AAI - Tema 3: Métodos simples de aprendizaje automático supervisado - Teoría

Naive Bayes

- Ficha
  - Necesidad de escalado: no
  - Supervisión: sí
  - Rapidez: alta
  - Interpretabilidad: alta
  - Uso: clasificación
  - Tamaño datos necesarios: Pequeño/medio
  - Escalabilidad: alta
  - Núm. características permitidas: alta
  - Probabilidades: sí (nativo)
  - Conocimiento: basado en modelo
- Puntos fuertes
  - No necesita escalado
  - Muy rápido
- Limitaciones
  - Fuertes suposiciones
  - No sirve para regresión
- Teoría
  - Aplica el teorema de Bayes a etiquetas vs características
  - Supone calcular $P(\text{características}|\text{etiqueta}) \Rightarrow$ modelo generativo
  - Supone suposiciones "ingenuas" para calcular modelo generativo
- Tipos
  - Gaussian Naive Bayes
    - Suposición: características se generan de distribución gaussiana (elipses)
    - Uso: continuas
    - Implementación en Scikit-learn: `skl.naive_bayes.GaussianNB`
  - Multinomial Naive Bayes
    - Suposición: características se generan de distribución multinomianl simple
    - Usa: texto
    - Implementación en Scikit-learn: `skl.naive_bayes.MultinomialNB`
- Hiperparámetros: pocos
- Características
  - Buen primer intento, por su rapidez e interpretabilidad
  - Escala bien a muchas dimensiones
  - Calcula probabilidades nativas
- Aplicación: detector spam

Teorema de Bayes aplicado a Naive Bayes:

$$P(\text{clases}|\text{características}) = \frac{P(\text{características} \mid \text{clases})\, P(\text{clases})}{P(\text{características})}$$

Se preguntó por esta fórmula en: EX.20230904.T.6, EX.20250902.T.1.

- Bibliografía
  - Capítulo 41 "In Depth: Naive Bayes Classification". En: Python Data Science Handbook. 3.ª ed. O'Reilly.

K-Nearest-Neighbors

- Ficha:
  - Necesidad de escalado: sí (depende de distancias)
  - Supervisión: sí (requiere datos previos)
  - Parametrización: no
  - Tamaño datos necesarios: Pequeño/medio
  - Interpretabilidad: media
  - Conocimiento: basado en instancia
  - Uso: clasificación, regresión
- Puntos fuertes
  - No requiere entrenamiento o reentrenamiento
  - Simplicidad
- Desventajas
  - No escala a instancias
  - No escala a dimensiones (maldición de la dimensionalidad)
  - Sensibilidad a escala
  - Sensibilidad a ruido/outliers
  - Alto uso en memoria
- Características
  - Aprendizaje perezoso
- Usos
  - Clasificación
    - Teoría
      - Dada una instancia, busca los k puntos más cercanos y asigna la clase más repetida
    - Hiperparámetros
      - k
        - k- $\Rightarrow$ overfitting
        - Para elegir k, se prueban todas y toma la de menor MSE
        - El valor de k no puede exceder el número de filas.
    - Implementación en scikit-learn: `neighbors.KNeighborsClassifier`
  - Regresión
    - Implementación en scikit-learn:
      - `KNeighborsRegressor`
        - Usa los k primeros vecinos
      - `RadiusNeighborsRegressor`

        - Usa todos vecinos en un radio r
- Aplicación
  - Sugerencias en base a otros usuarios
- Bibliografía
  - LEE, Wei-Meng. Capítulo 9 "Supervised Learning—Classification Using K-Nearest Neighbors (KNN)". En: Python Machine Learning. Wiley.
  - Apartado 1.6.4 "Regression" de 1.6 "Neighbors" en Scikit-learn User Guide.
    * `KNeighborsRegressor`
    * `RadiusNeighborsRegressor`

Árboles de decisión

- Ficha
  - Necesidad de escalado: no
  - Supervisión: sí
  - Rapidez: alta
  - Interpretabilidad: alta
  - Tamaño datos necesarios: Pequeño/medio
  - Uso: clasificación, regresión
- Fortalezas
  - Distingue modelos no lineales
  - No require ordenamiento
- Desventajas
  - Tiende a sobreentrenar
- Hiperparámetros
  - Profundidad (a +, +overfitting)
  - Núm. instancias por hoja (a +, +underfitting)
- Teoría
  - Para cada punto, se asigna para su clasificación
- Algoritmos
  - CART (hijos = 2)
  - ID3 (hijos $\geq$ 2)
- Medidas de impureza
  - Impuridad de Gini
  - Entropía
- Usos
  - Clasificación
  - Regresión

Profundidad de un árbol

Un árbol binario perfectamente balanceado tiene una profundidad de:

$$\text{profundidad} = \log_2(n),$$

donde n es el número de instancias.

Equivalencias aproximadas de logaritmos (útil para resolverlo sin calculadora):

Base 2	Base 10
2^{10}	10^3
2^{20}	10^6
2^{30}	10^9

Fórmula del logaritmo:

$$\log_b(x) = y \iff b^y = x$$

Aparece en ejercicio AAI.EX.20250204.T.3.

Medidas de impureza

Las métricas de impureza tratan de ser reducidas en cada división o *split* de nodos.

La **impureza de Gini** es una métrica de impureza. Cuando es 0, todas las instancias de entrenamiento del nodo pertenecen a la misma clase.

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

donde

- $p_{i,k}$ es el radio de las instancias de clase k en las instancias de entrenamiento del nodo i.

La **entropía** mide desorden; alcanza su valor mínimo (0) cuando el conjunto es completamente puro y está formado por una sola clase.

Bibliografía:

- Básica
  - GÉRON, Aurélien. Capítulo 6 “Decision trees”. En: HOML. 3.ª ed. O’Reilly. Págs. 195-209