Práctica 2

Inteligencia de Negocio

Amin Kasrou Aouam

UNIVERSIDAD
DE GRANADA

2020-12-13

Índice

Práctica 2

Apartado 1

Introducción

En este apartado, visualizaremos los resultados de la práctica anterior y los interpretaremos.

Procesamiento de datos

Mantenemos el mismo preprocesamiento de datos que en la práctica anterior, lo cual supone que para haremos una gráfica para cada tipo de preprocesamiento (eliminación de valores nulos e imputación con la media).

Ejecución

Para obtener las gráficas ejecutamos el siguiente comando, desde la raíz del proyecto:

```
python src/P1/processing.py drop
```

En el caso de que queramos los resultados al aplicar la imputación de la media, ejecutamos el siguiente comando:

```
python src/P1/processing.py fill
```

El conjunto de las gráficas se encontrará en el directorio **docs/assets**.

Gráfica de curva ROC

Una curva ROC nos permite medir el rendimiento de un clasificador según el umbral de discriminación en un problema de clasificación, i.e. nos permite saber como de bueno es nuestro modelo distinguiendo las diferentes clases.

Procedemos a mostrar cada uno de los modelos con un color distinto, además del *AUC*, dado que este parámetro nos permite asignar un valor numérico al rendimiento de cada modelo.

1. Eliminación de valores

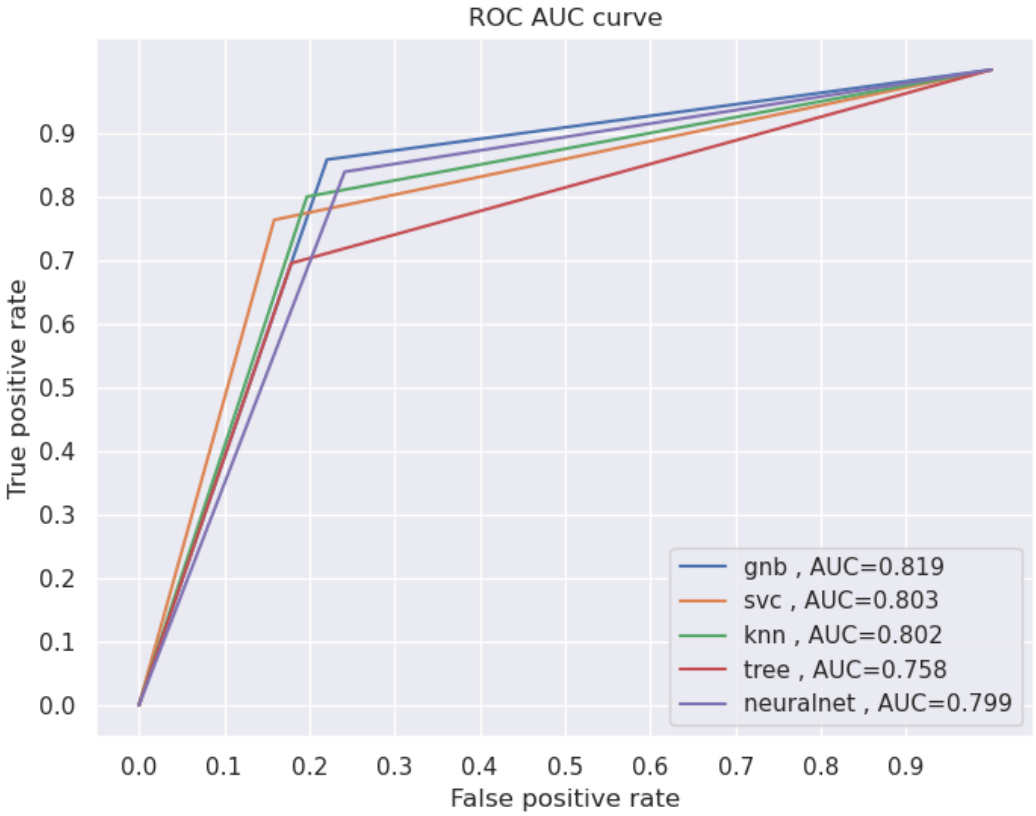

Figura 1: Curva ROC AUC con eliminación de valores

Observamos que el algoritmo *Naive Bayes* es el que obtiene una mejor puntuación, aunque no hay una gran diferencia de valores en el *AUC* entre los demás modelos.

2. Imputación de valores

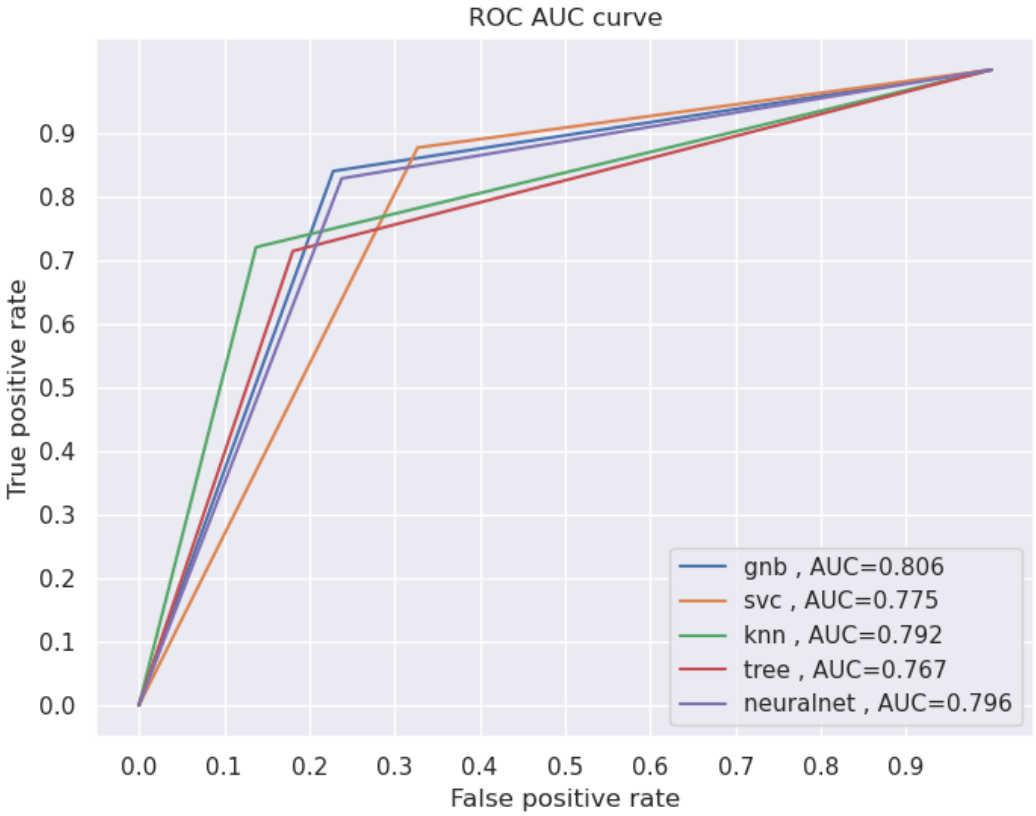

Figura 2: Curva ROC AUC con imputación de valores

Observamos que el algoritmo *Naive Bayes* sigue obteniendo la mejor puntuación, aunque vemos que las curvas han sido alteradas debido al proceso de imputación.

En la práctica anterior llegamos a la conclusión de que no era un buen método de preprocesamiento en nuestro caso particular.

Matriz de confusión

Una matriz de confusión nos permite visualizar el rendimiento de un algoritmo, al incluir el número de:

- Verdaderos positivos
- Falsos positivos
- Verdaderos negativos
- Falsos negativos

Procedemos a generar un *heatmap* por cada algoritmo, para comparar su rendimiento.

1. Eliminación de valores

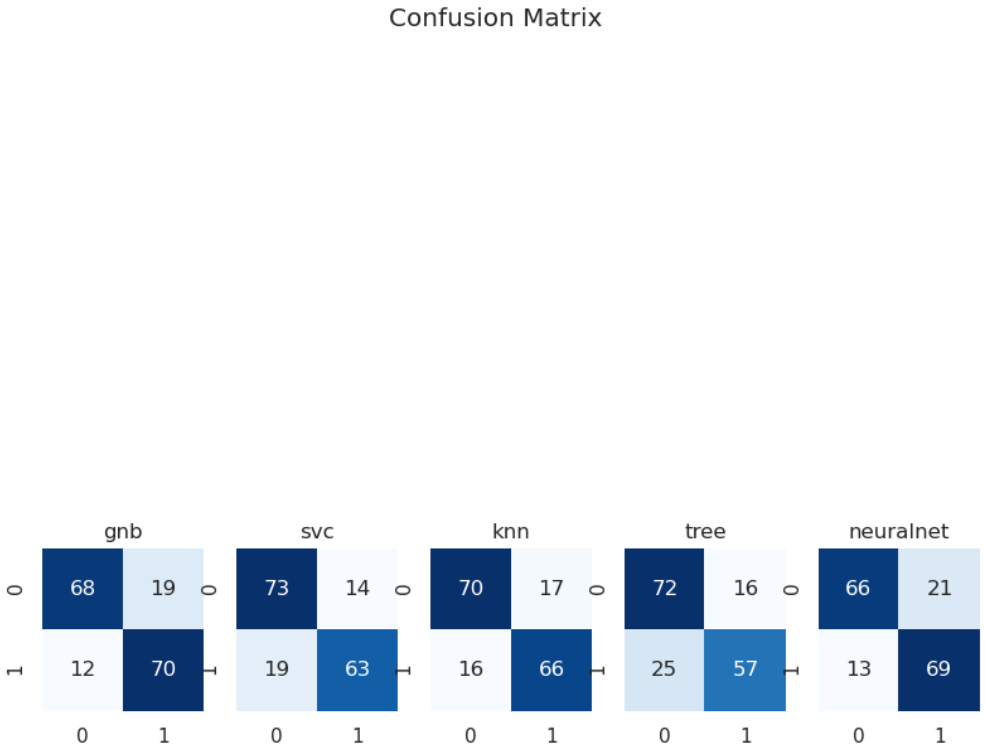

Figura 3: Matriz de confusión con eliminación de valores

Observamos que el algoritmo *Linear SVC* es el que obtiene una mejor puntuación, dado que nos presenta el menos número de falsos negativos y falsos positivos.

2. Imputación de valores

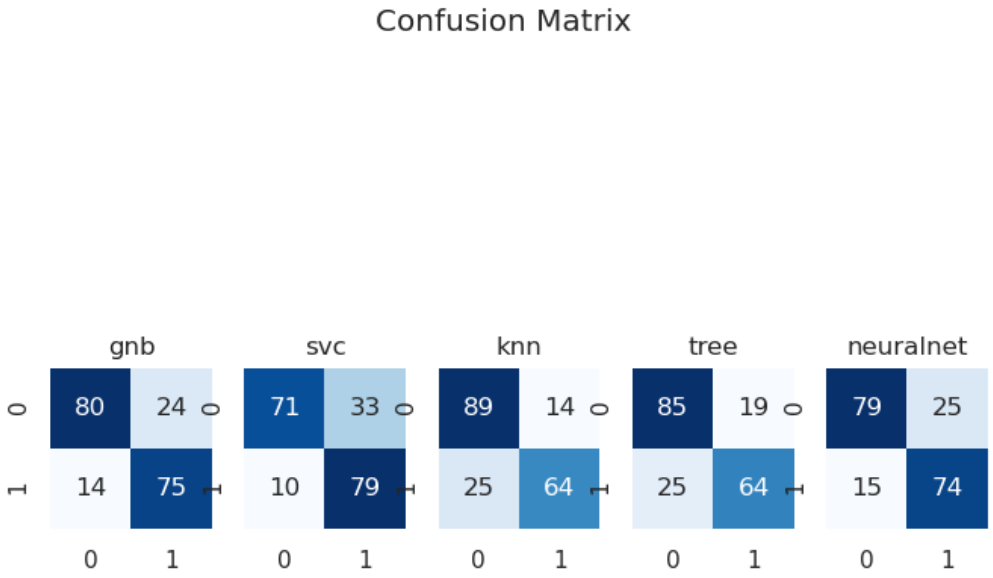

Figura 4: Matriz de confusión con imputación de valores

Observamos que el rendimiento de ciertos algoritmos se degrada en la detección de falsos positivo o falsos negativos. El *Linear SVC* se ve afectado en la detección de falsos positivos, y en este apartado lo supera el *K-NN*.

Correlación entre atributos

Vamos a tratar de observar qué atributos están más relacionados con el resultado del diagnóstico, para determinar cual de ellos es más discriminativo.

Procedemos a generar un histograma para cada uno de los atributos, para ello discretizamos el atributo de la edad.

1. Eliminación de valores

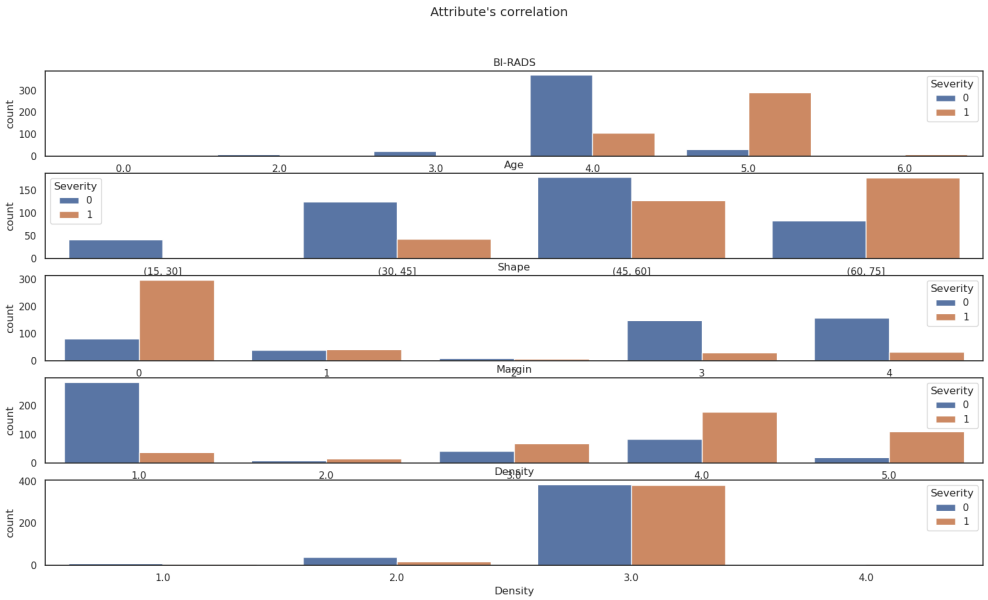

Figura 5: Correlación de atributos con eliminación de valores

Observamos que los atributos que más peso tienen son:

- *Margin*
- *Shape*
- *Age*

2. Imputación de valores

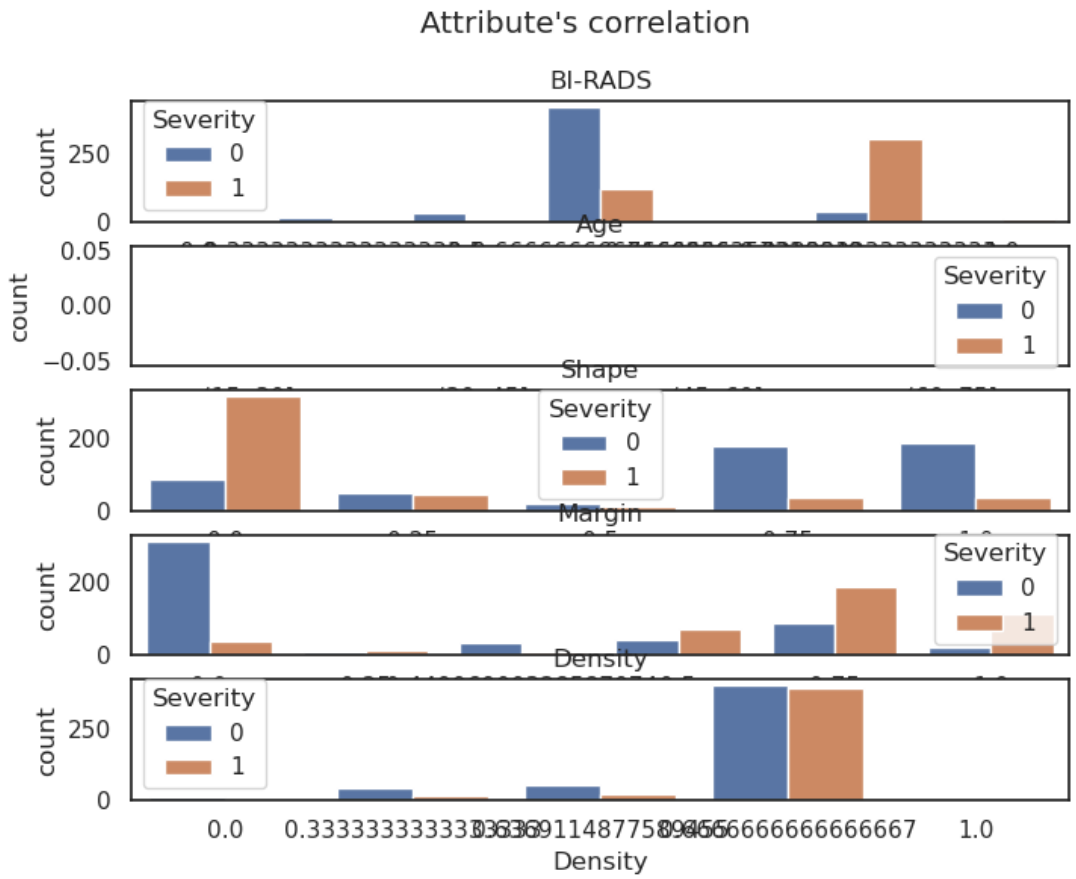

Figura 6: Correlación de atributos con imputación de valores

Observamos que los atributos son menos discriminativos al realizar la imputación, lo cual nos confirma que esta técnica de preprocesamiento es nociva para nuestro caso de estudio.

Apartado 2

Introducción

En este apartado, usaremos distintos algoritmos de *clustering* para resolver un problema de agrupación.