A1.4 Selección de características

En la lectura interactiva de este módulo trabajamos con la base de datos de calidad de vino. Ahí, programamos "a mano", métodos de selección de características. Pero, al final, te comenté que existen otras funciones que te permiten realizar este proceso de forma más sencilla y veloz. En esta actividad deberás generar un modelo de regresión lineal múltiple que contenga solamente las variables seleccionadas por un proceso de selección hacia adelante y eliminación hacia atrás.

Utilizaremos el archivo de nombre "A1.4 Vino Tinto.csv", donde podrás encontrar información para 1,599 observaciones distintas, con 11 mediciones para cada una de ellas, así como con una variable de salida, la calidad asignada a dicho vino. Los datos se descargaron del UCI Machine Learning Repository, y originalmente se reportaron en una publicación científica para la revista Decision Support Systems.

La base de datos cuenta con la siguiente información:

- "**acidezFija**". La acidez fija del vino, medida en gramos de ácido tartárico por decímetro cúbico.
- "**acidezVolatil**". La acidez volátil del vino, medida en gramos de ácido acético por decímetro cúbico.
- "**acidoCitrico**". Gramos de tácito cítrico por decímetro cúbico.
- "**azucarResidual**". Gramos de azúcar por decímetro cúbico.
- "**cloruros**". Gramos de cloruro de sodio por decímetro cúbico.
- "**dioxidoAzufreLibre**". Miligramos de dióxido de azufre libre por decímetro cúbico.
- "**dioxidoAzufreTotal**". Miligramos de dióxido de azufre total por decímetro cúbico.
- "**densidad**". Medida en gramos por centímetro cúbico.
- "**pH**". Valor del vino en la escala de pH.
- "**sulfatos**". Gramos de sulfato de potasio por decímetro cúbico.
- "**alcohol**". Volúmen percentil de alcohol en el vino.
- "**calidad**". Mediana de la calidad otorgada por al menos tres catadores, en escala del 0 (muy malo) al 10 (excelente).

Desarrolla los siguientes puntos en una *Jupyter Notebook*, tratando, dentro de lo posible, que cada punto se trabaje en una celda distinta. Los comentarios en el código siempre son bienvenidos, de preferencia, aprovecha el *markdown* para generar cuadros de descripción que ayuden al lector a comprender el trabajo realizado.

1. Importa los datos del archivo "Vino Tinto.csv" a tu ambiente de trabajo. Este archivo lo encontrarás en la misma página donde descargaste esta plantilla. Revisa las dimensiones del *data frame* e imprime en consola tanto dichas dimensiones como las primeras 5 filas de datos.

2. Separa el *data frame* en datos de entrenamiento y datos de prueba con una proporción 80/20. Es decir, el 80% de los datos se usarán para entrenar el modelo y el resto para validar sus resultados. Asegúrate que la partición sea aleatoria, no es una buena práctica simplemente tomar las primeras observaciones para entrenar y las últimas para probar. Imprime en pantalla las dimensiones de ambos conjuntos de datos. Revisa y asegúrate que la cantidad de observaciones de ambos conjuntos de datos sumen a la cantidad de datos original.

3. Genera la metodología de selección hacia adelante e imprime en consola los índices o los nombres de las características seleccionadas. Para realizar este proceso, te

recomiendo que utilices la función “SequentialFeatureSelector” de la librería “mlxtend.feature_selection”, en este enlace encontrarás más información sobre la misma. Lo más probable es que cuando hayas descargado Anaconda, esta librería no se haya incluido en la distribución, por lo que deberás instalarla manualmente; al final de las instrucciones de la actividad te indico cómo hacerlo. Aquí te dejo una descripción de los parámetros que te recomiendo usar:

a. **estimator**. Un modelo de regresión lineal. Te recomiendo usar la función LinearRegression de la librería sklearn.linear_model.

b. **k_features**. Se puede seleccionar la cantidad de variables de salida que se desean, pero te recomiendo mejor usar un rango, y que el algoritmo determine el número adecuado. Por ejemplo, puedes definir el parámetro como (2,8), si te interesa que el método seleccione entre 2 y 8 variables.

c. **forward**. Determina si se hace selección hacia adelante (True) o hacia atrás (False); en este caso queremos hacer selección hacia adelante.

d. **scoring**. La métrica que se usará para determinar si un modelo es mejor que otro, te recomiendo definirla como ‘r2’ para usar la R cuadrada.

e. **cv**. Si se desea realizar validación cruzada, y cuántas instancias de la misma. Te recomiendo definir este parámetro como 10.

4. Entrenar un modelo que solamente contenga las variables seleccionadas, predecir la respuesta en las observaciones de prueba y medir la capacidad de predicción del modelo usando la R cuadrada, imprimiendo dicho valor en consola. Para el primer paso, simplemente necesitas usar la función *fit* en el modelo de regresión lineal creado previamente, asegurándote de no introducir toda la información de X, sino solo de las variables seleccionadas. Para realizar las predicciones, puedes usar la función *predict* en los datos de prueba, pero recuerda para dichos datos también seleccionar solo las variables de interés. Para el último paso, te recomiendo usar la función *r2_score* de sklearn.metrics.

5. Realizar un proceso de selección hacia atrás a partir de las variables seleccionadas por el método de selección hacia adelante e imprimir en consola los índices o nombres de las variables seleccionadas. Para realizar este proceso, te recomiendo usar la misma función del paso 3, pero definiendo ahora forward=False. También te recomiendo especificar una menor cantidad de variables posibles, por ejemplo: k_features=(2,5).

6. Repetir el paso 4, pero para un modelo que contenga solamente las variables seleccionadas en el paso 5. Imprime en pantalla un breve texto que describa tu opinión sobre la diferencia en R cuadrada medida entre los modelos de los pasos 4 y 6, ¿cuál modelo consideras que es mejor? ¿Por qué?

Instalación de librería mlxtend:

1. Abre “Anaconda”.
2. Abre “Powershell Prompt” (o, en su defecto, CMD.exe Prompt).
3. Escribe en la consola el siguiente código:

```
conda install mlxtend --channel conda-forge
```

4. Te aparecerá un resumen de las instancias que se van a descargar, simplemente teclea la letra ‘y’ y presiona “Enter” para aceptar e iniciar la descarga.
5. Para confirmar que el paquete se haya instalado, escribe en la consola el siguiente código:

```
conda list mlxtend
```

6. Después de unos segundos te aparecerán los datos de la librería, si es que se instaló satisfactoriamente

En caso de tener problemas con la instalación, te recomiendo que uses entonces la función “SequentialFeatureSelector” (mismo nombre), pero de la librería sklearn.feature_selection; en este enlace podrás encontrar más información sobre la función.

Su uso es muy similar, solamente los parámetros cambian un poco, así como sus propiedades. Por ejemplo, con esta función no puedes manejar un rango de variables de salida, necesitas indicar un número de variables a seleccionar específico.