

Proyecto 2: Clasificadores

Programación – Grupo 9292

Profesora: María Fernanda Sánchez Puig

Alumno: Gael Peña Fonseca

Resumen del proceso:

1. Entrenamiento mediante `ent_modelos(self, X_train, Y_train)` :
Se usa `.fit()` para ajustar los parámetros de los modelos con los datos de entrenamiento.
2. Evaluación mediante `ev_modelos(self, X_test, Y_test)`:
Se usa `predict()` para generar predicciones y calcular métricas con datos de pruebas.
3. Resultados organizados:
Métricas como precisión y F1-score se almacenan en `self.resultados`.

¿Qué sucedió?

Se transformaron las etiquetas diagnosis en valores numéricos (M=1, B =0) utilizando un loop básico, posteriormente se seleccionaron características relevantes sin incluir id y diagnosis, como consecuencia los datos se dividieron en 80% entrenamiento y 20% prueba para evaluar el desempeño.

Se uso StandardScaler de sklearn para escalar las características, normalizando valores y evitando sesgos en clasificadores como la Maquina de Soporte Vectorial.

Resultados

Archivos CSV generados:

datos_entrenamiento.csv: contiene datos escalados y etiquetas para entrenamiento.

datos_prueba.csv: Contiene datos escalados y etiquetas para prueba.

Entrenamiento de los clasificadores

Se entrenaron los cinco clasificadores usando `.fit()` de sklearn, el método `.fit()` es el proceso de entrenamiento del modelo, donde, aprende a ajustar parámetros internos para relacionar los datos de entrada(X) con las etiquetas (y)

Clasificadores entrenados:

1. Regresión Logística: Ajustó un modelo lineal para separar las clases.
2. k-NN: Clasificó puntos según los vecinos más cercanos.
3. MSV: Encontró un hiperplano óptimo para dividir las clases.
4. Árbol de Decisión: Generó reglas para clasificar datos.
5. Random Forest: Combinó múltiples árboles de decisión para mejorar precisión.

Resultados:

Todos los clasificadores se entrenaron exitosamente.

Regresión Logística y *k-NN* mostraron buen desempeño durante el entrenamiento.

Evaluación de clasificadores

Se usó `.predict()` de sklearn para hacer predicciones en los datos de prueba.

Una vez entrenado el modelo con `.fit()`, `.predict()` genera predicciones sobre datos nuevos usando los parámetros ajustados.

Resultados:

Regresión Logística y k-NN destacaron con precisión de 0.99.

Otros clasificadores como Árbol de Decisión y Random Forest alcanzaron precisiones de 0.96 y 0.97 respectivamente.

Comparación y selección del mejor modelo

¿Qué se hizo?

1. Se extrajo la precisión de cada clasificador desde el reporte de clasificación.
2. Se compararon las precisiones en un gráfico de barras.
3. El modelo con la mayor precisión fue seleccionado como el mejor.

Resultados:

3. Regresión Logística fue seleccionado como el mejor modelo con precisión de 0.99.

4. Se destacó su capacidad para separar clases linealmente en los datos evaluados