Universidad Nacional Autónoma de México.
IIMAS

Minería de Datos

Semestre 2026-1.

D.I. Ileana Angélica Grave Aguilar

Tarea 1

Integrantes:

- Villalón Pineda Luis Enrique .

Los algoritmos que desarrolle aquí, fueron los mismos que desarrolle durante la practica , para ver si puedo mejorar mis métricas , moviendo hiperparámetros e ir comparando si verdaderamente podemos mejorar los resultados y que afecta. .

Algoritmo	**Suposiciones del banco de datos**	**Ventajas**	**Desventajas**	**Parámetros**
SVM	Datos numéricos y escalados. Clases lineal o casi linealmente separables (dependiendo del kernel).	Robusto frente a sobreajuste en alta dimensión. Eficaz con separación clara. Maximiza el margen.	Sensible a la escala. Difícil de interpretar. Costoso en tiempo con grandes datasets.	Variables predictoras (X) y etiquetas (y).
KNN	Variables numéricas o codificadas. Distancias deben tener sentido. Sin supuestos estadísticos.	Sencillo e intuitivo. No requiere entrenamiento. Funciona con fronteras no lineales.	Costoso en predicción. Sensible a escala y ruido. Bajo rendimiento con muchos datos.	X, y, métrica de distancia.
Árboles de Decisión	Variables numéricas o categóricas. No requiere normalización.	Interpretables. Manejan variables mixtas. No necesitan escalado.	Tienden al sobreajuste. Sensibles a pequeñas variaciones.	X, y.
Random Forest	No requiere normalización. Variables parcialmente correlacionadas. Dataset grande.	Reduce sobreajuste. Alta precisión. Estima importancia de variables.	Menos interpretable. Requiere más memoria.	X, y.
XGBoost	Datos numéricos o categóricos codificados. No requiere normalización estricta.	Muy preciso. Control de regularización. Maneja datos faltantes.	Difícil de ajustar. Riesgo de sobreajuste. Menos interpretable.	X, y.
Regresión Logística	Independencia de observaciones. Relación lineal entre variables y log-odds. Sin multicolinealidad fuerte.	Fácil de interpretar. Rápida y eficiente. Proporciona probabilidades de clase.	No modela relaciones no lineales. Sensible a multicolinealidad.	X, y.

Tabla 1: Suposiciones, ventajas, desventajas y parámetros de los principales algoritmos de clasificación.

Algoritmo	Hiperparámetros	Efecto de los hiperparámetros
SVM	C, kernel, gamma, degree.	**C**: penaliza errores; valores bajos mayor margen, menos sobreajuste. **kernel**: tipo de frontera (lineal, RBF, polinomial, sigmoide). **gamma**: alcance de influencia de puntos; valores altos sobreajuste. **degree**: grado del polinomio si se usa kernel polinomial.
KNN	n_neighbors, weights, metric, p.	**n_neighbors**: número de vecinos; valores bajos sobreajuste, altos subajuste. **weights**: 'uniform' (todos igual) o 'distance' (mayor peso a cercanos). **metric**: tipo de distancia (euclidiana, manhattan...). **p**: parámetro de Minkowski (p=1 manhattan, p=2 euclidiana).
Árboles de Decisión	max_depth, min_samples_split, min_samples_leaf, criterion.	**max_depth**: profundidad máxima; valores bajos reducen sobreajuste. **min_samples_split**: mínimo de muestras para dividir nodo; alto valor regulariza. **min_samples_leaf**: mínimo en hoja; alto valor suaviza modelo. **criterion**: función de impureza ('gini', 'entropy').
Random Forest	n_estimators, max_depth, min_samples_split, max_features, bootstrap.	**n_estimators**: número de árboles; alto valor menor varianza, mayor costo. **max_depth**: controla profundidad (más profundo más complejo). **min_samples_split / leaf**: igual que en árbol simple. **max_features**: nº de variables por división; bajo más diversidad. **bootstrap**: si usa muestreo con reemplazo.
XGBoost	n_estimators, learning_rate, max_depth, subsample, colsample_bytree, gamma, lambda, alpha.	**n_estimators**: número de árboles; alto valor más ajuste. **learning_rate**: tamaño de paso; bajo aprendizaje lento pero más preciso. **max_depth**: profundidad de cada árbol. **subsample**: fracción de datos usada por árbol; bajo menos sobreajuste. **colsample_bytree**: fracción de variables; bajo mayor robustez. **gamma**: penaliza divisiones adicionales. **lambda**, **alpha**: regularización L2 y L1.
Regresión Logística	penalty, C, solver, max_iter.	**penalty**: tipo de regularización ('l1', 'l2', 'elasticnet'). **C**: inverso de la regularización; bajo mayor penalización. **solver**: método de optimización ('liblinear', 'lbfgs'...). **max_iter**: número máximo de iteraciones; alto asegura convergencia.

Tabla 2: Hiperparámetros de los algoritmos de clasificación y efecto de su variación.