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

Laboratorio 10: KNN y Bayes

Alumno: Hurtado Morales Emiliano - 2021630390

Maestro: Andrés García Floriano

Grupo: 6CV3

Fecha de entrega: 22/11/2024

Ciclo Escolar: 2025 – 1

Introducción

En el ámbito de la clasificación supervisada, los algoritmos de aprendizaje automático desempeñan un papel fundamental al permitir categorizar datos en clases predefinidas. En este laboratorio, exploramos dos algoritmos de clasificación populares: Naive Bayes y K-Nearest Neighbors (KNN). Cada uno de estos métodos tiene enfoques distintos y se basa en supuestos específicos. Naive Bayes, un clasificador probabilístico basado en el teorema de Bayes, es eficaz y rápido, especialmente cuando las características de los datos son independientes. KNN, por otro lado, es un método basado en instancias que clasifica una nueva instancia en función de sus vecinos más cercanos en el espacio de características.

Para evaluar estos algoritmos, se utilizaron tres datasets de diferentes características: Iris, Wine y Digits, que representan datos de flores, vinos y dígitos manuscritos, respectivamente. Estos datasets permiten probar los clasificadores en contextos variados, desde datos linealmente separables hasta problemas más complejos de clasificación visual. Además, se implementaron tres métodos de validación de rendimiento: Hold-Out (70/30), Validación Cruzada de 10 particiones (10-Fold Cross-Validation) y Leave-One-Out. Cada método de validación tiene características únicas en términos de estabilidad y generalización, proporcionando diferentes perspectivas sobre el rendimiento de los clasificadores.

El objetivo principal de este experimento es analizar el rendimiento de Naive Bayes y KNN en cada dataset, utilizando diferentes configuraciones y métodos de validación para obtener una comprensión completa de sus capacidades y limitaciones. Asimismo, se examinaron cómo varía el rendimiento de KNN en función del número de vecinos (K) y la influencia que tienen los métodos de validación en los resultados de precisión y en las matrices de confusión.

Explicación de los Algoritmos

Naive Bayes:

- **Principio Básico**: Naive Bayes es un clasificador probabilístico basado en el teorema de Bayes. Este algoritmo asume que las características son independientes entre sí, lo cual es una suposición fuerte (de ahí el término "Naive").
- **Función de Decisión**: Calcula la probabilidad posterior de cada clase dada una instancia de entrada y clasifica esa instancia en la clase con la mayor probabilidad posterior.
- **Cálculo de Probabilidades**: En este caso, se usa la versión Gaussiana de Naive Bayes (GaussianNB), que es adecuada para datos continuos. Esta variante asume que las características siguen una distribución normal.
- **Ventajas y Desventajas**: Su principal ventaja es su eficiencia computacional y la simplicidad en la implementación, mientras que su principal desventaja es la fuerte suposición de independencia entre características, que rara vez se cumple en la práctica.

```
# Implementación del clasificador Naive Bayes
def clasificador_naive_bayes(datos_entrenamiento, etiquetas_entrenamiento):
    """
    Entrena un clasificador Naive Bayes en los datos de entrenamiento.

    Args:
        datos_entrenamiento (array): Matriz de características para el
entrenamiento.
        etiquetas_entrenamiento (array): Etiquetas de clase correspondientes a
los datos de entrenamiento.

    Returns:
        GaussianNB: Modelo de Naive Bayes entrenado.
    """
    modelo = GaussianNB()
    modelo.fit(datos_entrenamiento, etiquetas_entrenamiento)
    return modelo
```

K-Nearest Neighbors (KNN):

- **Principio Básico**: KNN es un clasificador basado en instancias que asigna una clase a una instancia de prueba en función de las clases de sus "K" vecinos más cercanos. La proximidad se mide generalmente mediante una distancia (e.g., Euclidiana).
- **Proceso de Clasificación**: El algoritmo calcula la distancia entre la instancia de prueba y todos los puntos de entrenamiento, selecciona los "K" vecinos más cercanos y asigna la clase predominante entre esos vecinos.
- **Elección de K**: El valor de K es crítico en el rendimiento de KNN. Un valor bajo de K puede llevar a un modelo ruidoso y sensible, mientras que un valor alto puede hacer que el modelo sea demasiado general. En este experimento, probamos con valores de K=3, K=5 y K=7 para observar cómo afecta la precisión.
- **Ventajas y Desventajas**: KNN es fácil de entender e implementar, pero es computacionalmente costoso para grandes volúmenes de datos, ya que requiere calcular distancias para todas las instancias de entrenamiento. También es sensible a la elección de K y la escala de los datos.

```
# Implementación del clasificador K-Nearest Neighbors (KNN)
def clasificador_knn(datos_entrenamiento, etiquetas_entrenamiento, k=3):
    """
    Entrena un clasificador KNN con un valor de K especificado en los datos de
entrenamiento.

    Args:
        datos_entrenamiento (array): Matriz de características para el
entrenamiento.
        etiquetas_entrenamiento (array): Etiquetas de clase correspondientes a
los datos de entrenamiento.
        k (int): Número de vecinos a considerar en el modelo KNN.

    Returns:
        KNeighborsClassifier: Modelo KNN entrenado.
    """
    modelo = KNeighborsClassifier(n_neighbors=k)
    modelo.fit(datos_entrenamiento, etiquetas_entrenamiento)
    return modelo
```

Resultados Esperados

Al evaluar el rendimiento de Naive Bayes y KNN en cada dataset y método de validación, se espera obtener los siguientes resultados:

- **Naive Bayes**: Dado que Naive Bayes es menos afectado por el tamaño del conjunto de datos y es rápido de entrenar, esperamos que funcione bien en los datasets Iris y Wine, que tienen características relativamente simples y distribuciones gaussianas aproximadas. Sin embargo, podría tener un rendimiento limitado en el dataset Digits, debido a la mayor complejidad y menor independencia entre características en datos de imágenes.
- **KNN**: El rendimiento de KNN depende de la elección del valor de K. Esperamos que con valores de K más bajos (como K=3), KNN sea más sensible a las variaciones y el ruido en los datos, pero con K=5 o K=7, es probable que los resultados sean más estables. En el dataset Digits, KNN debería tener un buen rendimiento, ya que la similitud entre las representaciones de dígitos vecinos podría ayudar a la clasificación.
- **Impacto de los Métodos de Validación**:
    - **Hold-Out**: Al depender de una sola partición, podría dar resultados menos estables.
    - **Validación Cruzada**: Debería proporcionar resultados más fiables debido al uso de múltiples particiones.
    - **Leave-One-Out**: Debería mostrar una precisión cercana a la media de los otros métodos, aunque con menor variabilidad, especialmente en conjuntos de datos pequeños.

Dataset Iris

```
Clasificador: Naive Bayes

Matriz de Confusión - Naive Bayes (Hold-Out) en Iris
Precisión: 0.98
            setosa  versicolor  virginica
setosa          19           0          0
versicolor       0          12          1
virginica        0           0         13

Matriz de Confusión - Naive Bayes (Validación Cruzada) en Iris
Precisión: 0.95
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          47          3
virginica        0           4         46

Matriz de Confusión - Naive Bayes (Leave-One-Out) en Iris
Precisión: 0.95
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          47          3
virginica        0           4         46

Clasificador: KNN con K=3

Matriz de Confusión - KNN (K=3) (Hold-Out) en Iris
Precisión: 1.00
            setosa  versicolor  virginica
setosa          19           0          0
versicolor       0          13          0
virginica        0           0         13

Matriz de Confusión - KNN (K=3) (Validación Cruzada) en Iris
Precisión: 0.97
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          47          3
virginica        0           2         48
```

```
Matriz de Confusión - KNN (K=3) (Leave-One-Out) en Iris
Precisión: 0.96
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          47          3
virginica        0           3         47

Clasificador: KNN con K=5

Matriz de Confusión - KNN (K=5) (Hold-Out) en Iris
Precisión: 1.00
            setosa  versicolor  virginica
setosa          19           0          0
versicolor       0          13          0
virginica        0           0         13

Matriz de Confusión - KNN (K=5) (Validación Cruzada) en Iris
Precisión: 0.97
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          46          4
virginica        0           1         49

Matriz de Confusión - KNN (K=5) (Leave-One-Out) en Iris
Precisión: 0.97
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          47          3
virginica        0           2         48

Clasificador: KNN con K=7

Matriz de Confusión - KNN (K=7) (Hold-Out) en Iris
Precisión: 1.00
            setosa  versicolor  virginica
setosa          19           0          0
versicolor       0          13          0
virginica        0           0         13
```

```
Matriz de Confusión - KNN (K=7) (Validación Cruzada) en Iris
Precisión: 0.97
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          46          4
virginica        0           1         49

Matriz de Confusión - KNN (K=7) (Leave-One-Out) en Iris
Precisión: 0.97
            setosa  versicolor  virginica
setosa          50           0          0
versicolor       0          46          4
virginica        0           1         49
```

Dataset Wine

```
Resultados para el dataset: Wine

Clasificador: Naive Bayes

Matriz de Confusión - Naive Bayes (Hold-Out) en Wine
Precisión: 1.00
         class_0  class_1  class_2
class_0       19        0        0
class_1        0       21        0
class_2        0        0       14

Matriz de Confusión - Naive Bayes (Validación Cruzada) en Wine
Precisión: 0.98
         class_0  class_1  class_2
class_0       57        2        0
class_1        0       69        2
class_2        0        0       48

Matriz de Confusión - Naive Bayes (Leave-One-Out) en Wine
Precisión: 0.98
         class_0  class_1  class_2
class_0       57        2        0
class_1        0       69        2
class_2        0        0       48

Clasificador: KNN con K=3

Matriz de Confusión - KNN (K=3) (Hold-Out) en Wine
Precisión: 0.74
         class_0  class_1  class_2
class_0       17        0        2
class_1        1       15        5
class_2        1        5        8
```

```
Matriz de Confusión - KNN (K=3) (Validación Cruzada) en Wine
Precisión: 0.72
         class_0  class_1  class_2
class_0       51        3        5
class_1        7       48       16
class_2        5       14       29

Matriz de Confusión - KNN (K=3) (Leave-One-Out) en Wine
Precisión: 0.72
         class_0  class_1  class_2
class_0       51        3        5
class_1        7       49       15
class_2        7       12       29

Clasificador: KNN con K=5

Matriz de Confusión - KNN (K=5) (Hold-Out) en Wine
Precisión: 0.74
         class_0  class_1  class_2
class_0       17        0        2
class_1        1       15        5
class_2        1        5        8

Matriz de Confusión - KNN (K=5) (Validación Cruzada) en Wine
Precisión: 0.67
         class_0  class_1  class_2
class_0       52        1        6
class_1        6       47       18
class_2        7       20       21

Matriz de Confusión - KNN (K=5) (Leave-One-Out) en Wine
Precisión: 0.70
         class_0  class_1  class_2
class_0       52        1        6
class_1        6       49       16
class_2        6       19       23
```

```
Clasificador: KNN con K=7

Matriz de Confusión - KNN (K=7) (Hold-Out) en Wine
Precisión: 0.76
         class_0  class_1  class_2
class_0       18        0        1
class_1        1       14        6
class_2        1        4        9

Matriz de Confusión - KNN (K=7) (Validación Cruzada) en Wine
Precisión: 0.67
         class_0  class_1  class_2
class_0       52        0        7
class_1        6       44       21
class_2        8       17       23

Matriz de Confusión - KNN (K=7) (Leave-One-Out) en Wine
Precisión: 0.66
         class_0  class_1  class_2
class_0       54        0        5
class_1        6       45       20
class_2        8       21       19
```

Dataset Digits

```
Resultados para el dataset: Digits

Clasificador: Naive Bayes

Matriz de Confusión - Naive Bayes (Hold-Out) en Digits
Precisión: 0.85
    0   1   2   3   4   5   6   7   8   9
0  52   0   0   0   0   0   0   1   0   0
1   0  37   2   0   0   0   0   2   6   3
2   0   3  31   0   0   0   1   0  12   0
3   0   0   2  41   0   0   1   0   8   2
4   0   0   0   0  51   0   2   7   0   0
5   0   0   0   1   0  62   1   2   0   0
6   0   0   0   0   1   1  51   0   0   0
7   0   0   0   0   0   1   0  54   0   0
8   0   2   0   0   0   0   0   2  39   0
9   0   1   1   1   0   2   1   7   4  42

Matriz de Confusión - Naive Bayes (Validación Cruzada) en Digits
Precisión: 0.81
     0    1    2    3    4    5    6    7    8    9
0  174    0    0    0    2    0    0    1    0    1
1    0  141    3    0    1    0    6    5   17    9
2    0   10  112    0    1    2    1    0   51    0
3    0    2    4  131    0    8    0    8   25    5
4    1    2    1    0  147    1    2   25    2    0
5    0    2    0    3    1  160    1    9    3    3
6    0    1    1    0    1    3  175    0    0    0
7    0    0    1    0    1    1    0  174    1    1
8    0   25    2    1    0    3    0   11  130    2
9    1   11    0    7    2    4    1   17   23  114
```

```
Matriz de Confusión - Naive Bayes (Leave-One-Out) en Digits
Precisión: 0.84
     0    1    2    3    4    5    6    7    8    9
0  174    0    0    0    2    1    0    1    0    0
1    0  148    1    0    0    0    4    5   16    8
2    0   13  110    1    1    1    1    0   50    0
3    0    2    3  139    0    7    0    7   22    3
4    1    2    1    0  149    1    2   22    3    0
5    0    0    0    3    0  168    1    6    3    1
6    0    1    1    0    1    2  176    0    0    0
7    0    0    0    0    1    1    0  176    0    1
8    0    8    1    1    0    3    0   12  149    0
9    1    8    0    5    2    3    1   18   20  122

Clasificador: KNN con K=3

Matriz de Confusión - KNN (K=3) (Hold-Out) en Digits
Precisión: 0.99
    0   1   2   3   4   5   6   7   8   9
0  53   0   0   0   0   0   0   0   0   0
1   0  50   0   0   0   0   0   0   0   0
2   0   0  47   0   0   0   0   0   0   0
3   0   0   0  54   0   0   0   0   0   0
4   0   0   0   0  60   0   0   0   0   0
5   0   0   0   0   0  66   0   0   0   0
6   0   0   0   0   0   0  53   0   0   0
7   0   0   0   0   0   0   0  54   0   1
8   0   1   0   0   0   0   0   0  42   0
9   0   0   0   1   1   1   0   0   1  55
```

```
Matriz de Confusión - KNN (K=3) (Validación Cruzada) en Digits
Precisión: 0.98
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  180    0    0    0    1    1    0    0    0
2    0    4  171    1    0    0    0    0    1    0
3    0    0    1  176    0    1    0    2    2    1
4    0    1    0    0  178    0    1    0    0    1
5    0    0    0    0    0  178    1    0    0    3
6    1    1    0    0    0    0  179    0    0    0
7    0    0    0    0    0    0    0  178    0    1
8    0    6    0    2    0    0    0    0  166    0
9    0    1    0    4    1    1    0    0    2  171

Matriz de Confusión - KNN (K=3) (Leave-One-Out) en Digits
Precisión: 0.99
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  182    0    0    0    0    0    0    0    0
2    0    0  177    0    0    0    0    0    0    0
3    0    0    0  181    0    0    0    2    0    0
4    0    0    0    0  181    0    0    0    0    0
5    0    0    0    0    0  178    1    0    0    3
6    0    0    0    0    0    0  181    0    0    0
7    0    0    0    0    0    0    0  178    0    1
8    0    4    0    1    0    0    0    0  169    0
9    0    1    0    3    1    1    0    0    2  172
```

```
Clasificador: KNN con K=5

Matriz de Confusión - KNN (K=5) (Hold-Out) en Digits
Precisión: 0.99
    0   1   2   3   4   5   6   7   8   9
0  53   0   0   0   0   0   0   0   0   0
1   0  50   0   0   0   0   0   0   0   0
2   0   0  47   0   0   0   0   0   0   0
3   0   0   0  54   0   0   0   0   0   0
4   0   0   0   0  60   0   0   0   0   0
5   0   0   0   0   0  65   0   0   0   1
6   0   0   0   0   0   0  53   0   0   0
7   0   0   0   0   0   0   0  55   0   0
8   0   0   0   0   0   0   0   0  43   0
9   0   0   0   1   1   1   0   0   0  56

Matriz de Confusión - KNN (K=5) (Validación Cruzada) en Digits
Precisión: 0.97
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  181    0    0    0    0    1    0    0    0
2    0    3  170    0    0    0    0    1    3    0
3    0    0    1  175    0    1    0    2    2    2
4    0    1    0    0  178    0    0    2    0    0
5    0    0    0    0    0  177    1    0    0    4
6    0    1    0    0    0    1  179    0    0    0
7    0    0    0    0    0    0    0  176    0    3
8    0   10    1    1    0    0    0    1  161    0
9    0    2    0    4    1    1    0    1    1  170
```

```
Matriz de Confusión - KNN (K=5) (Leave-One-Out) en Digits
Precisión: 0.99
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  182    0    0    0    0    0    0    0    0
2    0    0  176    1    0    0    0    0    0    0
3    0    0    0  181    0    0    0    2    0    0
4    0    0    0    0  180    0    0    1    0    0
5    0    0    0    0    0  180    1    0    0    1
6    0    0    0    0    0    0  180    0    1    0
7    0    0    0    0    0    0    0  178    0    1
8    0    4    0    1    0    0    0    0  169    0
9    0    1    0    3    1    2    0    0    2  171

Clasificador: KNN con K=7

Matriz de Confusión - KNN (K=7) (Hold-Out) en Digits
Precisión: 0.99
    0   1   2   3   4   5   6   7   8   9
0  53   0   0   0   0   0   0   0   0   0
1   0  50   0   0   0   0   0   0   0   0
2   0   0  47   0   0   0   0   0   0   0
3   0   0   0  54   0   0   0   0   0   0
4   0   0   0   0  60   0   0   0   0   0
5   0   0   0   0   0  64   1   0   0   1
6   0   0   0   0   0   0  53   0   0   0
7   0   0   0   0   0   0   0  55   0   0
8   0   0   0   0   0   0   0   0  43   0
9   0   0   0   1   1   1   0   0   0  56
```

```
Matriz de Confusión - KNN (K=7) (Validación Cruzada) en Digits
Precisión: 0.97
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  180    1    0    0    0    1    0    0    0
2    0    3  168    0    0    0    0    1    5    0
3    0    0    1  173    0    1    0    2    4    2
4    0    1    0    0  177    0    0    2    0    1
5    0    0    0    0    0  177    1    0    0    4
6    0    2    0    0    0    1  178    0    0    0
7    0    0    0    0    0    0    0  178    0    1
8    0    9    1    2    0    0    0    1  160    1
9    0    2    0    2    1    2    0    2    1  170

Matriz de Confusión - KNN (K=7) (Leave-One-Out) en Digits
Precisión: 0.99
     0    1    2    3    4    5    6    7    8    9
0  178    0    0    0    0    0    0    0    0    0
1    0  182    0    0    0    0    0    0    0    0
2    0    0  176    0    0    0    0    1    0    0
3    0    0    0  179    0    1    0    2    1    0
4    0    0    0    0  178    0    0    2    1    0
5    0    0    0    0    0  179    1    0    0    2
6    0    0    0    0    0    0  180    0    1    0
7    0    0    0    0    0    0    0  179    0    0
8    0    5    0    2    0    0    0    0  167    0
9    0    1    0    2    1    2    0    0    1  173
```

Conclusión

A través de este experimento, se ha obtenido una visión integral del rendimiento de los clasificadores Naive Bayes y KNN en diferentes contextos de datos y métodos de validación. Los resultados demuestran que Naive Bayes es particularmente eficaz en datasets con características simples y donde las suposiciones de independencia entre características no afectan significativamente su desempeño, como en los datasets Iris y Wine. No obstante, en el dataset Digits, donde las relaciones entre características son más complejas, Naive Bayes muestra una menor precisión, evidenciando su limitación en problemas de clasificación visual compleja.

Por otro lado, el rendimiento de KNN depende en gran medida de la elección del valor de K y del método de validación. Para valores de K bajos, el modelo puede ser más sensible al ruido, mientras que valores de K altos tienden a mejorar la estabilidad del clasificador. En el dataset Digits, KNN mostró un rendimiento notablemente superior a Naive Bayes, debido a la naturaleza del problema y la capacidad de KNN para aprovechar las relaciones de proximidad en datos visuales. Además, la validación cruzada y el método Leave-One-Out ofrecen una evaluación más robusta y precisa en comparación con Hold-Out, ya que reducen la varianza de los resultados y maximizan el uso de los datos para el entrenamiento y la evaluación.

En conclusión, cada clasificador y método de validación tiene ventajas y limitaciones específicas, y su efectividad depende del tipo de datos y del contexto de aplicación. Mientras que Naive Bayes es adecuado para problemas donde la independencia entre características es asumible, KNN es más versátil en problemas donde la similitud entre instancias juega un papel importante. Este experimento subraya la importancia de seleccionar cuidadosamente tanto el modelo como el método de validación en función del problema, para obtener un rendimiento óptimo y resultados confiables.