Procesamiento de Lenguaje Natural (NLP)

Clase 4: Proyecto Integrador - Clasificador de Noticias

Prof. D.Sc. BARSEKH-ONJI Aboud

Faculty of Engineering
Universidad Anáhuac México Sur

17 de octubre de 2025

Agenda de la Clase

1. El Desafío Final: Construir Nuestro Modelo
2. El Flujo de Trabajo de Machine Learning
3. Ejemplo Práctico en MATLAB: Clasificador de Noticias
4. Análisis de Resultados
5. Conclusiones Finales del Curso

Agenda

1. El Desafío Final: Construir Nuestro Modelo
2. El Flujo de Trabajo de Machine Learning
3. Ejemplo Práctico en MATLAB: Clasificador de Noticias
4. Análisis de Resultados
5. Conclusiones Finales del Curso

¿Dónde nos quedamos?

Resumen de la Clase 3

Aprendimos a usar una herramienta pre-entrenada (**VADER**) para realizar Análisis de Sentimientos. ¡Fue rápido y efectivo!

La Limitación

VADER es excelente para saber si un texto es positivo o negativo, pero... ¿Y si queremos clasificar texto en nuestras **propias categorías**?

- Por ejemplo, clasificar correos en "Spam" vs "No Spam".
- O clasificar noticias en "Deportes", "Tecnología", "Finanzas".

Para esto, no podemos usar un modelo genérico. Necesitamos **entrenar nuestro propio modelo** con nuestros propios datos y categorías.

Objetivo: El Proyecto Integrador

Nuestra Misión

Construir, entrenar y evaluar un modelo de Machine Learning de principio a fin que pueda **clasificar reportes de fábrica en diferentes categorías** (ej. "Calidad", "Mantenimiento", "Seguridad").

Se tiene que unir todo lo aprendido:

Clase 1 (Limpieza) + **Clase 2** (Vectorización) $\rightarrow$ **Clase 4** (Entrenamiento y Predicción)

Agenda

1. El Desafío Final: Construir Nuestro Modelo
2. El Flujo de Trabajo de Machine Learning
3. Ejemplo Práctico en MATLAB: Clasificador de Noticias
4. Análisis de Resultados
5. Conclusiones Finales del Curso

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).
2. **Dividir los Datos:** Separar los datos en un conjunto de **entrenamiento** (para enseñar al modelo) y uno de **prueba** (para evaluarlo).

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).
2. **Dividir los Datos:** Separar los datos en un conjunto de **entrenamiento** (para enseñar al modelo) y uno de **prueba** (para evaluarlo).
3. **Preprocesar y Vectorizar:** Aplicar las técnicas de las clases 1 y 2 para convertir el texto de entrenamiento en una matriz TF-IDF.

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).
2. **Dividir los Datos:** Separar los datos en un conjunto de **entrenamiento** (para enseñar al modelo) y uno de **prueba** (para evaluarlo).
3. **Preprocesar y Vectorizar:** Aplicar las técnicas de las clases 1 y 2 para convertir el texto de entrenamiento en una matriz TF-IDF.
4. **Entrenar el Modelo:** Alimentar al algoritmo de Machine Learning con la matriz numérica y las etiquetas para que aprenda los patrones.

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).
2. **Dividir los Datos:** Separar los datos en un conjunto de **entrenamiento** (para enseñar al modelo) y uno de **prueba** (para evaluarlo).
3. **Preprocesar y Vectorizar:** Aplicar las técnicas de las clases 1 y 2 para convertir el texto de entrenamiento en una matriz TF-IDF.
4. **Entrenar el Modelo:** Alimentar al algoritmo de Machine Learning con la matriz numérica y las etiquetas para que aprenda los patrones.
5. **Probar el Modelo:** Usar el modelo entrenado para hacer predicciones sobre los datos de prueba (que nunca ha visto).

El Mapa del Proyecto: 6 Pasos Clave

Todo proyecto de Machine Learning supervisado sigue un flujo de trabajo bien definido. Hoy lo aplicaremos al texto.

1. **Cargar los Datos:** Leer nuestro conjunto de datos (textos y sus etiquetas correctas).
2. **Dividir los Datos:** Separar los datos en un conjunto de **entrenamiento** (para enseñar al modelo) y uno de **prueba** (para evaluarlo).
3. **Preprocesar y Vectorizar:** Aplicar las técnicas de las clases 1 y 2 para convertir el texto de entrenamiento en una matriz TF-IDF.
4. **Entrenar el Modelo:** Alimentar al algoritmo de Machine Learning con la matriz numérica y las etiquetas para que aprenda los patrones.
5. **Probar el Modelo:** Usar el modelo entrenado para hacer predicciones sobre los datos de prueba (que nunca ha visto).
6. **Evaluar el Rendimiento:** Comparar las predicciones del modelo con las etiquetas reales para ver qué tan bien lo hizo.

¿Por qué dividir los datos en Entrenamiento y Prueba?

Analogía del Examen

- El **conjunto de entrenamiento** son los ejercicios que un alumno hace en clase con el profesor. Aprende de ellos.
- El **conjunto de prueba** es el examen final. Contiene preguntas que el alumno nunca ha visto, y su resultado nos dice si *realmente* aprendió a generalizar o si solo memorizó las respuestas de la clase.

Regla de Oro

Nunca se debe evaluar un modelo con los mismos datos con los que fue entrenado. Hacerlo nos daría una falsa sensación de un rendimiento perfecto.

Típicamente se usa una división 70 % para entrenamiento y 30 % para prueba.

Agenda

1. El Desafío Final: Construir Nuestro Modelo
2. El Flujo de Trabajo de Machine Learning
3. Ejemplo Práctico en MATLAB: Clasificador de Noticias
4. Análisis de Resultados
5. Conclusiones Finales del Curso

Paso 1 y 2: Cargar y Dividir los Datos

Objetivo

Usaremos un conjunto de datos de MATLAB llamado `factoryReports.csv`. Contiene descripciones de eventos y su categoría.

```
% 1. Cargar los Datos
data = readtable('factoryReports.csv');
textos = data.Description;
categorias = categorical(data.Category); % Etiquetas a predecir

% 2. Dividir los datos en Entrenamiento (80%) y Prueba (20%)
cvp = cvpartition(categorias, 'Holdout', 0.20);
idxEntrenamiento = training(cvp);
textosEntrenamiento = textos(idxEntrenamiento);
categoriasEntrenamiento = categorias(idxEntrenamiento);

idxPrueba = test(cvp);
textosPrueba = textos(idxPrueba);
categoriasPrueba = categorias(idxPrueba);
```

Paso 3: Preprocesar y Vectorizar

Objetivo

Convertir el **texto de entrenamiento** en una matriz numérica TF-IDF. ¡Esto ya lo sabemos hacer!

```
% Preprocesamos los textos de ENTRENAMIENTO
documentosEntrenamiento = tokenizedDocument(lower(textosEntrenamiento));
documentosEntrenamiento = erasePunctuation(documentosEntrenamiento);
documentosEntrenamiento = removeStopWords(documentosEntrenamiento);
documentosEntrenamiento = normalizeWords(documentosEntrenamiento, 'Style', '
    stem');

% Creamos la bolsa de palabras A PARTIR DE LOS DATOS DE ENTRENAMIENTO
bolsa = bagOfWords(documentosEntrenamiento);

% Creamos la matriz TF-IDF de ENTRENAMIENTO
matrizEntrenamiento = tfidf(bolsa);
```

Punto Clave

Paso 4: Entrenar el Modelo Clasificador

Objetivo

Alimentar a un algoritmo de Machine Learning con nuestros datos para que .ªprenda". Usaremos un clasificador **SVM** (*Support Vector Machine*), un modelo muy potente para texto.

```
% Entrenamos un modelo de clasificacion multiclase (SVM por defecto)
% Le damos la matriz numerica y las etiquetas correctas para que aprenda.
modeloClasificador = fitcecoc(matrizEntrenamiento, categoriasEntrenamiento);

disp('¡Modelo entrenado con exito!');
```

En MATLAB..

La función `fitcecoc` (Fit multiclass ECOC) es un comando de alto nivel que entrena un clasificador robusto.

Paso 5: Probar el Modelo

Objetivo

Ahora, preprocesamos los datos de prueba y le pedimos al modelo que prediga sus categorías.

```
% 1. Preprocesamos los datos de PRUEBA (los que el modelo no ha visto)
documentosPrueba = tokenizedDocument(lower(textosPrueba));
documentosPrueba = erasePunctuation(documentosPrueba);
documentosPrueba = removeStopWords(documentosPrueba);
documentosPrueba = normalizeWords(documentosPrueba, 'Style', 'stem');

% 2. Vectorizamos usando LA MISMA 'bolsa' del entrenamiento
matrizPrueba = tfidf(bolsa, documentosPrueba);

% 3. Hacemos las predicciones
predicciones = predict(modeloClasificador, matrizPrueba);
```

Paso 5: Probar el Modelo

Punto Crítico

Al vectorizar los datos de prueba, debemos usar la **misma 'bolsa' (vocabulario)** que creamos con los datos de entrenamiento para asegurar que las matrices sean compatibles.

Paso 6: Evaluar el Rendimiento

Objetivo

Comparar las `predicciones` del modelo con las `categoriasPrueba` (las respuestas correctas) para medir su rendimiento.

```
% Comparamos las predicciones con las etiquetas reales
correctas = sum(predicciones == categoriasPrueba);
total = numel(categoriasPrueba);
exactitud = correctas / total;

fprintf('La exactitud (accuracy) del modelo es: %.2f%%\n', exactitud * 100);

% Creamos una matriz de confusion para un analisis mas detallado
figure;
confusionchart(categoriasPrueba, predicciones);
title('Matriz de Confusion');
```

La **exactitud** (accuracy) nos dice el porcentaje de veces que el modelo acertó.

Agenda

La Matriz de Confusión: ¿Dónde se equivoca?

La exactitud es un buen número, pero no cuenta toda la historia. La **matriz de confusión** nos muestra exactamente qué aciertos y errores cometió el modelo.

Cómo leerla:

- **La diagonal principal (en azul oscuro):** Muestra los **aciertos**. Por ejemplo, el número en la casilla (Quality, Quality) es cuántas veces un reporte que *era* de calidad fue *predicho* como calidad.
- **Fuera de la diagonal (en azul claro):** Muestra los **errores**. Por ejemplo, el número en la casilla (Maintenance, Quality) es cuántas veces un reporte que *era* de mantenimiento fue *confundido* con uno de calidad.

Agenda

1. El Desafío Final: Construir Nuestro Modelo
2. El Flujo de Trabajo de Machine Learning
3. Ejemplo Práctico en MATLAB: Clasificador de Noticias
4. Análisis de Resultados
5. Conclusiones Finales del Curso

Resumen del estudio de NLP

Los puntos analizados:

- **Clase 1:** Empezamos con texto "sucio" y aprendimos a **limpiarlo y estandarizarlo**.
- **Clase 2:** Convertimos esas palabras limpias en **matrices numéricas (TF-IDF)**, el lenguaje de la IA.
- **Clase 3:** Usamos un modelo pre-entrenado (**VADER**) para realizar nuestra primera tarea práctica: el **Análisis de Sentimientos**.
- **Clase 4:** ¡Construimos nuestro **propio clasificador de texto desde cero**, siguiendo un flujo de trabajo profesional de Machine Learning!