la aplicación web

En esta parte, implementaremos el modelo de PNL entrenado que predice el sentimiento de la reseña de una película (positiva o negativa). Puede acceder al código fuente y al conjunto de datos aquí.

La aplicación web de ciencia de datos mostrará un campo de texto para agregar la reseña de la película y un botón simple para enviar la reseña y hacer predicciones.

Importar paquetes importantes

El primer paso es crear un archivo de Python llamado app.py y luego importar los paquetes de Python necesarios para el modelo de PNL streamlit y entrenado.

```
# import packages
import streamlit as st
```

```
import os
import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

# text preprocessing modules
from string import punctuation

# text preprocessing modules
from nltk.tokenize import word_tokenize

import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re  # regular expression
import joblib

import warnings

warnings.filterwarnings("ignore")
# seeding
np.random.seed(123)

# load stop words
stop_words = stopwords.words("english")
```

Función para limpiar la revisión

Las reseñas pueden tener palabras y caracteres innecesarios que no necesitamos a la hora de hacer predicciones.

Limpiaremos la reseña eliminando palabras vacías, números y puntuación. Luego convertiremos cada palabra a su forma base utilizando el proceso de lematización en el paquete NLTK.

La función text_cleaning() se encargará de todos los pasos necesarios para limpiar nuestra revisión antes de realizar una predicción.

```
# function to clean the text
@st.cache
def text_cleaning(text, remove_stop_words=True, lemmatize_words=True):
    # Clean the text, with the option to remove stop_words and to lemmatize
word

    # Clean the text
    text = re.sub(r"[^A-Za-z0-9]", " ", text)
    text = re.sub(r"\'s", " ", text)
    text = re.sub(r"http\S+", " link ", text)
    text = re.sub(r"\b\d+(?:\.\d+)?\s+", "", text)  # remove numbers

    # Remove punctuation from text
    text = "".join([c for c in text if c not in punctuation])

    # Optionally, remove stop words
    if remove_stop_words:
```

```
        text = text.split()
        text = [w for w in text if not w in stop_words]
        text = " ".join(text)

    # Optionally, shorten words to their stems
    if lemmatize_words:
        text = text.split()
        lemmatizer = WordNetLemmatizer()
        lemmatized_words = [lemmatizer.lemmatize(word) for word in text]
        text = " ".join(lemmatized_words)

    # Return a list of words
    return text
```

Función para hacer predicción

La función de Python llamada make_prediction() realizará las siguientes tareas.

Recibe la reseña y límpiala.
Cargue el modelo de PNL entrenado.
Haz una predicción.
Estima la probabilidad de la predicción.
Finalmente, devolverá la clase predicha y su probabilidad.

```
# functon to make prediction
@st.cache
def make_prediction(review):

    # clearn the data
    clean_review = text_cleaning(review)

    # load the model and make prediction
    model = joblib.load("sentiment_model_pipeline.pkl")

    # make prection
    result = model.predict([clean_review])

    # check probabilities
    probas = model.predict_proba([clean_review])
    probability = "{:.2f}".format(float(probas[:, result]))

    return result, probability
```

Nota: si el modelo de PNL entrenado predice 1, significa Positivo y si predice 0, significa Negativo.

Crear título y descripción de la aplicación

Puede crear el título de su aplicación web y su descripción utilizando el método title() y write() de streamlit.

```
# Set the app title
st.title("Sentiment Analyisis App")
st.write(
    "A simple machine laerning app to predict the sentiment of a movie's review"
)
```

Para mostrar la aplicación web, debe ejecutar el siguiente comando en su terminal.

```
streamlit run app.py
```

Luego verá que la aplicación web aparece automáticamente en su navegador web o puede usar la URL local creada http://localhost:8501.

Cree un formulario para recibir la reseña de una película

El siguiente paso es crear un formulario simple utilizando streamlit. El formulario mostrará un campo de texto para agregar su reseña y debajo del campo de texto, mostrará un botón simple para enviar la reseña agregada y luego hacer una predicción.

```
# Declare a form to receive a movie's review
form = st.form(key="my_form")
review = form.text_input(label="Enter the text of your movie review")
submit = form.form_submit_button(label="Make Prediction")
```

Hacer predicciones y mostrar resultados

Nuestro último fragmento de código es hacer predicciones y mostrar resultados cada vez que un usuario agrega una reseña de una película y hace clic en el botón "hacer predicción" en la sección del formulario.

Después de hacer clic en el botón, la aplicación web ejecutará la función make_prediction() y mostrará el resultado en la aplicación web en el navegador.

```
if submit:
    # make prediction from the input text
    result, probability = make_prediction(review)

    # Display results of the NLP task
    st.header("Results")
```

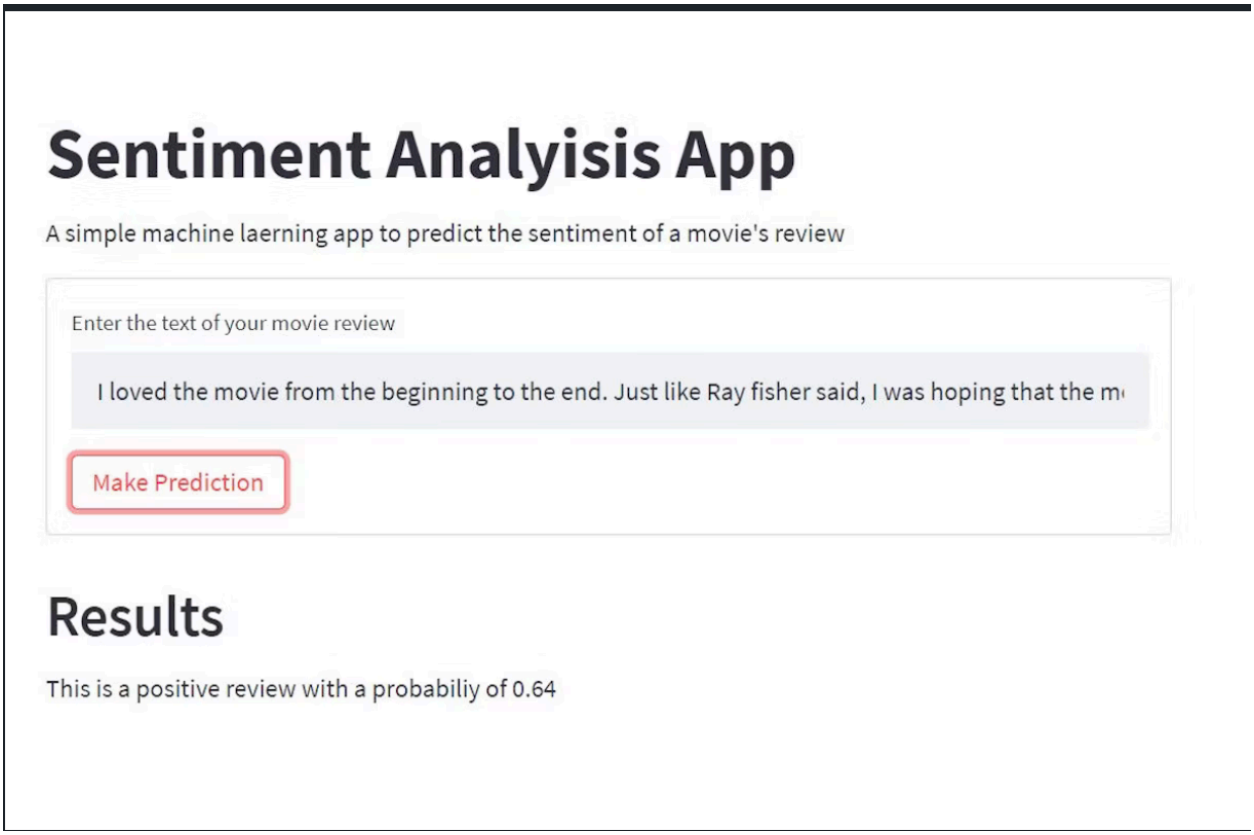
```
    if int(result) == 1:
        st.write("This is a positive review with a probabiliy of ",
probability)
    else:
        st.write("This is a negative review with a probabiliy of ",
probability)
```

4. Pruebe la aplicación web

Con unas pocas líneas de código, hemos creado una aplicación web de ciencia de datos simple que puede recibir una reseña de una película y predecir si es una reseña positiva o negativa.

Para probar la aplicación web, complete el campo de texto agregando una reseña de película de su elección. Agregué la siguiente reseña de la película sobre la película Justice League de Zack Snyder estrenada en 2021.

Luego haga clic en el botón hacer predicción y vea el resultado.



Como puede ver en la aplicación web que hemos creado, el modelo de PNL entrenado predice que la reseña agregada es positiva con una probabilidad de 0,64.

Le recomiendo que agregue otra reseña de la película en la aplicación web de ciencia de datos que hemos creado y la pruebe nuevamente.

5. Conclusión

Hay muchas funciones y componentes de Streamlit que puede utilizar para desarrollar una aplicación web de ciencia de datos de la forma que desee. Lo que ha aprendido aquí son algunos de los elementos comunes de streamlit.