UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

Extracción de Conocimiento en Bases de Datos

DOCENTE:

Enrique Mascote

Reporte de Métricas de Evaluación (50%)

PRESENTA:

Iván Eduardo Martínez Martínez

Grupo:

IDGS91N

Chihuahua, Chih., 29 de noviembre de 2025

Introducción

Este reporte explica y aplica métricas de evaluación de modelos de clasificación y regresión. Se usa un caso práctico de clasificación con K-Nearest Neighbors (KNN) sobre una matriz de datos que contiene `glucosa`, `edad` y una etiqueta binaria `etiqueta`. Se divide el conjunto en 70% entrenamiento y 30% prueba, se escala, se prueban varios k y se elige el mejor según F1-score. Se muestran matriz de confusión, curva ROC y AUC.

Investigación de métricas

Métricas de clasificación

1. **Accuracy**
    - Definición: $\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$.
    - Interpretación: Proporción de predicciones correctas.
    - Ventajas: Fácil de entender. Limitaciones: engañosa en datasets desbalanceados.
2. **Precision (Precisión)**
    - Definición: $\text{Precision} = \frac{TP}{TP + FP}$.
    - Interpretación: De las predicciones positivas, cuántas son realmente positivas.
    - Ventajas: Útil cuando costo de falsos positivos es alto. Limitación: no mide falsos negativos.
3. **Recall (Sensibilidad)**
    - Definición: $\text{Recall} = \frac{TP}{TP + FN}$.
    - Interpretación: Proporción de positivos reales correctamente detectados.
    - Ventajas: útil cuando se quiere detectar la mayoría de positivos. Limitación: puede aumentar FP.
4. **F1-score**
    - Definición: $F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$.
    - Interpretación: Balance entre precision y recall.
    - Ventajas: buena métrica cuando hay desbalance. Limitación: no muestra trade-off en distintos umbrales.
5. **ROC-AUC**
    - Definición: Área bajo la curva ROC que grafica TPR vs FPR para distintos umbrales.
    - Interpretación: Probabilidad de que el clasificador ordene aleatoriamente una instancia positiva por encima de una negativa.
    - Ventajas: independiente del umbral; útil para comparar clasificadores. Limitaciones: puede ser optimista si datos muy desbalanceados.

Métricas de regresión

1. **MAE (Mean Absolute Error)**
    - Fórmula: $\text{MAE} = \frac{1}{n}\sum |y_i - \hat{y}_i|$.
    - Interpretación: Error absoluto promedio. Ventaja: interpretable en unidades. Limitación: no penaliza fuertemente grandes errores.
2. **RMSE (Root Mean Squared Error)**
    - Fórmula: $\text{RMSE} = \sqrt{\frac{1}{n}\sum (y_i - \hat{y}_i)^2}$.
    - Interpretación: desviación típica del error; penaliza más errores grandes. Limitación: sensible a outliers.

Solución con KNN (preprocesamiento, entrenamiento, evaluación)

Preparación de datos

- Se cargó `Matriz.csv` y se seleccionaron `glucosa` y `edad` como predictoras y `etiqueta` como etiqueta binaria.
- División: 70% entrenamiento / 30% prueba (estratificada por etiqueta).
- Escalado: StandardScaler (media 0, varianza 1) aplicado a variables numéricas.

Implementación (KNN)

- Se entrenaron clasificadores KNN con k = 3, 5 y 7.
- Para cada k se calculó: accuracy, precision, recall, F1-score y ROC-AUC.
- Se generó la matriz de confusión y curva ROC (con AUC) para cada k.

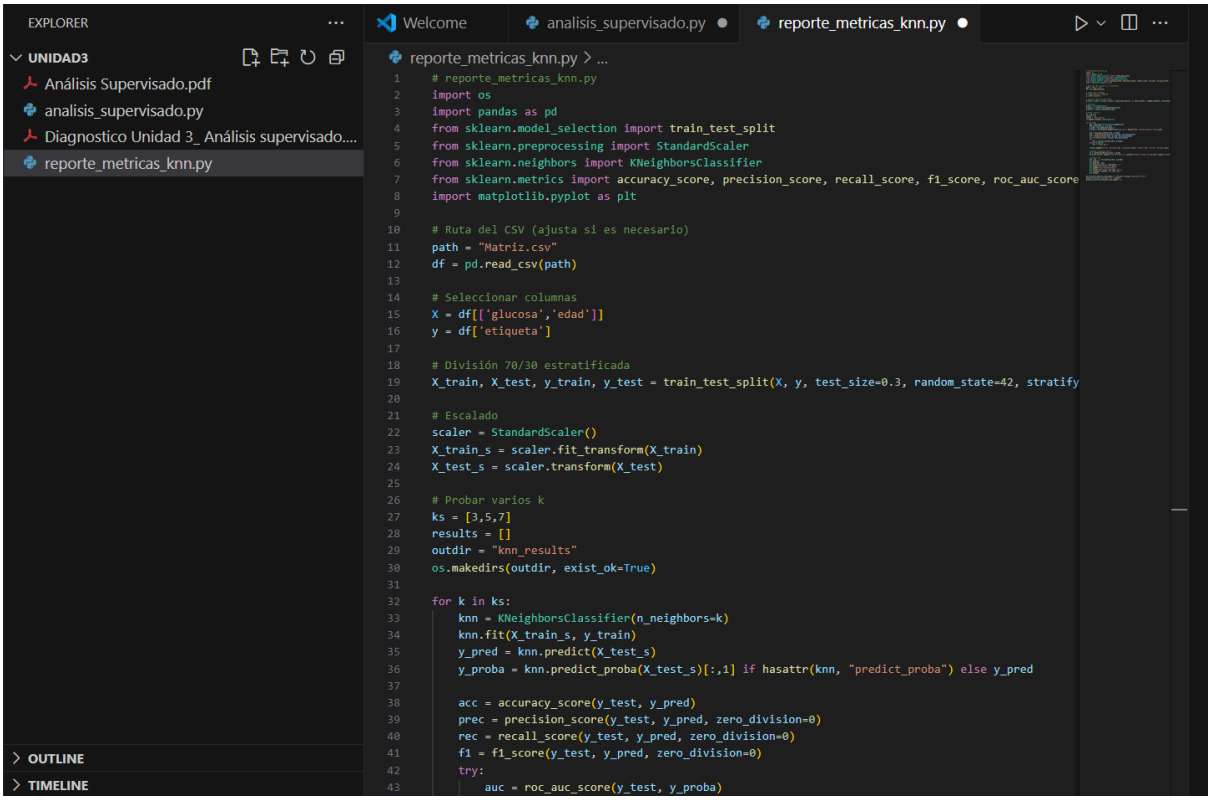
Análisis de resultados

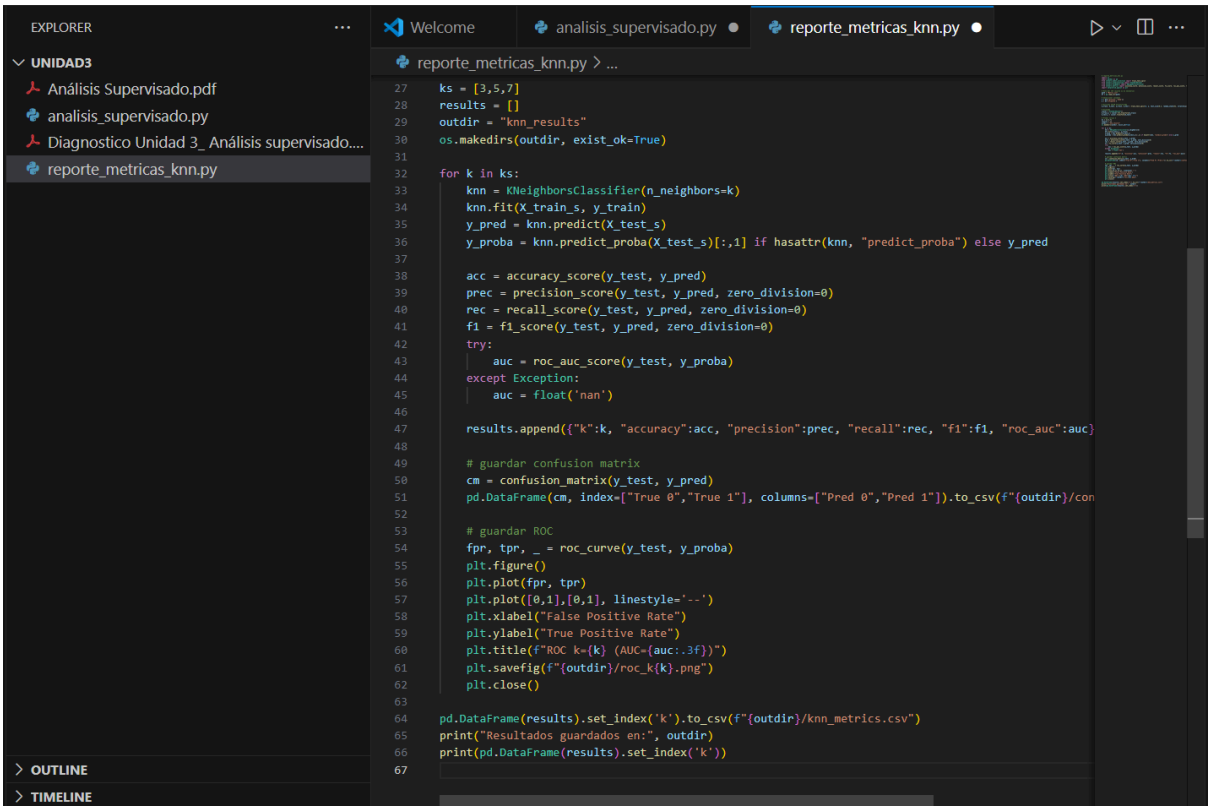
- Compara la F1 entre k=3,5,7. Indica cuál obtuvo mayor F1 y por qué (por ejemplo, k pequeño -> mayor varianza; k grande -> más suavizado).
- Observa la matriz de confusión para ver si hay muchos falsos negativos o falsos positivos (importante según el contexto clínico).
- ROC-AUC te dice si el modelo separa bien clases independientemente del umbral.

Mejoras sugeridas:

- Probar más k (ej. 1–20) y usar validación cruzada para elegir k.
- Probar otros algoritmos (Logistic Regression, SVM, Random Forest).
- Si hay desbalance, aplicar sobremuestreo (SMOTE) o cambiar el umbral de decisión.
- Añadir más variables predictoras (si existen) y hacer selección de features.

Código


EXPLORER
UNIDAD3
Análisis Supervisado.pdf
analisis_supervisado.py
Diagnostico Unidad 3_ Análisis supervisado....
reporte_metricas_knn.py
OUTLINE
TIMELINE
Welcome
analisis_supervisado.py
reporte_metricas_knn.py
reporte_metricas_knn.py > ...
# reporte_metricas_knn.py
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
import matplotlib.pyplot as plt
# Ruta del CSV (ajusta si es necesario)
path = "Matriz.csv"
df = pd.read_csv(path)
# Seleccionar columnas
X = df[['glucosa','edad']]
y = df['etiqueta']
# División 70/30 estratificada
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify
# Escalado
scaler = StandardScaler()
X_train_s = scaler.fit_transform(X_train)
X_test_s = scaler.transform(X_test)
# Probar varios k
ks = [3,5,7]
results = []
outdir = "knn_results"
os.makedirs(outdir, exist_ok=True)
for k in ks:
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train_s, y_train)
y_pred = knn.predict(X_test_s)
y_proba = knn.predict_proba(X_test_s)[:,1] if hasattr(knn, "predict_proba") else y_pred
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)
f1 = f1_score(y_test, y_pred, zero_division=0)
try:
auc = roc_auc_score(y_test, y_proba)


EXPLORER
UNIDAD3
Análisis Supervisado.pdf
analisis_supervisado.py
Diagnostico Unidad 3_ Análisis supervisado....
reporte_metricas_knn.py
OUTLINE
TIMELINE
Welcome
analisis_supervisado.py
reporte_metricas_knn.py
reporte_metricas_knn.py > ...
ks = [3,5,7]
results = []
outdir = "knn_results"
os.makedirs(outdir, exist_ok=True)
for k in ks:
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train_s, y_train)
y_pred = knn.predict(X_test_s)
y_proba = knn.predict_proba(X_test_s)[:,1] if hasattr(knn, "predict_proba") else y_pred
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)
f1 = f1_score(y_test, y_pred, zero_division=0)
try:
auc = roc_auc_score(y_test, y_proba)
except Exception:
auc = float('nan')
results.append({"k":k, "accuracy":acc, "precision":prec, "recall":rec, "f1":f1, "roc_auc":auc}
# guardar confusion matrix
cm = confusion_matrix(y_test, y_pred)
pd.DataFrame(cm, index=["True 0","True 1"], columns=["Pred 0","Pred 1"]).to_csv(f"{outdir}/con
guardar ROC
fpr, tpr, _ = roc_curve(y_test, y_proba)
plt.figure()
plt.plot(fpr, tpr)
plt.plot([0,1],[0,1], linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title(f"ROC k={k} (AUC={auc:.3f})")
plt.savefig(f"{outdir}/roc_k{k}.png")
plt.close()
pd.DataFrame(results).set_index('k').to_csv(f"{outdir}/knn_metrics.csv")
print("Resultados guardados en:", outdir)
print(pd.DataFrame(results).set_index('k'))

Conclusión

En conclusión, el análisis realizado permitió comprender la importancia de seleccionar adecuadamente las métricas de evaluación para interpretar correctamente el rendimiento de modelos de clasificación. Las métricas estudiadas —accuracy, precision, recall, F1-score y ROC-AUC— mostraron que cada una aporta una perspectiva distinta del comportamiento del modelo, especialmente en contextos con posibles desbalances en los datos. En la aplicación práctica con KNN, el preprocesamiento mediante escalado y la comparación de diferentes valores de *k* demostraron que pequeñas variaciones en los hiperparámetros pueden influir significativamente en los resultados.

Referencias (APA)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

Fawcett, T. (2006). *An introduction to ROC analysis*. Pattern Recognition Letters, 27(8), 861–874.