Instituto Politécnico Nacional

Escuela Superior de Cómputo

Profesor:

Andres Garcia Floriano

Alumno:

Hernández Jiménez Erick Yael

Patiño Flores Samuel

Robert Garayzar Arturo

5BV1

Practica :
Clasificador Bayesiano Ingenuo

Índice

1. Introducción

1.1. Clasificador Bayesiano

El **clasificador bayesiano** es un tipo de modelo de aprendizaje automático basado en el teorema de Bayes, un principio fundamental de la teoría de probabilidad que describe la probabilidad de un evento basado en el conocimiento previo de condiciones relacionadas con ese evento. En términos simples, el clasificador bayesiano determina la probabilidad de que una instancia pertenezca a una clase específica dado un conjunto de características observadas. Esta probabilidad se calcula combinando las probabilidades de las características con las probabilidades a priori de las clases.

Matemáticamente, el teorema de Bayes se expresa como:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

donde: - $P(C|X)$ es la probabilidad posterior de la clase C dado los datos X. - $P(X|C)$ es la probabilidad de observar los datos X dado que pertenecen a la clase C. - $P(C)$ es la probabilidad a priori de la clase C. - $P(X)$ es la probabilidad de los datos X.

1.2. Clasificador Bayesiano Ingenuo

El **clasificador bayesiano ingenuo** (Naive Bayes) es una variante simplificada y popular del clasificador bayesiano que asume que todas las características son independientes entre sí, dado el valor de la clase. Aunque esta suposición de independencia rara vez se cumple en la realidad, el clasificador bayesiano ingenuo ha demostrado ser altamente eficaz en una amplia gama de aplicaciones, incluyendo la clasificación de texto, el filtrado de spam, el análisis de sentimientos y el diagnóstico médico.

El adjetivo 'ingenuo' se refiere a la fuerte hipótesis de independencia que hace el modelo. A pesar de esta limitación, Naive Bayes funciona bien debido a su simplicidad, rapidez y eficacia en contextos en los que muchas otras técnicas se complicarían o resultarían menos efectivas.

El proceso de clasificación con Naive Bayes implica calcular la probabilidad de que una instancia dada pertenezca a una clase específica utilizando las probabilidades individuales de sus características. Se selecciona la clase con la mayor probabilidad. La eficiencia de Naive Bayes y su capacidad para manejar tanto características numéricas como categóricas lo convierten en una opción atractiva para problemas de clasificación supervisada.

En particular, los datasets usados ya presentan las etiquetas mínimas necesarias para realizar predicciones, por lo que nos apegamos a usar la misma cantidad de características 'k' que contienen. Los datasets son 'iris', 'wine' y 'mushroom' o 'agaricus-lepiota'

1.3. Métodos de división de datos

En esta práctica se usaron 3 métodos de división de datos para entrenamiento y prueba:

1.3.1. Estratificación

La **estratificación** es una técnica utilizada para mantener la proporción de las clases en los conjuntos de entrenamiento y prueba durante la partición de los datos. Esto es especialmente útil cuando las clases están desbalanceadas, es decir, cuando algunas categorías tienen muchas más observaciones que otras. La estratificación garantiza que cada subconjunto de datos mantenga la misma distribución de clases que el conjunto original, proporcionando una evaluación más representativa del rendimiento del modelo.

1.3.2. Validación Leave-One-Out (LOO)

La **validación Leave-One-Out (LOO)** es un caso particular de validación cruzada en el que el número de subconjuntos es igual al número de observaciones en el conjunto de datos. En cada iteración, se utiliza una única observación como conjunto de prueba y el resto de las observaciones como conjunto de entrenamiento. Esto se repite tantas veces como observaciones haya, lo que da lugar a evaluaciones altamente exhaustivas. Aunque ofrece una estimación imparcial del error de generalización, LOO puede ser computacionalmente costoso para conjuntos de datos grandes debido a la gran cantidad de particiones.

1.3.3. n-fold Cross-Validation

La **validación cruzada de n-fold** implica dividir el conjunto de datos en n subconjuntos (o 'folds') de aproximadamente el mismo tamaño. En cada iteración, uno de estos subconjuntos se utiliza como conjunto de prueba y el resto como conjunto de entrenamiento. Este proceso se repite n veces, con cada subconjunto utilizado como conjunto de prueba una vez. El rendimiento se evalúa promediando las métricas obtenidas en cada iteración. Este método proporciona un equilibrio entre la cantidad de datos de entrenamiento disponibles y el tiempo computacional, y su versión más común es la *10-fold cross-validation*.

2. Desarrollo

El clasificador bayesiano, como se observa en los resultados obtenidos, muestra un alto nivel de precisión en los conjuntos de datos Iris, Wine y Mushroom, tanto con la validación mediante particionado estratificado como con la técnica Leave-One-Out.

- **Precisión general**: En el caso del particionado estratificado, la precisión fue consistentemente alta: alrededor de $0{,}911$ para Iris, $0{,}981$ para Wine y $0{,}998$ para Mushroom. Por otro lado, con Leave-One-Out, la precisión promedio alcanzó $0{,}953$ para Iris, $0{,}966$ para Wine y $0{,}997$ para Mushroom, lo que confirma la robustez del modelo bayesiano en diversos contextos.
- **Matriz de confusión con particionado estratificado**: Esta matriz muestra los aciertos y errores de clasificación. Por ejemplo, para el conjunto de datos Iris, los valores indican la cantidad de elementos correctamente clasificados y los errores entre las clases. Las matrices confusión para los conjuntos de datos Wine y Mushroom presentan un muy bajo número de errores de clasificación, evidenciando la alta capacidad predictiva.
- **Matriz de confusión con Leave-One-Out**: La matriz de confusión acumulada muestra cómo el clasificador mantiene un buen rendimiento al evaluar cada muestra individualmente. Los valores indican que las predicciones erróneas son mínimas en comparación con las predicciones correctas, confirmando su eficacia.

3. Conclusiones

Hernández Jiménez Erick Yael:

Una vez realizado el preprocesamiento del set de datos, se pueden destacar la siguiente conclusión: se debe analizar el efecto de cada herramienta que se usará para balancear o manejar los valores nulos ya que un manejo deliberado de estas conllevara errores en la interpretación del set resultante y que puede afectar completamente al análisis posterior o procesamiento por machine learning

Patiño Flores Samuel:

En esta práctica, se abordaron dos problemas comunes en el análisis de datos: los datos desbalanceados y los datos faltantes. Para tratar el desbalanceo en las clases, se aplicó la técnica SMOTE, que permitió generar ejemplos de la clase minoritaria, evitando el sesgo hacia la clase mayoritaria. En cuanto a los datos faltantes, se utilizó imputación, lo que evitó la pérdida de información valiosa al reemplazar los valores ausentes con estimaciones basadas en los datos disponibles.

Robert Garayzar Arturo:

La aplicación de técnicas para reducir el desbalance de clases y eliminar los valores perdidos permitió mejorar la calidad del conjunto de datos utilizado. Estas acciones resultaron en un dataset más equilibrado y libre de inconsistencias, lo cual es esencial para obtener resultados más precisos y confiables en modelos de análisis y predicción. Al reducir el sesgo que podría generarse por el desbalance de clases y asegurar que los datos estén completos, se logró una base más sólida para futuros análisis, lo que contribuirá a generar modelos más robustos y generalizables. Este proceso destacó la importancia de la preparación de datos como un paso fundamental en cualquier proyecto de ciencia de datos.