Nombre: Alexandra Cuartas Orozco
CC: 32295342
Asignatura: INTRUCCIÓN A LA INTELIGENCIA ARTIFICIAL
Cohorte: 15
Fecha: 19-01-2024

Actividad No. 1

Titulo: Spam email classification
Resumen: Se intenta clasificar la información de correos entre “ham” y “spam” junto con el mensaje
Origen: https://www.kaggle.com/datasets/ashfakyeafi/spam-email-classification?resource=download
Número total de variables: 5573
Numero de variables cuantitativas: 0
Numero de variables cualitativas: 3 (ham, spam, other)
Variable a predecir: spam
Algoritmo de predicción: modelo de clasificación de Naive Bayes

Actividad 2

Actividad #2
1. Instalar librerías de IA en Python (Sklearn, keras)
2. Seleccionar modelo de aprendizaje.
3. Cargar los datos del conjunto seleccionado en la actividad #1.
4. Preparar las variables de entrenamiento.
5. Realizar proceso de normalización y entrenamiento del modelo.

Modelo 1

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Asegúrate de ajustar la ruta al archivo CSV
csv_path = 'D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\email.csv'
df = pd.read_csv(csv_path)

# Muestra las primeras filas del DataFrame para verificar la carga
print(df.head())

X = df['Message']  # Variable predictora (contenido del correo electrónico)
y = df['Category']  # Variable objetivo (Spam o Ham)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Convierte el texto en vectores de características
vectorizer = CountVectorizer()
```

```
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Crea un modelo de clasificación Naive Bayes
model = MultinomialNB()

# Entrena el modelo
model.fit(X_train_vectorized, y_train)

# Realiza predicciones en el conjunto de prueba
y_pred = model.predict(X_test_vectorized)

# Calcula la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print(f'Precisión del modelo: {accuracy}')

# Imprime el informe de clasificación
print("Informe de clasificación:\n", classification_report(y_test, y_pred))
```

6. Exportar el modelo de aprendizaje.

```
PS D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA>  d:; cd 'd:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA'; & 'C:\Users\User\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '60000' '--' 'D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\from sklearn.py'
D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\from sklearn.py:8: SyntaxWarning: invalid escape sequence '\D'
  csv_path = 'D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\email.csv'
  Category                                            Message
0      ham  Go until jurong point, crazy.. Available only ...
1      ham                      Ok lar... Joking wif u oni...
2     spam  Free entry in 2 a wkly comp to win FA Cup fina...
3      ham  U dun say so early hor... U c already then say...
4      ham  Nah I don't think he goes to usf, he lives aro...
Precisión del modelo: 0.9847533632286996
C:\Users\User\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\User\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\User\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1497: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Informe de clasificación:

	precision	recall	f1-score	support
ham	0.99	0.99	0.99	958
spam	0.97	0.92	0.95	157
{"mode":"full"	0.00	0.00	0.00	0
accuracy			0.98	1115
macro avg	0.65	0.64	0.65	1115
weighted avg	0.99	0.98	0.99	1115

7. En un documento responder las siguientes preguntas:

- **Precisión del modelo:** la precisión del modelo mirando la fila "accuracy" en el informe de clasificación sería 0.98 o 98%.
- **Variables de entrenamiento:** Las variables de entrenamiento son las características (columnas) que se utilizaron para entrenar el modelo son “Category” y “Message”.
- **Variable predicha:** La variable predicha es la variable que el modelo está tratando de predecir. En este ejemplo, sería la columna de la variable objetivo que es “Category”
- **Hiper parámetros del modelo:** Fue MultinominalNB

Modelo 2

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.feature_extraction.text import TfidfVectorizer

# Asegúrate de ajustar la ruta al archivo CSV
csv_path = 'D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\email.csv'
df = pd.read_csv(csv_path)

# Supongamos que 'feature_columns' son las columnas de características y 'target_column' es la variable objetivo
feature_columns = ['Message']
target_column = 'Category'

# Separa las características (X) y la variable objetivo (y)
X = df[feature_columns]
y = df[target_column]

# Inicializa el vectorizador TF-IDF
tfidf_vectorizer = TfidfVectorizer()
```

```
# Aplica el vectorizador a la columna 'Message'
message_tfidf = tfidf_vectorizer.fit_transform(X['Message'])

# Convierte el resultado en un DataFrame y concaténalo con las características existentes
X_encoded = pd.concat([X, pd.DataFrame(message_tfidf.toarray(),
columns=tfidf_vectorizer.get_feature_names_out())], axis=1)

# Divide los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)

# Identifica las columnas numéricas
numeric_columns = X_encoded.select_dtypes(include=['float64', 'int64']).columns

# Normaliza solo las columnas numéricas (opcional, dependiendo del modelo)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train[numeric_columns])
X_test_scaled = scaler.transform(X_test[numeric_columns])

# Entrena el modelo
model = RandomForestClassifier()
model.fit(X_train_scaled, y_train)

# Realiza predicciones en el conjunto de prueba
y_pred = model.predict(X_test_scaled)

# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print(f'Precisión del modelo: {accuracy}')

# Imprimir informe de clasificación
print("Informe de clasificación:\n", classification_report(y_test, y_pred))
```

6. Exportar el modelo de aprendizaje.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras.
https://aka.ms/PSWindows

PS D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA>  & 'C:\Users\User\AppData\Local\
Programs\Python\Python312\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.python-
2023.22.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '60412' '--' 'D:\Documents\
Fullstack\Intro inteligencia artificial\ProyectoIA\RandomForest.py'
D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\RandomForest.py:9: SyntaxWarning:
invalid escape sequence '\D'
  csv_path = 'D:\Documents\Fullstack\Intro inteligencia artificial\ProyectoIA\email.csv'
Precisión del modelo: 0.9811659192825112
```

Informe de clasificación:

	precision	recall	f1-score	support
ham	0.98	1.00	0.99	958
spam	1.00	0.87	0.93	157
accuracy			0.98	1115
macro avg	0.99	0.93	0.96	1115
weighted avg	0.98	0.98	0.98	1115

7. En un documento responder las siguientes preguntas:

- **Precisión del modelo:** La precisión del modelo es en este caso, la precisión general del modelo es aproximadamente 98.12%.
- **Variables de entrenamiento:** Al procesar la columna 'Message' mediante TF-IDF y combinarla con las características existentes, estas son tus variables de entrenamiento.
- **Variable predicha:** predice la columna 'Category', que contiene etiquetas como 'ham' o 'spam'.
- **Hiperparámetros del modelo:** están utilizando los valores predeterminados solamente.