TALLER DE PROCESAMIENTO DE SEÑALES

2do Cuatrimestre 2026 - Trabajo Práctico Nº 8 - Modelos de Lenguaje y Sistemas de Recomendación

---

Se desea crear un sistema para recomendar películas. El proyecto estará separado en tres etapas: un codificador de *embedding* que convierta un título de película en un vector, un buscador que devuelva la película válida más cercana al título ingresado y un filtro colaborativo que recomiende películas afines.

(**a**) *Modelo de Lenguaje:*

1. Descargar la base de datos `movies.csv`. En ella, los usuarios calificaron (del 1 al 5) diferentes películas (0 significa sin calificar).
2. La biblioteca `statisticvectors` posee diferentes embeddings estáticos de palabras; en este trabajo utilizaremos el modelo `neuml/word2vec`. Implementar una *bolsa de palabras* que transforme cualquier *string* en un vector. Los pasos a seguir son:
   - Convertir las mayúsculas en minúsculas.
   - Eliminar caracteres extraños.
   - Unificar espacios en blanco.
   - Convertir el *string* en una lista de palabras.
   - Utilizar el modelo `word2vec` de `statisticvectors`.

(**b**) *Buscador:* Se desea diseñar un buscador de títulos de películas, de manera que si el usuario comete algún error u omisión cuando lo escribe, el buscador pueda entender.

1. Se desea medir que tan parecidos son dos *embeddings*. Para ello, implementar un código que calcule la *similitud coseno*. : La similitud coseno se define como el coseno del ángulo entre dos vectores $\mathbf{SC}(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$.
2. Implementar un buscador utilizando $1-$vecino más cercano (como algoritmo de regresión, donde las etiquetas son los mismos predictores) con la similitud coseno como métrica. Dicho buscador debe, dado un *string* (y su correspondiente *embedding*), predecir la película de la base de datos a la que hace referencia.

(**c**) *Sistema de Recomendación:* Se desea diseñar el sistema de recomendación y utilizarlo para recomendarnos películas.

1. Agregar un usuario a la base de datos con al menos 10 películas calificadas. Utilice el buscador para no tener que escribir los títulos perfectos.
2. Utilizando *gradiente descendente* entrenar un filtro colaborativo. Elegir la dimensión del espacio latente, el multiplicador λ y el *learning-rate* justificando dicha elección.
3. Graficar el riesgo regularizado empírico en función del número de iteraciones.
4. Crear un *rating* ponderando en partes iguales la salida del filtro colaborativo y la calificación media de las películas.
5. Recomendar las 5 películas **no vistas** con más alto *rating* al usuario creado anteriormente.