Seminario: Analítica con Python

Machine Learning

Profesor: Juan Felipe Nájera

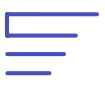
Fecha: Lunes, 27 de octubre de 2025

¿Qué es Machine Learning?

El **Machine Learning** es una rama de la Inteligencia Artificial que permite a los sistemas aprender automáticamente de la experiencia y mejorar sin ser programados explícitamente. Se basa en algoritmos que identifican patrones en grandes volúmenes de datos.

¿Cómo funciona?

Utiliza datos históricos para entrenar modelos predictivos. Estos algoritmos analizan la información, encuentran relaciones complejas y aprenden a realizar tareas específicas sin intervención humana directa.



¿Para qué sirve?

Su objetivo es predecir resultados, clasificar información, automatizar tareas y descubrir insights. Se aplica en sistemas de recomendación, detección de fraude, diagnósticos médicos y optimización de procesos.

Técnicas Clave de Machine Learning

Varios casos de estudio presentan un problema único, que requiere la aplicación de una técnica de Machine Learning específica para obtener resultados óptimos.

A continuación, exploramos las técnicas y sus aplicaciones:

Regresión

La regresión es una técnica de aprendizaje supervisado que modela la relación entre una variable dependiente (el objetivo) y una o más variables independientes (predictores) para predecir un valor continuo.

¿Cómo funciona? Construye un modelo matemático que encuentra la mejor línea (o curva) que se ajusta a los puntos de datos, permitiendo estimar el valor de la variable objetivo para nuevas entradas.

Ejemplos de aplicación:

- Predicción de precios de bienes raíces basándose en características como tamaño, ubicación y número de habitaciones.
- Estimación de la demanda de productos para optimizar inventarios.
- Sistema de Scouting y valaroación de jugadores de fútbol profesional.

Algoritmos comunes: Regresión Lineal, Regresión Polinómica, Regresión de Mínimos Cuadrados Ordinarios (OLS), Regresión de Ridge/Lasso, Regresión por Máquinas de Soporte Vectorial (SVR), Árboles de Decisión para Regresión, Random Forest para Regresión.

Ventajas y casos de uso: Simple de interpretar, útil para comprender las relaciones entre variables y realizar pronósticos continuos. Se aplica en finanzas, economía, salud y muchos otros campos para la evaluación de riesgos y el análisis de tendencias.

Clasificación

La clasificación es una técnica de aprendizaje supervisado que asigna puntos de datos a categorías o clases predefinidas, basándose en un conjunto de datos previamente etiquetados.

¿Cómo funciona? El algoritmo aprende patrones de un conjunto de datos de entrenamiento (con ejemplos etiquetados) para construir un modelo que pueda predecir la clase de nuevos puntos de datos sin etiqueta.

Ejemplos de aplicación:

- Detección de spam: clasificar correos electrónicos como spam o no spam.
- Predicción de popularidad de canciones
- Predicción de retrasos de vuelos
- Predicción de utilidad de reseñas

Algoritmos comunes: Regresión Logística, K-Vecinos más Cercanos (K-NN), Máquinas de Soporte Vectorial (SVM), Árboles de Decisión, Random Forest, Naive Bayes, Redes Neuronales.

Ventajas y casos de uso: Esencial para sistemas de toma de decisiones, filtrado de información y detección de anomalías. Ampliamente utilizada en seguridad, medicina, marketing y procesamiento de lenguaje.

Técnicas Clave de Machine Learning

Varios casos de estudio presentan un problema único, que requiere la aplicación de una técnica de Machine Learning específica para obtener resultados óptimos.

A continuación, exploramos las técnicas y sus aplicaciones:

Clustering

El clustering es una técnica de aprendizaje no supervisado que agrupa puntos de datos similares en conjuntos (clusters) sin la necesidad de etiquetas previas. Su objetivo es encontrar estructuras inherentes en los datos.

¿Cómo funciona? Identifica similitudes entre los puntos de datos y los agrupa de manera que los elementos dentro de un cluster sean más parecidos entre sí que a los de otros clusters.

Ejemplos de aplicación:

- Segmentación de clientes: agrupar clientes con comportamientos de compra similares para campañas de marketing dirigidas.
- Detección de anomalías: identificar transacciones fraudulentas o comportamientos inusuales.
- Agrupación de documentos: organizar grandes colecciones de textos por temas.
- Compresión de imágenes: reducir la cantidad de colores en una imagen.

Algoritmos comunes: K-Means, DBSCAN, Agrupamiento Jerárquico, Modelos de Mezclas Gaussianas (GMM).

Ventajas y casos de uso: Permite descubrir patrones ocultos y obtener insights sin etiquetas preexistentes. Muy utilizado en marketing, biología, ciberseguridad y análisis exploratorio de datos.

Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural (PLN) es una rama de la IA que permite a las computadoras entender, interpretar y manipular el lenguaje humano.

¿Cómo funciona? Combina técnicas de lingüística computacional y modelos de Machine Learning para analizar texto y voz, extrayendo significado, identificando entidades y comprendiendo el contexto.

Ejemplos de aplicación:

- Chatbots y asistentes virtuales: comprender y responder preguntas de los usuarios.
- Traducción automática: traducir texto de un idioma a otro.
- Análisis de sentimiento en redes sociales para entender la opinión pública sobre un producto o marca.
- Análisis de reseñas de hoteles.

Algoritmos comunes: Modelos de Lenguaje (RNN, LSTM, Transformers como BERT y GPT), Naive Bayes (para clasificación de texto), Modelos de Tópicos (Latent Dirichlet Allocation - LDA).

Ventajas y casos de uso: Facilita la interacción humano-computadora, automatiza la gestión de grandes volúmenes de texto y permite extraer valor de datos no estructurados. Indispensable en servicio al cliente, investigación, periodismo y marketing.

Técnicas Clave de Machine Learning

Varios casos de estudio presentan un problema único, que requiere la aplicación de una técnica de Machine Learning específica para obtener resultados óptimos.

A continuación, exploramos las técnicas y sus aplicaciones:

Pronósticos de Series Temporales

Esta técnica se enfoca en predecir valores futuros basándose en la secuencia y patrones de datos históricos que están indexados por el tiempo (series temporales).

¿Cómo funciona? Analiza las tendencias, estacionalidad y los componentes residuales en los datos históricos para proyectar su comportamiento futuro, asumiendo que los patrones pasados continuarán en el futuro.

Ejemplos de aplicación:

- Predicción de ventas futuras para optimización de la cadena de suministro.
- Estimación de la demanda de electricidad o agua para una gestión eficiente de recursos.
- Previsión del tráfico de un sitio web o aplicación.
- Predicción del clima o fluctuaciones del mercado de valores.

Algoritmos comunes: ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), Exponential Smoothing (Holt-Winters), Prophet, Redes Neuronales Recurrentes (RNNs) y LSTMs para secuencias complejas.

Ventajas y casos de uso: Fundamental para la planificación estratégica, la asignación de recursos y la toma de decisiones basada en el futuro. Es crucial en finanzas, comercio minorista, energía y logística.

Sistemas de Recomendación

Los sistemas de recomendación son algoritmos que filtran y predicen las preferencias del usuario para sugerirle elementos (productos, películas, noticias, etc.) que probablemente le interesen.

¿Cómo funciona? Pueden usar enfoques de filtrado colaborativo (basado en el comportamiento de usuarios similares), filtrado basado en contenido (basado en atributos del ítem) o una combinación híbrida para generar sugerencias personalizadas.

Ejemplos de aplicación:

- Recomendaciones de productos en plataformas de e-commerce (Amazon, Mercado Libre).
- Sugerencias de películas y series en servicios de streaming (Netflix, HBO Max).
- Personalización de noticias y artículos en plataformas de contenido (Google News, Facebook).
- Recomendaciones musicales (Spotify).

Algoritmos comunes: Filtrado Colaborativo (basado en usuario/ítem), Factorización Matricial (SVD, ALS), Modelos basados en Redes Neuronales (Deep Learning), Algoritmos de Reglas de Asociación.

Ventajas y casos de uso: Mejoran la experiencia del usuario, aumentan el engagement y las ventas al presentar contenido relevante. Son omnipresentes en plataformas digitales y de comercio electrónico.

Proceso Fundamental de Machine Learning

Aunque cada técnica de Machine Learning tiene sus particularidades, todas comparten un proceso fundamental que las guía desde los datos iniciales hasta una aplicación funcional.

Preparación de Datos

Recolectar, limpiar, transformar y seleccionar las características relevantes de los datos en bruto para asegurar su calidad y formato adecuado para el entrenamiento.

Entrenamiento del Modelo

Alimentar los datos preparados al algoritmo de ML elegido para que aprenda patrones y construya un modelo predictivo o descriptivo.

Evaluación y Despliegue

Evaluar el rendimiento del modelo utilizando métricas relevantes, ajustarlo si es necesario e integrarlo en aplicaciones del mundo real una vez validado.

Proceso estándar de ML Supervisado (Regresión/Clasificación)

Construir un modelo de Machine Learning es un proceso estructurado que implica varias etapas, desde la preparación de los datos hasta el despliegue del modelo.

Estos son los pasos clave a seguir:

Preparar los Datos

- Seleccionar **Features (X)** y **Target (y)**.
- Convertir datos categóricos a numéricos (ej. One-Hot Encoding).
    - Los modelos no entiendo texto.
- Librería clave: `scikit-learn.preprocessing.OneHotEncoder`.

Dividir los Datos (Train/Test Split)

- Separar el conjunto de datos en entrenamiento (ej. 80%) y prueba (ej. 20%).
    - Nunca evaluamos con los mismos datos que usamos para entrenar.
- Crucial para una evaluación imparcial del modelo.
- Librería clave: `scikit-learn.model_selection.train_test_split`.

Elegir y Entrenar el Modelo

- Importar el algoritmo deseado (ej. RandomForestRegressor).
- Ajustar el modelo a los datos de entrenamiento con `.fit(X_train, y_train)`.
    - De esta manera lo entrenamos.
- Librería clave: `scikit-learn.ensemble.RandomForestRegressor`.

Evaluar el Modelo

- Realizar predicciones en el conjunto de prueba con `.predict(X_test)`.
- Comparar predicciones con valores reales usando métricas (ej. RMSE).
    - Esa métrica nos dice, en promedio, cuánto se equivoca el modelo.
- Librería clave: `scikit-learn.metrics.mean_squared_error`.

Guardar el Modelo Entrenado

- Preservar el modelo (y el OneHotEncoder) para su uso futuro en producción (API).
- Asegura la reusabilidad sin re-entrenamiento.
- Librería clave: `joblib.dump()`.

Clustering

El **Clustering** es una técnica de Machine Learning no supervisado cuyo objetivo es encontrar grupos (clústeres) naturales dentro de un conjunto de datos, sin tener etiquetas predefinidas.

Un ejemplo práctico sería: ¿Podemos agrupar los videojuegos en "segmentos de mercado" (ej. "Éxitos de crítica", "Juegos de nicho", "Fracasos populares") usando solo sus características intrínsecas?

El Proceso de Clustering

Definición de Features (X)

Seleccionar las variables que describen los elementos a agrupar (ej. `critic_score`, `user_score`, `total_sales`).

- En clustering, no existe una variable objetivo (`y`).

Preprocesamiento de Datos

- **Escalado:** Usar `StandardScaler` o `MinMaxScaler` para asegurar que todas las variables tengan el mismo peso, crucial para algoritmos como K-Means.
- **Encoding:** Convertir variables categóricas (ej. `genre`) a numéricas con `OneHotEncoder`.

Selección del Algoritmo

- **K-Means:** Rápido y común, asume clústeres esféricos.
- **DBSCAN:** Ideal para formas irregulares y detección de outliers.
- **Jerárquico:** Permite visualizar la anidación de grupos a través de dendrogramas.

Encontrar el Número de Clústeres (K-Means)

- **Método del Codo:** Graficar la inercia (suma de distancias al cuadrado) para diferentes `k` y buscar el punto de inflexión.
- **Coeficiente de Silueta:** Mide la cohesión y separación de los clústeres; un valor cercano a 1 indica clústeres bien definidos.

Entrenamiento y Etiquetado

Ajustar el modelo a los datos preprocesados (ej. `kmeans.fit(X_procesado)`) para obtener las etiquetas de clúster para cada fila (`etiquetas = kmeans.labels_`).

Evaluación y Perfilado

Evaluar con el Coeficiente de Silueta (no hay `y_test`). El paso clave es el **Análisis Cualitativo**: agrupar el DataFrame original por las nuevas etiquetas y analizar las medias o medianas de las features para describir cada clúster.

Ejemplo: "El Clúster 1 tiene `critic_score` alto y `total_sales` bajas (Juegos de Nicho)".

PLN (Procesamiento de Lenguaje Natural)

El Procesamiento de Lenguaje Natural (PLN) se enfoca en convertir texto no estructurado en datos numéricos, permitiendo la aplicación de modelos de Machine Learning para extraer información valiosa y realizar predicciones.

Un ejemplo común es clasificar reseñas de usuarios sobre un producto o servicio como "Positivas", "Negativas" o "Neutrales", un proceso conocido como **Análisis de Sentimiento**.

El Proceso en PLN

Recolección y Definición de Datos

Identificar el corpus de texto (ej. columna `user_reviews`) y definir el problema específico de PLN a resolver (ej. Clasificación de Texto).

Limpieza y Normalización de Texto

- Convertir a minúsculas y eliminar puntuación/números/caracteres especiales.
- **Tokenización:** Dividir el texto en palabras individuales.
- **Stop Words:** Remover palabras comunes sin significado (ej. "el", "la").
- (Opcional) Lematización/Stemming para reducir palabras a su raíz.

Vectorización (Transformación a Números)

Proceso crítico para convertir palabras limpias en una matriz numérica (`X`).

- **CountVectorizer (Bag of Words):** Cuenta la frecuencia de cada palabra.
- **TfidfVectorizer (TF-IDF):** Pondera palabras por su importancia en el documento y en el corpus.

Definición de Variables y División

`X` es la matriz numérica vectorizada y `y` es la etiqueta a predecir (ej. sentimiento).

Aplicar `train_test_split` para dividir los datos en conjuntos de entrenamiento y prueba.

Modelado

Aplicar algoritmos de Machine Learning que funcionan bien con texto vectorizado:

- Naive Bayes (MultinomialNB)
- Regresión Logística
- Random Forest / LightGBM

Evaluación

Evaluar el rendimiento del modelo utilizando métricas de clasificación estándar como:

- Accuracy, Precision, Recall, F1-Score y la Matriz de Confusión.

Series Temporales (Time Series)

El objetivo del análisis de **Series Temporales** es predecir un valor numérico futuro, basándose en sus valores pasados y el orden cronológico. Esto nos permite entender patrones y anticipar eventos.

Un ejemplo práctico podría ser: ¿Cómo predecir las `total_sales` globales para el próximo trimestre, basándose en el historial de ventas?

Proceso Paso a Paso de Análisis de Series Temporales

Preparación de Datos (Indexado)

- Asegurar que la columna de fecha sea de tipo `datetime`.
- Establecer la fecha como el índice del DataFrame (`df.set_index('fecha')`).
- Asegurar la granularidad (ej. re-muestrear a mensual: `df.resample('M').sum()`).
- Verificar y rellenar valores faltantes (ej. `ffill()` o `interpolate()`).

Análisis y Descomposición

Visualizar la serie para identificar sus componentes clave:

- **Tendencia (Trend):** ¿Sube o baja a largo plazo?
- **Estacionalidad (Seasonality):** ¿Existen patrones repetitivos (ej. picos de ventas cada Navidad)?
- **Residuo (Residuals):** El ruido aleatorio o parte inexplicada.

Librería clave: `statsmodels.tsa.seasonal.seasonal_decompose`.

División de Datos (¡Sin Aleatoriedad!)

NUNCA USAR `train_test_split` en series temporales, el orden cronológico es fundamental.

- Los datos de entrenamiento deben ser los más antiguos.
- Los datos de prueba deben ser los más recientes.

Ejemplo: `train = df.loc['2000-01-01':'2014-12-31']`, `test = df.loc['2015-01-01':'2015-12-31']`.

(Modelos Clásicos) Estacionariedad

Muchos modelos (como ARIMA) requieren que la serie sea "estacionaria" (media y varianza constantes a lo largo del tiempo).

- Usar la Prueba de Dickey-Fuller (ADF Test) para verificar la estacionariedad.
- Si no es estacionaria, aplicar diferenciación (`.diff()`) hasta que lo sea.

Modelado

Se pueden usar diversos enfoques:

Modelos Estadísticos:

- `ARIMA`: Usa los retardos (lags) y errores pasados.
- `SARIMA`: Como ARIMA, pero incluye componentes de Estacionalidad.

Modelos de ML (Feature Engineering):

- Crear features basadas en el tiempo: "mes", "año", "día_de_la_semana".
- Crear features de retardo (lags): (ej. ventas_hace_1_mes, ventas_hace_1_año).
- Crear medias móviles (rolling means).
- Con estas features, se puede usar un `LGBMRegressor` o `RandomForestRegressor` normal.

Evaluación

- Entrenar el modelo en el conjunto de `train` y predecir el rango de fechas de `test`.
- Comparar las predicciones (`y_pred`) contra los valores reales (`y_test`).
- Métricas comunes: `RMSE` o `MAE` (que están en la misma escala de la variable original).

Sistemas de Recomendación

Objetivo: Sugerir ítems (ej. videojuegos) relevantes a un usuario, basándose en sus preferencias o las de usuarios similares.

Ejemplo: "Usuarios que compraron 'Wii Sports' también compraron..." o "Basado en tu gusto por 'Halo', te recomendamos...".

Definición del Enfoque y Datos

¿Qué datos tenemos?

- **Explícitos:** Ratings, puntuaciones (ej. `user_score`).
- **Implícitos:** Compras, clics, tiempo jugado (ej. `total_sales`).

¿Tenemos datos por usuario?

Enfoque 1: Filtrado Basado en Contenido

Lógica: "Si te gustó X, te recomiendo Y, porque Y es similar a X".

- Vectorizar características del ítem (ej. `genre`, `developer`, `platform`).
- Calcular **Similitud de Coseno** (`TfidfVectorizer`).
- Recomendar los 10 ítems más similares.

Enfoque 2: Filtrado Colaborativo

Lógica: "Si al Usuario A le gustaron los mismos juegos que a ti, y a A le gustó un juego que tú no has visto, te lo recomiendo".

Requisito: Matriz Usuario-Ítem (filas=usuarios, columnas=juegos, valor=rating).

- Encontrar usuarios similares.
- Recomendar ítems de usuarios similares que tú no hayas visto.
- (Avanzado) **Factorización de Matrices (SVD):** Descubre "factores latentes".

Evaluación del Modelo

- **Offline:** `train_test_split`, medir **RMSE**.
- **Ranking:** Precision@K / Recall@K.
- **Online:** Pruebas A/B, Tasa de Clics (CTR), Tasa de Conversión.

¿Preguntas?