Informática y Telecomunicaciones. Ingeniería en Informática.

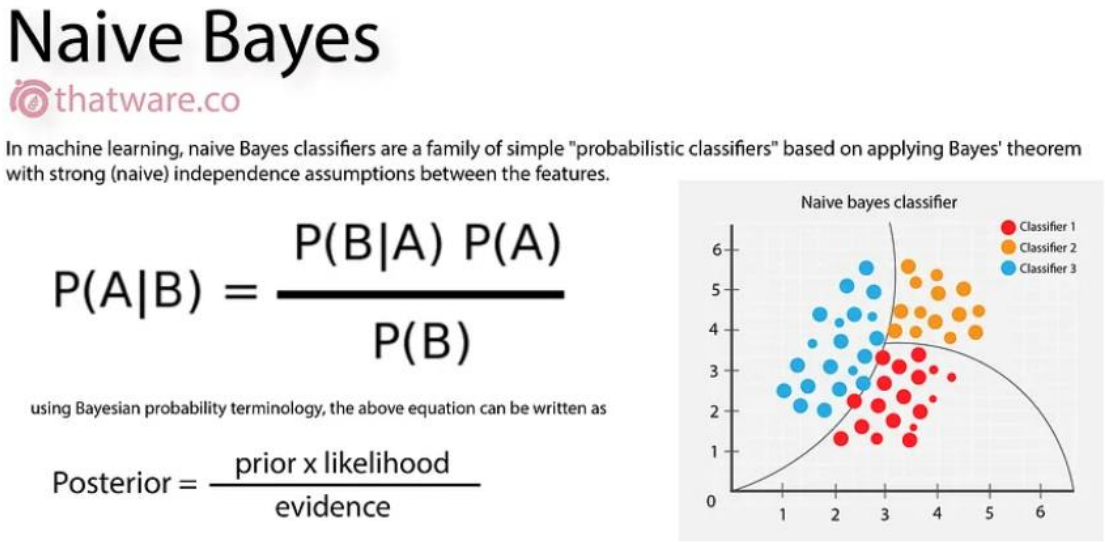
Inacap

Algoritmo Naive Bayes en dataset de vinos.

NOMBRE: MATIAS FRANCISCO CAVIERES BELMAR
CARRERA: INGENIERIA INFORMATICA
ASIGNATURA: APLICACIONES DE INTELIGENCIA ARTIFICIAL
PROFESOR: ERNESTO EDUARDO VIVANCO TAPIA
FECHA: 23/09/2024

1 Introducción

El algoritmo naive bayes es un algoritmo de aprendizaje automático que se utiliza para la clasificación de diferentes atributos en base a la clase. El algoritmo ha demostrado ser altamente eficaz en diversas aplicaciones, como el filtrado de spam y análisis de sentimientos entre otros.



Naive Bayes es fácil de implementar y muy rápido para entrenar, siendo una gran solución al momento de querer entrenar un modelo de IA. Tiene una gran eficiencia para manejar grandes volúmenes de datos mediante a sus fundamentos probabilísticos, tanto en términos de almacenamiento como en tiempo de procesamiento.

2 Desarrollo

2.1 Sobre el dataset utilizado

El dataset *Wine Recognition*, disponible en la biblioteca *scikit-learn*, es un conjunto de datos clásico utilizado en la clasificación. Originalmente lo recolectó el Instituto de la Universidad de Forense de Italia. Contiene información sobre muestras de vino provenientes de tres variedades cultivadas en la región italiana de la Toscana.

El conjunto de datos tiene las siguientes características clave:

- Características: 13 atributos químicos de los vinos (por ejemplo, alcohol, ácido málico, magnesio, entre otros).
- Número de muestras: 178.
- Etiquetas de clase: 3 clases correspondientes a las tres variedades de vino.
- Objetivo: Clasificar los vinos en una de las tres clases basándose en sus propiedades químicas.

2.2 Herramientas y generación del código

Para el desarrollo de la actividad, se utiliza Anaconda para poder ejecutar un entorno de jupyter notebook, de instalan librerias como numpy para el manejo numérico, matplotlib para la visualización gráfica y sklearn para importar el dataset, generacion de la matriz de confusión y el modelo naive bayes.

2.3 Codigo utilizado en el analisis.

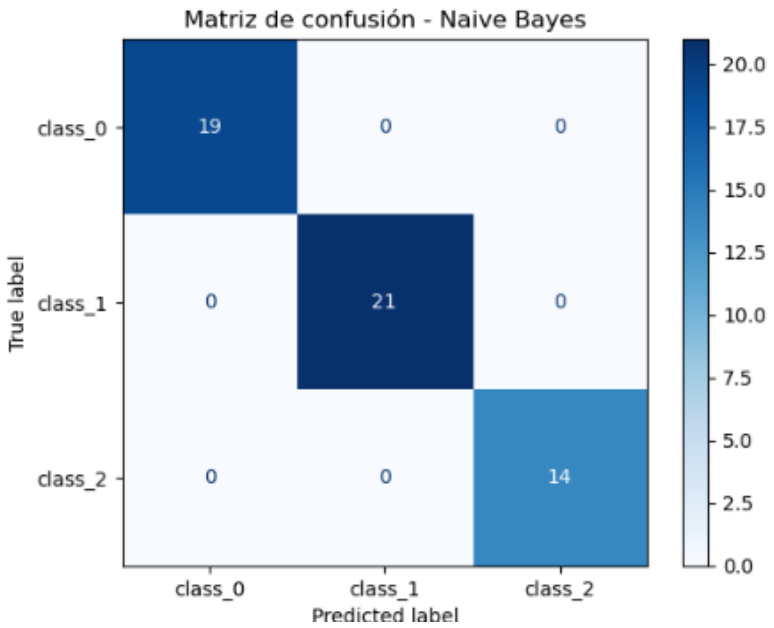
```
# Importar las bibliotecas necesarias
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix,
ConfusionMatrixDisplay

# Cargar el dataset Wine
wine = load_wine()
X = wine.data  # Características
y = wine.target  # Etiquetas de clase

# Dividir el conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Inicializar el clasificador Naive Bayes
nb_classifier = GaussianNB()

# Entrenar el modelo
nb_classifier.fit(X_train, y_train)

# Hacer predicciones sobre el conjunto de prueba
y_pred = nb_classifier.predict(X_test)

# Evaluar el rendimiento del modelo
accuracy = accuracy_score(y_test, y_pred)
print(f'Exactitud del modelo: {accuracy * 100:.2f}%')

# Matriz de confusión
conf_matrix = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,
display_labels=wine.target_names)

# Graficar la matriz de confusión
disp.plot(cmap=plt.cm.Blues)
plt.title('Matriz de confusión - Naive Bayes')
plt.show()
```

3 Resultados

Al ejecutar el código anterior en el entorno de jupyter notebook, se puede visualizar la siguiente matriz de confusion:

Exactitud del modelo: 100.00%



El modelo presenta una exactitud del 100.00%, significa que clasifica de manera correcta todas las muestras del conjunto de prueba.

La **matriz de confusión** es fundamental para visualizar el rendimiento de clasificación:

- Las filas representan las clases verdaderas.
- Las columnas representan las predicciones del modelo.

Un análisis clave de la matriz de confusión es:

- Cuando los valores en la diagonal principal son altos, significa que el modelo clasifica correctamente la mayoría de las muestras.
- Si hay valores significativos fuera de la diagonal, el modelo comete más errores en esas clases particulares.

4 Conclusiones

Los resultados obtenidos representan una situación ideal, pero es importante tener en cuenta que al trabajar con data sets más complejos o con datos reales se puede ver afectada la exactitud. Sin embargo, el data set de vinos presenta una exactitud perfecta al clasificar todos los datos en las líneas diagonales, tiene un rendimiento excelente y las características estan bien separadas entre las clases, las clases presentan una distribución equilibrada y el modelo no presenta un sesgo hacia una clase en particular, lo que evidencia que la matriz de confusion es perfecta.