**UNIVERSIDAD PRIVADA ANTENOR ORREGO FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE SISTEMAS**



* **CURSO : ORGANIZACIÓN Y GESTION DE EMPRESAS**
* **DOCENTE : Carlos Edwin Julca Castillo**
* **ALUMNOS :** 
  + - * **Pacherres Tam Santiago**
      * **Arroyo Teran Peter**
      * **Narvaez Rojas David**

**I. Introducción**

**1.1 Titulo del Proyecto**

Sistema de Recomendación de Películas utilizando Filtrado Colaborativo y Redes Neuronales Artificiales

**1.2 Antecedentes**

En la actualidad, las plataformas digitales ofrecen una gran cantidad de contenidos audiovisuales. Esta abundancia plantea un reto significativo: ¿cómo ayudar a los usuarios a encontrar contenido relevante y personalizado? Ante esta problemática, han surgido los sistemas de recomendación, herramientas que permiten filtrar y sugerir información basándose en el comportamiento del usuario o en características de los productos. Los primeros enfoques utilizaban técnicas básicas como filtros por popularidad o sistemas de puntuación. Con el avance del aprendizaje automático, los modelos actuales han evolucionado hacia algoritmos más complejos, como el **filtrado colaborativo**, los **métodos basados en contenido**, y más recientemente, el uso de **redes neuronales profundas (Deep Learning)** para mejorar la precisión de las recomendaciones.

**1.3 Problema a resolver**

Los usuarios de plataformas con catálogos amplios de películas suelen enfrentarse a la dificultad de encontrar contenido adecuado a sus gustos. Esta sobrecarga de información genera una experiencia ineficiente y reduce el nivel de satisfacción. El problema a resolver en este proyecto es el diseño e implementación de un sistema inteligente que, mediante el análisis de preferencias pasadas y patrones de comportamiento, pueda predecir qué películas son más propensas a interesar a cada usuario.

**1.4 Objetivos**

**Objetivo General:**

Desarrollar un sistema de recomendación de películas basado en técnicas de inteligencia artificial, que combine modelos de filtrado colaborativo con redes neuronales para ofrecer sugerencias personalizadas a los usuarios.

**Objetivos Específicos:**

* Implementar un modelo de recomendación utilizando la librería scikit-surprise y técnicas de filtrado colaborativo.
* Desarrollar una red neuronal artificial capaz de predecir calificaciones de usuario utilizando TensorFlow.
* Evaluar el desempeño de ambos modelos mediante métricas de error como MAE y RMSE.
* Complementar el desarrollo técnico con fundamentos teóricos de redes neuronales, CNN y aprendizaje por refuerzo

**II. Requerimientos del sistema**

**2.1 Definición del Dominio**

El dominio del presente proyecto se enmarca en el sistema de recomendación de contenido digital, específicamente orientado a películas. Este sistema opera sobre un conjunto de datos extraído de usuarios y sus interacciones con productos audiovisuales (ratings), perteneciente a una plataforma ficticia de entretenimiento. En este contexto, el objetivo principal es identificar patrones de preferencia entre los usuarios para predecir qué películas les pueden interesar, en función de sus valoraciones pasadas y las de usuarios con gustos similares.

**2.2 Determinación de Requisitos (Requerimientos)**

**Requisitos Funcionales:**

* El sistema debe permitir predecir la calificación que un usuario le daría a una película no vista.
* El sistema debe recomendar películas en función de las calificaciones previas del usuario.
* El sistema debe entrenarse con datos históricos de interacciones usuario-película.
* El sistema debe incorporar al menos dos enfoques de recomendación (filtrado colaborativo y red neuronal artificial).
* El sistema debe mostrar resultados de prueba y evaluación.

**Requisitos No Funcionales:**

* El modelo debe tener un tiempo de respuesta aceptable (<1 segundo por predicción).
* El sistema debe tener un error de predicción (RMSE o MAE) dentro de rangos aceptables para modelos básicos de recomendación.
* El sistema debe ser modular, reutilizable y documentado.

**III. PRE-PROCESAMIENTO y Normalización (Planteamiento del Data-Set; Aprendizaje Supervisado o Aprendizaje no-Supervisado)**

**3.1 Medidas, Datos, Bases de Datos y Elaboración del Data-Set**

El sistema utiliza como fuente de datos el conjunto de archivos provenientes del dataset **MovieLens**, ampliamente utilizado en sistemas de recomendación. Se hace uso de los siguientes archivos:

* movies.csv: contiene los títulos y géneros de las películas.
* ratings.csv: incluye las calificaciones que los usuarios han otorgado a las películas.
* tags.csv: contiene etiquetas personalizadas que los usuarios asocian a las películas.
* links.csv: mapea las películas a identificadores externos (como IMDb o TMDb).

Para construir el dataset principal, se realizó una **unión entre ratings.csv y movies.csv** mediante el campo movieId, eliminando columnas irrelevantes como timestamp. Este proceso permite contar con un dataframe base con las columnas userId, movieId, rating y title, que es usado tanto para el filtrado colaborativo como para el modelo neuronal.

**3.2 Normalización y/o Filtrado de Datos**

Como parte del preprocesamiento:

* Se eliminaron registros con datos faltantes (aunque en MovieLens no es común).
* Se transformaron los IDs de usuario y película en valores numéricos consecutivos para facilitar su uso en la red neuronal.
* En el modelo neuronal, las entradas se normalizan y escalan adecuadamente para evitar sesgos durante el entrenamiento.

Además, se filtraron las películas con pocas calificaciones, ya que una baja densidad de interacciones puede afectar negativamente la calidad del modelo.

**3.3 Planteamiento de:**

El dataset final se dividió en tres subconjuntos para garantizar una correcta evaluación del modelo:

* **Dataset de entrenamiento (training set)**: Se utilizó el 80% de los datos combinados de ratings.csv y movies.csv para entrenar los modelos de aprendizaje supervisado, como la red neuronal (MLPRegressor) y el clasificador binario (LogisticRegression)
* **Dataset de pruebas (test set)**: El 20% restante se reservó para evaluar el desempeño de los modelos sobre datos no vistos, calculando métricas como el MAE, MSE y accuracy
* **Set de validación por Cross-Validation**: Para el modelo colaborativo implementado con Surprise.KNNBasic, se aplicó validación cruzada utilizando cross\_validate() con 5 particiones (KFold), con el objetivo de evitar sobreajuste y evaluar la capacidad de generalización del sistema.

**IV. Aprendizaje del modelo del sistema**

**4.1 Planteamiento del Modelo de Aprendizaje.**

Para el desarrollo del sistema de recomendación se plantearon **dos enfoques complementarios** de aprendizaje supervisado:

* **Filtrado colaborativo basado en descomposición matricial**, utilizando el algoritmo SVD (Singular Value Decomposition) implementado en la librería scikit-surprise. Este modelo parte de la interacción entre usuarios y películas para predecir nuevas calificaciones a partir de patrones latentes.
* **Red neuronal artificial de retropropagación (backpropagation)** desarrollada con TensorFlow. El modelo toma como entrada el par (usuario, película) y aprende a predecir el rating, usando capas densas y funciones de activación ReLU.

Ambos modelos aprenden a partir de calificaciones reales otorgadas por los usuarios y son evaluados usando métricas estándar del aprendizaje supervisado.

**4.2 Desarrollo e Implementación del Modelo**

**Filtrado colaborativo con scikit-surprise:**

* Se utilizó el algoritmo SVD entrenado con validación cruzada de 5 particiones (KFold).
* El modelo aprende factores latentes de usuario y película que explican las calificaciones.
* Se empleó la métrica RMSE (Root Mean Squared Error) para evaluar la calidad de las predicciones.

**Red neuronal artificial:**

* Se codificaron los userId y movieId como variables numéricas.
* El modelo neuronal contiene una capa de entrada con dos neuronas (una por ID), seguida de capas densas ocultas con activación ReLU, y una salida lineal.
* Se utilizó el optimizador Adam y la función de pérdida MSE (error cuadrático medio).
* Se aplicó una división en 80% entrenamiento, 10% validación, 10% prueba.

Esta combinación de enfoques permite comparar resultados y complementar la solució

**V. Comprobación y Despliegue (Deploy) del Sistema**

**5.1 Entrenamiento del Modelo: Uso del Data-Set de entrenamiento**

Ambos modelos fueron entrenados utilizando el **80% del dataset completo**:

* En el modelo de **filtrado colaborativo (SVD)**, el entrenamiento se realizó con la clase Trainset de scikit-surprise, la cual convierte el dataset en una matriz usuario-ítem optimizada.
* En el modelo de **red neuronal**, los datos fueron convertidos a tensores de entrada y normalizados. El entrenamiento se llevó a cabo mediante el método fit() de Keras, utilizando batches y varias épocas para minimizar el error.

**5.2 Ejecución y Pruebas del Modelo**

Se realizaron predicciones utilizando el 10% de datos reservados para prueba:

* En el caso de SVD, se utilizó el método test() para obtener predicciones y medir el rendimiento con RMSE.
* Para la red neuronal, se utilizó evaluate() de TensorFlow sobre el conjunto de test para obtener las métricas MAE y MSE.

Además, se generaron ejemplos de predicción de calificaciones para usuarios específicos con películas que no habían calificado previamente, y se compararon contra valores esperados.

**5.3 Ejecución de la validación del Modelo**

Se emplearon diferentes estrategias de validación:

* **Validación cruzada** (cross\_validate) en el modelo de filtrado colaborativo, lo cual garantiza resultados más estables.
* **Validación directa** (validation\_split) en el modelo neuronal, reservando automáticamente una fracción de los datos para evaluar la generalización durante el entrenamiento.

Ambos enfoques permitieron detectar y mitigar el sobreajuste, y ajustar hiperparámetros como número de neuronas y tasa de aprendizaje.

**5.4 Deploy del APP o Web del Sistema de Predicción, de Clasificación, de Segmentación o Asociación \***

* Utilizar **Streamlit** o **Gradio** para mostrar las recomendaciones al usuario final.
* Exportar el modelo como archivo .h5 (en el caso de la red neuronal) para integrarlo a una interfaz.
* Subir el proyecto completo a un repositorio público en GitHub, incluyendo:  
  + Notebook .ipynb
  + Archivos .csv del dataset
  + Código del modelo
  + Resultados y métricas

Con esto, el sistema está preparado para ser mostrado de forma interactiva o servir como base para desarrollos más complejos.

**VI. Conclusiones**

- El modelo basado en MLP mostró buena capacidad predictiva, y el modelo SVD tuvo resultados sólidos con menor tiempo de entrenamiento.

- La implementación de una interfaz interactiva en Google Colab facilita la usabilidad y demuestra la aplicabilidad práctica del sistema.

- Se validaron diferentes enfoques de aprendizaje (supervisado y no supervisado), lo cual enriqueció la solución.

- Este sistema puede ampliarse fácilmente para integrar nuevos datos, otros tipos de contenidos (series, música) o desplegarse como aplicación web usando Streamlit o Gradio.