INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
INTELIGENCIA ARTIFICIAL
ING. EN SISTEMAS COMPUTACIONALES

Laboratorio 9: Clasificadores Euclidiano y 1NN

Alumno: Hurtado Morales Emiliano - 2021630390

Maestro: Andrés García Floriano

Grupo: 6CV3

Fecha de entrega: 14/11/2024

Ciclo Escolar: 2025 – 1

Introducción

En el ámbito del aprendizaje automático, la clasificación es una de las tareas supervisadas más fundamentales y ampliamente utilizadas. Su objetivo es asignar etiquetas a nuevas instancias basándose en ejemplos previos conocidos, permitiendo la identificación de patrones y la toma de decisiones informadas. Esta tarea es fundamental en múltiples campos, como el reconocimiento de imágenes, la detección de fraudes, el diagnóstico médico y la categorización de textos, entre otros.

Existen diversos algoritmos de clasificación, cada uno con características y aplicaciones específicas. En este laboratorio, exploraremos dos clasificadores sencillos pero efectivos: el clasificador basado en la **distancia Euclidiana** y el clasificador **1-Nearest Neighbor (1NN)**. Ambos métodos se basan en el concepto de proximidad en un espacio multidimensional, donde las instancias que están más cerca en términos de distancia se consideran similares. Esta proximidad permite inferir la etiqueta de una instancia desconocida en función de sus "vecinos" en el conjunto de datos de entrenamiento.

El clasificador Euclidiano y el clasificador 1NN son algoritmos no paramétricos, lo cual significa que no hacen suposiciones sobre la distribución subyacente de los datos. Esta propiedad los convierte en métodos versátiles y aplicables a una amplia variedad de conjuntos de datos, aunque su simplicidad los hace más sensibles al ruido y a la escala de las características. Estos clasificadores son fáciles de implementar y entender, lo cual los convierte en una excelente introducción a las técnicas de clasificación.

Para evaluar el rendimiento de estos algoritmos, se probarán en tres conjuntos de datos bien conocidos en el ámbito de la ciencia de datos: **Iris**, **Wine** y **Digits**. Estos datasets presentan características distintas en términos de complejidad, dimensionalidad y número de clases, lo que permitirá analizar cómo se comportan los clasificadores en diferentes escenarios. Además, se emplearán tres métodos de validación: **Hold-Out (70/30)**, **10-Fold Cross-Validation** y **Leave-One-Out**, los cuales ayudarán a obtener una evaluación precisa y representativa del rendimiento de cada clasificador.

En este reporte, se presenta una descripción detallada de los algoritmos implementados, los resultados obtenidos en cada conjunto de datos y un análisis de la efectividad de cada método de validación. A través de esta exploración, se busca no solo evaluar el desempeño de los clasificadores Euclidiano y 1NN, sino también entender los beneficios y limitaciones de cada técnica en contextos prácticos de clasificación.

Explicación de los Algoritmos

Clasificador Euclidiano

El **clasificador Euclidiano** se basa en la distancia Euclidiana entre un punto de prueba y los puntos de entrenamiento. La distancia Euclidiana es una medida de similitud que calcula la "proximidad" entre dos puntos en un espacio multidimensional. Este clasificador asigna la clase del punto de entrenamiento más cercano al punto de prueba. Formalmente, la distancia Euclidiana entre dos puntos p y q en un espacio de n dimensiones se define como:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Donde p_i y q_i representan las coordenadas de los puntos en la dimensión i. Este algoritmo es fácil de implementar y funciona bien en problemas de clasificación simples donde las clases están bien separadas en el espacio de características. Sin embargo, puede ser sensible a la escala de los datos y al ruido, por lo que se recomienda preprocesar los datos para estandarizar sus características.

```
# Clasificador Euclidiano basado en el centroide
class ClasificadorEuclidiano:
    """
    Clasificador basado en la distancia al centroide. Calcula el centroide de
cada clase y asigna la
    clase del centroide más cercano a cada instancia de prueba.
    """
    def entrenar(self, X_entrenamiento, y_entrenamiento):
        """
        Calcula el centroide para cada clase en el conjunto de entrenamiento.

        Parámetros:
        - X_entrenamiento: Matriz de características para entrenamiento.
        - y_entrenamiento: Vector de etiquetas de clase para entrenamiento.
        """
        self.centroides = {}
        clases = np.unique(y_entrenamiento)
        for clase in clases:
            # Calcula el centroide como el promedio de los puntos de cada clase
            self.centroides[clase] = X_entrenamiento[y_entrenamiento ==
clase].mean(axis=0)

    def predecir(self, X_prueba):
        """
        Predice las clases para las instancias de prueba basándose en la
distancia al centroide más cercano.

        Parámetro:
```

```
        - X_prueba: Matriz de características para las instancias de prueba.

        Retorna:
        - Array con las etiquetas de clase predichas.
        """
        y_predicho = []
        for x in X_prueba:
            # Calcula la distancia al centroide de cada clase
            distancias = {clase: distance.euclidean(x, centroide) for clase,
centroide in self.centroides.items()}
            # Encuentra la clase del centroide más cercano
            clase_min = min(distancias, key=distancias.get)
            y_predicho.append(clase_min)
        return np.array(y_predicho)
```

Clasificador 1NN (1-Nearest Neighbor)

El **clasificador 1-Nearest Neighbor (1NN)** es un método de clasificación basado en instancias que asigna la clase del vecino más cercano a cada punto de prueba. Al igual que el clasificador Euclidiano, utiliza la distancia Euclidiana para medir la proximidad entre puntos, pero se enfoca en una estructura de "vecindad" donde solo considera el punto más cercano para hacer una predicción. Este enfoque es no paramétrico, lo que significa que no hace ninguna suposición sobre la distribución de los datos. Aunque es simple y efectivo en muchos casos, el 1NN puede ser computacionalmente costoso con conjuntos de datos grandes y es sensible a la presencia de ruido en los datos.

Ambos algoritmos fueron implementados en Python y se evaluaron en tres datasets diferentes, utilizando tres métodos de validación para analizar su precisión y estabilidad.

```
# Clasificador 1NN
class Clasificador1NN:
    """
    Clasificador 1-Nearest Neighbor (1NN). Similar al clasificador Euclidiano,
    asigna la clase del vecino más cercano basándose en la distancia Euclidiana.
    """
    def entrenar(self, X_entrenamiento, y_entrenamiento):
        """
        Guarda el conjunto de entrenamiento.

        Parámetros:
        - X_entrenamiento: Matriz de características para entrenamiento.
        - y_entrenamiento: Vector de etiquetas de clase para entrenamiento.
        """
        self.X_entrenamiento = X_entrenamiento
        self.y_entrenamiento = y_entrenamiento

    def predecir(self, X_prueba):
```

```
        """
        Predice las clases para las instancias de prueba usando el vecino más
cercano.

        Parámetro:
        - X_prueba: Matriz de características para las instancias de prueba.

        Retorna:
        - Array con las etiquetas de clase predichas.
        """
        y_predicho = []
        for x in X_prueba:
            # Calcula la distancia Euclidiana a todas las instancias de
entrenamiento
            distancias = [distance.euclidean(x, x_entreno) for x_entreno in
self.X_entrenamiento]
            # Encuentra el índice de la instancia más cercana
            indice_min = np.argmin(distancias)
            # Asigna la clase de la instancia más cercana
            y_predicho.append(self.y_entrenamiento[indice_min])
        return np.array(y_predicho)
```

Validaciones

Las validaciones son técnicas que se utilizan para medir el rendimiento de los modelos de clasificación en distintos conjuntos de datos. En este laboratorio, se emplearon tres métodos de validación: **Hold-Out (70/30)**, **10-Fold Cross-Validation**, y **Leave-One-Out**. A continuación, se detalla cada uno:

1. **Hold-Out (70/30):**
    - Esta técnica divide el conjunto de datos en dos partes: 70% para el entrenamiento del modelo y 30% para la prueba.
    - Es una de las validaciones más simples, pero su desventaja es que el rendimiento del modelo puede depender de cómo se dividan los datos. Si se utiliza una partición con características poco representativas del conjunto completo, los resultados pueden ser menos confiables.
    - Este método es rápido y útil en datasets grandes, pero puede no ser adecuado para conjuntos de datos pequeños, ya que sacrifica una parte significativa de los datos solo para la prueba.
2. **10-Fold Cross-Validation:**
    - En este método, el conjunto de datos se divide en 10 partes iguales (folds). En cada iteración, una de las partes se utiliza para prueba y las otras nueve para entrenamiento. Este proceso se repite 10 veces, utilizando cada fold una vez como prueba.

- Al final, se calcula el promedio de la precisión y otros resultados para obtener una métrica general.
- Este método es más robusto que Hold-Out, ya que utiliza todas las partes del conjunto de datos para entrenamiento y prueba, lo que reduce la varianza y mejora la representatividad de la evaluación.

3. **Leave-One-Out (LOO):**
   - En Leave-One-Out, cada instancia individual del conjunto de datos se utiliza una vez como conjunto de prueba, mientras que el resto de los datos se utilizan para el entrenamiento.
   - Este método es el más exhaustivo, ya que utiliza casi todos los datos para entrenar el modelo en cada iteración, obteniendo así una estimación muy precisa de su rendimiento.
   - Sin embargo, es computacionalmente costoso, especialmente para conjuntos de datos grandes, ya que requiere un número de iteraciones igual al número de instancias en el conjunto de datos.

Desempeño del Clasificador

El desempeño del clasificador se evalúa en términos de dos métricas clave:

1. **Precisión (Accuracy):**
   - La precisión es la proporción de predicciones correctas sobre el total de predicciones realizadas. Esta métrica es útil para entender qué tan efectivo es el clasificador en general.
   - La precisión se calcula como:

$$\text{Precisión} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

   - En el laboratorio, se observó que ambos clasificadores (Euclidiano y 1NN) logran alta precisión en datasets como Iris, donde las clases están bien definidas. Sin embargo, en datasets como Wine, donde las características de las clases pueden superponerse, la precisión disminuye.

2. **Matriz de Confusión:**
   - La matriz de confusión es una tabla que muestra el rendimiento del clasificador en cada clase, permitiendo identificar en cuáles clases el modelo tiende a confundirse.
   - Cada columna de la matriz representa las predicciones de una clase, mientras que cada fila representa las instancias de la clase real.
   - La matriz de confusión nos ayuda a entender los errores específicos del clasificador y es especialmente útil cuando se evalúan modelos en datasets con múltiples clases, como el dataset Digits.

Datasets Utilizados

1. **Dataset Iris:**
    - Este es un conjunto de datos muy conocido en el aprendizaje automático, que contiene 150 muestras de flores de iris, divididas en tres clases: Iris Setosa, Iris Versicolour, e Iris Virginica.
    - Cada muestra tiene cuatro características: longitud y ancho del sépalo, y longitud y ancho del pétalo.
    - Las clases en este dataset están bien separadas en el espacio de características, lo que facilita la clasificación y permite que ambos clasificadores (Euclidiano y 1NN) obtengan un alto rendimiento.
2. **Dataset Wine:**
    - Este dataset contiene datos de tres tipos diferentes de vinos cultivados en la región italiana de Piamonte. Cada vino está representado por 13 características, que incluyen componentes químicos como el ácido málico y la alcalinidad.
    - A diferencia del dataset Iris, las clases en Wine tienen una mayor superposición en el espacio de características, lo que hace más difícil para los clasificadores distinguir entre ellas.
    - Debido a esta complejidad, el rendimiento de los clasificadores disminuye en comparación con el dataset Iris.
3. **Dataset Digits:**
    - Este conjunto de datos contiene imágenes de dígitos escritos a mano, cada una representada como una cuadrícula de 8x8 píxeles (un total de 64 características por muestra).
    - El objetivo es clasificar cada imagen en uno de los 10 dígitos (0 a 9).
    - Este dataset es más complejo debido a la alta dimensionalidad y a la similitud entre ciertas clases (por ejemplo, 3 y 8). Sin embargo, ambos clasificadores logran buenos resultados gracias a la estructura interna del dataset y la adecuada separación de las clases.

Resultados Esperados

Para evaluar el rendimiento de los clasificadores, utilizamos tres métodos de validación:

- **Hold-Out (70/30)**: Separa el 70% de los datos para entrenamiento y el 30% para prueba.
- **10-Fold Cross-Validation**: Divide los datos en 10 partes y usa cada parte como prueba mientras entrena con las otras nueve.
- **Leave-One-Out**: Cada instancia es utilizada una vez como prueba y el resto como entrenamiento.

Los resultados de cada clasificador se muestran a continuación para cada dataset:

Clasificador Euclidiano

Hold Out 70/30

```
Resultados del Clasificador Euclidiano

Dataset: Iris
Método de Validación: holdout
Precisión: 0.9556
Matriz de Confusión:
                Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                    19                      0                     0
Real versicolor                 0                     11                     2
Real virginica                  0                      0                    13

Dataset: Wine
Método de Validación: holdout
Precisión: 0.7593
Matriz de Confusión:
              Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                  17                   0                   2
Real class_1                   0                  14                   7
Real class_2                   0                   4                  10

Dataset: Digits
Método de Validación: holdout
Precisión: 0.8889
Matriz de Confusión:
        Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0            52             0             0             0             0             1             0             0             0             0
Real 1             0            35             7             0             0             0             0             0             4             4
Real 2             0             1            43             1             0             0             0             0             2             0
Real 3             0             0             0            48             0             0             0             1             4             1
Real 4             0             3             0             0            55             0             0             2             0             0
Real 5             0             0             0             0             1            52             1             0             0            12
Real 6             1             0             0             0             0             0            52             0             0             0
Real 7             0             0             0             0             0             1             0            54             0             0
Real 8             0             3             0             0             0             1             0             0            38             1
Real 9             0             1             0             1             1             1             0             3             1            51
```

10-Fold Cross-Validation

```
Dataset: Iris
Método de Validación: kfold
Precisión: 0.9267
Matriz de Confusión:
                Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                    50                      0                     0
Real versicolor                 0                     45                     5
Real virginica                  0                      6                    44

Dataset: Wine
Método de Validación: kfold
Precisión: 0.7248
Matriz de Confusión:
              Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                  50                   0                   9
Real class_1                   3                  49                  19
Real class_2                   1                  17                  30

Dataset: Digits
Método de Validación: kfold
Precisión: 0.9032
Matriz de Confusión:
        Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0           177             0             0             0             1             0             0             0             0             0
Real 1             0           144            11             0             0             1             3             0             6            17
Real 2             1             5           158             4             0             0             0             2             5             2
Real 3             0             1             1           161             0             1             0             7             8             4
Real 4             0             5             0             0           168             0             0             5             3             0
Real 5             0             0             0             0             1           162             1             0             0            18
Real 6             1             4             0             0             0             0           175             0             1             0
Real 7             0             0             0             0             0             2             0           175             2             0
Real 8             0            14             1             1             0             4             1             2           142             9
Real 9             0             2             0             1             3             4             0             7             2           161
```

Leave-One-Out

```
Dataset: Iris
Método de Validación: leaveoneout
Precisión: 0.9200
Matriz de Confusión:
                Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                    50                      0                     0
Real versicolor                 0                     45                     5
Real virginica                  0                      7                    43

Dataset: Wine
Método de Validación: leaveoneout
Precisión: 0.7247
Matriz de Confusión:
              Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                  50                   0                   9
Real class_1                   3                  49                  19
Real class_2                   1                  17                  30

Dataset: Digits
Método de Validación: leaveoneout
Precisión: 0.9021
Matriz de Confusión:
        Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0           177             0             0             0             1             0             0             0             0             0
Real 1             0           143            10             1             0             1             3             0             7            17
Real 2             1             5           158             4             0             0             0             2             5             2
Real 3             0             1             1           161             0             1             0             7             8             4
Real 4             0             5             0             0           168             0             0             5             3             0
Real 5             0             0             0             0             1           161             1             0             0            19
Real 6             1             4             0             0             0             0           175             0             1             0
Real 7             0             0             0             0             0             2             0           175             2             0
Real 8             0            14             2             1             0             4             1             2           142             8
Real 9             0             3             0             1             3             4             0             6             2           161
```

Clasificador 1NN

Hold Out 70/30

```
Resultados del Clasificador 1NN

Dataset: Iris
Método de Validación: holdout
Precisión: 1.0000
Matriz de Confusión:
              Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                  19                      0                     0
Real versicolor               0                     13                     0
Real virginica                0                      0                    13

Dataset: Wine
Método de Validación: holdout
Precisión: 0.7963
Matriz de Confusión:
            Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                17                   0                   2
Real class_1                 3                  16                   2
Real class_2                 1                   3                  10

Dataset: Digits
Método de Validación: holdout
Precisión: 0.9833
Matriz de Confusión:
        Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0            53             0             0             0             0             0             0             0             0             0
Real 1             0            50             0             0             0             0             0             0             0             0
Real 2             0             0            47             0             0             0             0             0             0             0
Real 3             0             0             0            53             0             0             0             0             1             0
Real 4             0             1             0             0            59             0             0             0             0             0
Real 5             0             0             0             0             0            65             0             0             0             1
Real 6             0             0             0             0             0             0            53             0             0             0
Real 7             0             0             0             0             0             0             0            54             0             1
Real 8             0             1             0             0             0             0             0             0            41             1
Real 9             0             0             0             2             1             0             0             0             0            56
```

10-Fold Cross-Validation

```
Dataset: Iris
Método de Validación: kfold
Precisión: 0.9600
Matriz de Confusión:
              Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                  50                      0                     0
Real versicolor               0                     47                     3
Real virginica                0                      3                    47

Dataset: Wine
Método de Validación: kfold
Precisión: 0.7310
Matriz de Confusión:
            Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                52                   3                   4
Real class_1                 5                  52                  14
Real class_2                 3                  19                  26

Dataset: Digits
Método de Validación: kfold
Precisión: 0.9878
Matriz de Confusión:
       Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0           178             0             0             0             0             0             0             0             0             0
Real 1             0           182             0             0             0             0             0             0             0             0
Real 2             0             0           176             1             0             0             0             0             0             0
Real 3             0             0             0           183             0             0             0             0             0             0
Real 4             0             0             0             0           181             0             0             0             0             0
Real 5             0             0             0             0             0           179             1             0             0             2
Real 6             0             1             0             0             0             0           180             0             0             0
Real 7             0             0             0             0             0             0             0           178             0             1
Real 8             0             6             0             0             0             0             0             0           168             0
Real 9             0             1             0             4             1             2             0             0             2           170
```

Leave-One-Out

```
Dataset: Iris
Método de Validación: leaveoneout
Precisión: 0.9600
Matriz de Confusión:
                Predicción setosa  Predicción versicolor  Predicción virginica
Real setosa                    50                      0                     0
Real versicolor                 0                     47                     3
Real virginica                  0                      3                    47

Dataset: Wine
Método de Validación: leaveoneout
Precisión: 0.7697
Matriz de Confusión:
             Predicción class_0  Predicción class_1  Predicción class_2
Real class_0                  52                   3                   4
Real class_1                   5                  54                  12
Real class_2                   3                  14                  31

Dataset: Digits
Método de Validación: leaveoneout
Precisión: 0.9883
Matriz de Confusión:
        Predicción 0  Predicción 1  Predicción 2  Predicción 3  Predicción 4  Predicción 5  Predicción 6  Predicción 7  Predicción 8  Predicción 9
Real 0           178             0             0             0             0             0             0             0             0             0
Real 1             0           182             0             0             0             0             0             0             0             0
Real 2             0             0           176             1             0             0             0             0             0             0
Real 3             0             0             0           183             0             0             0             0             0             0
Real 4             0             0             0             0           181             0             0             0             0             0
Real 5             0             0             0             0             0           179             1             0             0             2
Real 6             0             1             0             0             0             0           180             0             0             0
Real 7             0             0             0             0             0             0             0           178             0             1
Real 8             0             5             0             0             0             0             0             0           169             0
Real 9             0             1             0             4             1             2             0             0             2           170
```

Conclusión

A lo largo de este laboratorio, hemos explorado el funcionamiento y desempeño de dos clasificadores basados en la proximidad: el **clasificador Euclidiano** y el **clasificador 1-Nearest Neighbor (1NN)**. Estos algoritmos fueron evaluados en tres conjuntos de datos de características distintas (Iris, Wine y Digits), y se utilizó una variedad de métodos de validación (Hold-Out, 10-Fold Cross-Validation y Leave-One-Out) para obtener una visión completa de su efectividad.

Los resultados muestran que ambos clasificadores son efectivos en escenarios donde las clases están bien separadas en el espacio de características, como en el dataset Iris. En este tipo de conjuntos de datos, los algoritmos lograron una alta precisión, demostrando su capacidad para distinguir correctamente entre instancias de diferentes clases. Sin embargo, en el dataset Wine, donde las clases tienen una mayor superposición, el rendimiento de ambos clasificadores disminuyó. Esto pone de manifiesto una limitación clave de los métodos basados en la distancia: su sensibilidad a la estructura interna de los datos y a la cercanía de las clases en el espacio de características.

El uso de los tres métodos de validación permitió obtener una evaluación detallada del desempeño de cada clasificador. La validación **Hold-Out** es rápida y proporciona resultados confiables en datasets grandes, aunque su dependencia en una única partición de los datos puede llevar a resultados menos representativos en datasets más pequeños. La validación mediante **10-Fold Cross-Validation** demostró ser un buen compromiso entre precisión y eficiencia computacional, ya que promedia los resultados obtenidos en múltiples particiones, lo cual reduce la varianza y mejora la estabilidad de los resultados. Finalmente, el método **Leave-One-Out** mostró ser el más preciso, ya que utiliza la mayor cantidad de datos posible en cada iteración. Sin embargo, su alto costo computacional limita su aplicabilidad en conjuntos de datos grandes como Digits.

En general, el clasificador Euclidiano y el clasificador 1NN son métodos simples y efectivos para problemas de clasificación cuando las clases están bien diferenciadas y el dataset es de tamaño moderado. Sin embargo, para aplicaciones más complejas, sería recomendable explorar clasificadores más avanzados y robustos ante ruido y alta dimensionalidad, como **SVM** o **Redes Neuronales**, que ofrecen una mayor flexibilidad para modelar relaciones complejas entre las clases.

En conclusión, este laboratorio ha permitido comprender los beneficios y limitaciones de los clasificadores basados en la distancia, así como la importancia de seleccionar un método de validación adecuado para cada contexto. Estos conocimientos son fundamentales en el análisis de datos y constituyen una base sólida para el estudio de algoritmos de clasificación más sofisticados. Este ejercicio resalta la relevancia de entender las características del conjunto de datos y el contexto de aplicación antes de seleccionar un modelo, ya que el rendimiento de los clasificadores depende en gran medida de estos factores.