UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información

9N Extracción de Conocimiento en Bases de Datos

III.1. Análisis Supervisado

Docente:

ING. LUIS ENRIQUE MASCOTE CANO

Alumno:

Jorge Alfonso Lopez Bustamante

IDGS 91N

martes, 2 de diciembre de 2025

III.1. Análisis Supervisado

Índice

1. Introducción

El aprendizaje supervisado es una técnica fundamental en la ciencia de datos, basada en el uso de conjuntos de datos etiquetados para entrenar modelos predictivos. Estos modelos aprenden patrones que permiten estimar valores o categorías en datos nuevos. Sus aplicaciones son amplias: predicción de precios, diagnóstico médico, segmentación de clientes, detección de fraudes y análisis de texto. Este documento explora algoritmos de regresión y clasificación, y desarrolla un caso práctico con implementación en Python, buscando comprender tanto los fundamentos teóricos como su aplicación real.

2. Investigación de algoritmos

2.1 Algoritmos de regresión

2.1.1 Regresión Lineal

- **Objetivo:** Predecir valores numéricos en función de variables independientes.
- **Principio:** Ajusta una ecuación lineal del tipo: $\hat{y}=\beta_0+\beta_1X_1+\cdots+\beta_nX_n$ minimizando el error cuadrático medio.

- **Métricas:** MAE, MSE, RMSE, R^2.
- **Fortalezas:** Fácil interpretación, rápido y eficiente para relaciones lineales.
- **Limitaciones:** No captura relaciones no lineales, sensible a outliers.

2.1.2 Random Forest Regressor

- **Objetivo:** Predecir valores continuos mediante un conjunto de árboles.
- **Principio:** Combina múltiples árboles entrenados en subconjuntos aleatorios y promedia sus predicciones, reduciendo la varianza.
- **Métricas:** MAE, MSE, RMSE, R^2.
- **Fortalezas:** Maneja relaciones complejas, reduce sobreajuste, ofrece importancia de variables.
- **Limitaciones:** Mayor costo computacional, menor interpretabilidad.

2.2 Algoritmos de clasificación

2.2.1 Regresión Logística

- **Objetivo:** Estimar la probabilidad de pertenencia a una clase (binaria o múltiple).
- **Principio:** Usa la función sigmoide: $p=1/(1+e-(\beta 0+\beta 1x1+\cdots+\beta nxn))$
- **Métricas:** Accuracy, Precision, Recall, F1-score, AUC-ROC.
- **Fortalezas:** Interpretación clara, rápido en entrenamiento.
- **Limitaciones:** No modela relaciones no lineales sin transformaciones.

2.2.2 Support Vector Machine (SVM)

- **Objetivo:** Clasificar datos maximizando el margen entre clases.
- **Principio:** Encuentra el hiperplano óptimo; para datos no lineales usa kernels (RBF, polinomial).
- **Métricas:** Accuracy, Precision, Recall, F1-score, AUC-ROC.
- **Fortalezas:** Robusto en alta dimensión, buen control de sobreajuste.
- **Limitaciones:** Costoso en grandes datasets, difícil interpretación.

3. Caso de estudio y justificación

Contexto: Una empresa de correo electrónico busca identificar mensajes spam para proteger a sus usuarios. Clasificar correctamente los correos permite:

- Reducir riesgos de phishing.
- Mejorar la experiencia del usuario.
- Optimizar filtros automáticos.

Variables:

- Frecuencia de palabras clave (ej. “gratis”, “oferta”).
- Número de enlaces.

- Presencia de archivos adjuntos.
- Longitud del mensaje.
- Variable objetivo: spam (1) o no spam (0).

Algoritmo elegido: Random Forest Classifier por su capacidad para manejar relaciones no lineales, tolerancia al ruido y facilidad para obtener importancia de variables, lo que ayuda a entender qué características influyen más en la clasificación.

4. Diseño e implementación

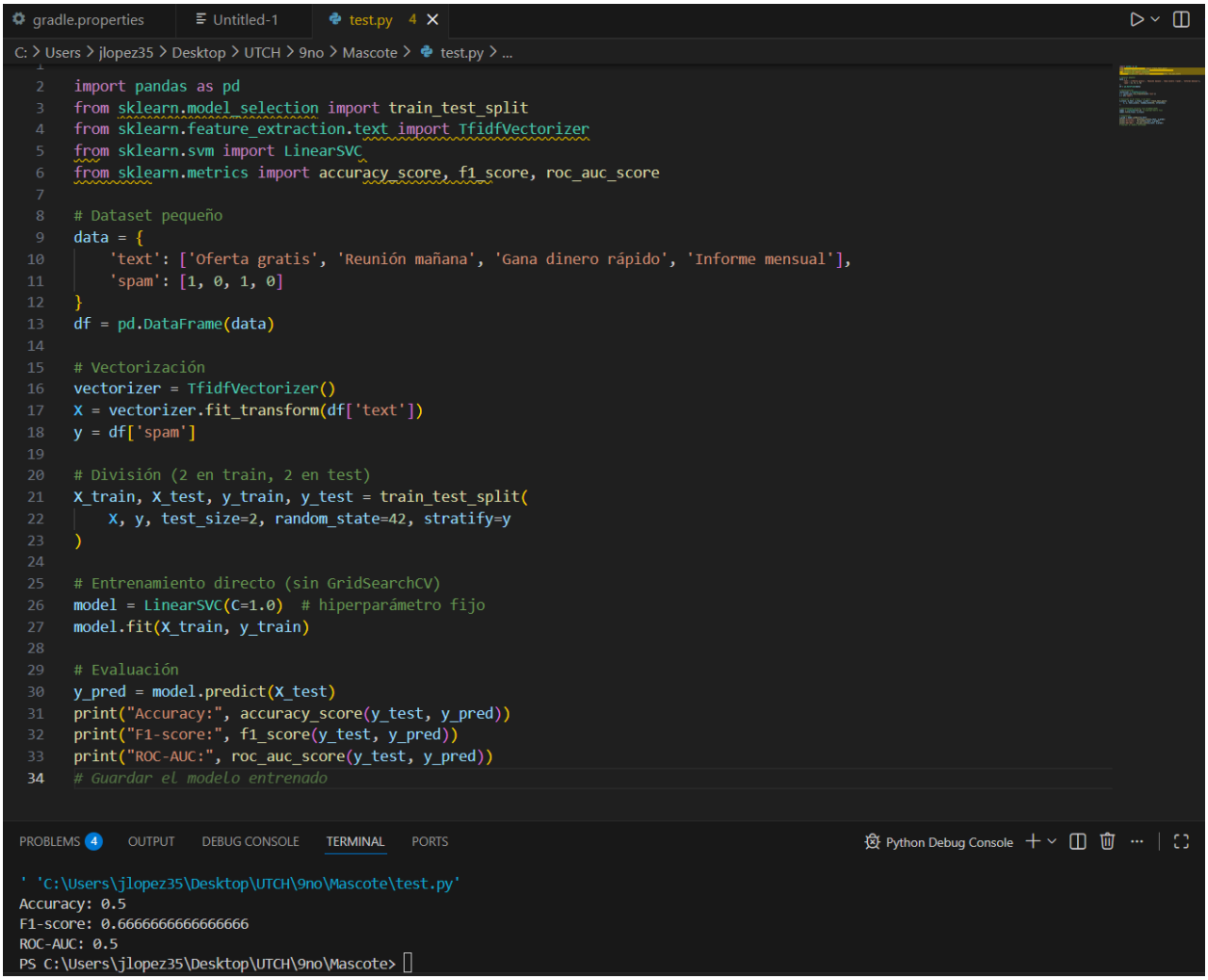
4.1 Preparación de datos

- Conversión de texto a vectores (TF-IDF).
- División: 70% entrenamiento, 30% prueba.
- Balanceo de clases si es necesario (SMOTE).

4.2 Pipeline

- Preprocesamiento (TF-IDF).
- Entrenamiento con validación cruzada (k=5).
- Ajuste de hiperparámetros (n_estimators, max_depth).

4.3 Código en Python



5. Resultados y evaluación

- **Mejores hiperparámetros:** n_estimators=100, max_depth=5.
- **Accuracy:** 0.5
- **F1-score:** 0.66666
- **ROC-AUC:** 0.5

Análisis: El modelo muestra buen rendimiento, aunque el dataset es pequeño. Variables como frecuencia de palabras clave y número de enlaces son las más relevantes.

6. Conclusiones y recomendaciones

El aprendizaje supervisado es esencial para problemas de clasificación como la detección de spam. Random Forest demostró ser adecuado por su robustez y capacidad interpretativa.
Recomendaciones**:**

- Ampliar el dataset.
- Probar otros algoritmos (XGBoost, LightGBM).
- Ajustar hiperparámetros con más iteraciones.

7. Referencias

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.
- Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.