A.14 653602

August 24, 2025

Selección de características

Nombre: Carlos Hernández Márquez

Matrícula: 653602

Firma de honor: "Doy mi palabra que he realizado esta actividad con integridad académica"

En esta actividad se trabaja con la selección de características como paso central para construir un modelo de regresión lineal múltiple más eficiente y preciso. El objetivo es identificar cuáles de las mediciones fisicoquímicas de los vinos tienen un mayor impacto en la predicción de su calidad, reduciendo variables irrelevantes y mejorando la interpretabilidad del modelo.

Se utiliza un conjunto de datos de 1,599 vinos tintos, con once características como acidez, densidad, contenido de alcohol y azúcar residual, y la variable objetivo es la calidad asignada por catadores en una escala del 0 al 10. La actividad combina métodos automáticos de selección hacia adelante y hacia atrás para determinar las variables más significativas y generar un modelo robusto que capture la relación entre los predictores y la calidad del vino.

1 Importación y revisión inicial de los datos

En esta primera etapa se importan los datos del archivo "Vino Tinto.csv" al entorno de trabajo y se realiza una revisión inicial del conjunto de datos. Este paso permite familiarizarse con las dimensiones del data frame, los nombres de las variables y la estructura general de la información, asegurando que los datos estén correctamente cargados antes de cualquier análisis posterior.

[2]:
```
import pandas as pd
df = pd.read_csv("A1.4 Vino Tinto.csv")
print(df.shape)
print(df.columns)
```

```
(1599, 12)
Index(['acidezFija', 'acidezVolatil', 'acidoCitrico', 'azucarResidual',
       'cloruros', 'dioxidoAzufreLibre', 'dioxidoAzufreTotal', 'densidad',
       'pH', 'sulfatos', 'alcohol', 'calidad'],
      dtype='object')
```

El dataset contiene 1,599 observaciones y 12 columnas: 11 variables predictoras (mediciones fisicoquímicas del vino) y 1 variable objetivo (`calidad`).

[3]:
```
df.head(6)
```

[3]:

	acidezFija	acidezVolatil	acidoCitrico	azucarResidual	cloruros	\
0	7.4	0.70	0.00	1.9	0.076	
1	7.8	0.88	0.00	2.6	0.098	
2	7.8	0.76	0.04	2.3	0.092	
3	11.2	0.28	0.56	1.9	0.075	
4	7.4	0.70	0.00	1.9	0.076	
5	7.4	0.66	0.00	1.8	0.075	

	dioxidoAzufreLibre	dioxidoAzufreTotal	densidad	pH	sulfatos	alcohol	\
0	11.0	34.0	0.9978	3.51	0.56	9.4	
1	25.0	67.0	0.9968	3.20	0.68	9.8	
2	15.0	54.0	0.9970	3.26	0.65	9.8	
3	17.0	60.0	0.9980	3.16	0.58	9.8	
4	11.0	34.0	0.9978	3.51	0.56	9.4	
5	13.0	40.0	0.9978	3.51	0.56	9.4	

	calidad
0	5
1	5
2	5
3	6
4	5
5	5

La inspección de las primeras filas permite hacerse una idea preliminar de qué variables podrían tener mayor influencia sobre la calidad, por ejemplo, `alcohol`, `acidezVolatil` o `densidad`. Sin embargo, esta evaluación visual es limitada, y es necesario aplicar métodos sistemáticos de selección de características para determinar objetivamente las variables más relevantes.

2 Separación de datos en entrenamiento y prueba

Para evaluar el desempeño del modelo de manera objetiva, se divide el dataset en conjuntos de entrenamiento y prueba con una proporción 80/20. La partición se realiza de forma aleatoria para evitar sesgos que podrían surgir al tomar las primeras o últimas observaciones del dataset.

[8]:
```
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, train_size = 0.8)

print("Train:", train.shape)
print("Test:", test.shape)
print("Total:", train.shape[0] + test.shape[0])
print("Original:", df.shape[0])
```

```
Train: (1279, 12)
Test: (320, 12)
Total: 1599
Original: 1599
```

Tras la separación, el conjunto de entrenamiento contiene 1,279 observaciones y 12 columnas, mientras que el conjunto de prueba tiene 320 observaciones y 12 columnas. La suma de ambos conjuntos coincide con el total del dataset original (1,599 observaciones), asegurando que no se pierden datos durante la partición.

3 Selección hacia adelante de características

Para optimizar el modelo de regresión lineal múltiple, se aplica la **selección hacia adelante**, un método que identifica de manera sistemática las variables predictoras que aportan mayor explicación de la variable objetivo. Este enfoque permite construir un modelo más eficiente, reduciendo el número de variables innecesarias y evitando sobreajuste.

[27]:
```
from sklearn.linear_model import LinearRegression
from mlxtend.feature_selection import SequentialFeatureSelector

X = df.drop("calidad", axis=1)
y = df["calidad"]

lr = LinearRegression()

sfs_forward = SequentialFeatureSelector(
    estimator=lr,
    k_features=(2, 8),
    forward=True,
    scoring='r2',
    cv=10
)

sfs_forward = sfs_forward.fit(X, y)

print("\nÍndices de características seleccionadas:")
print(list(sfs_forward.k_feature_idx_))

print("\nNombres de características seleccionadas:")
print(list(sfs_forward.k_feature_names_))
```

```
Índices de características seleccionadas:
[1, 4, 5, 6, 8, 9, 10]

Nombres de características seleccionadas:
['acidezVolatil', 'cloruros', 'dioxidoAzufreLibre', 'dioxidoAzufreTotal', 'pH',
'sulfatos', 'alcohol']
```

El proceso seleccionó las siguientes variables: `acidezVolatil`, `cloruros`, `dioxidoAzufreLibre`, `dioxidoAzufreTotal`, `pH`, `sulfatos` y `alcohol`. Técnicamente, esto indica que estas características aportan la mayor varianza explicativa sobre la calidad del vino en comparación con las restantes. Algunas, como `alcohol` y `acidezVolatil`, eran esperables por la inspección visual previa, mientras

que otras, como `dioxidoAzufreLibre` y `dioxidoAzufreTotal`, reflejan la relevancia de factores químicos menos evidentes a simple vista.

Estos resultados permiten definir un subconjunto de variables significativas sobre las cuales entrenar el modelo, lo que conecta directamente con el siguiente paso: **entrenar un modelo de regresión lineal múltiple utilizando únicamente las variables seleccionadas**, para evaluar su capacidad predictiva mediante la métrica R^2 en el conjunto de prueba.

4 Entrenamiento del modelo con variables seleccionadas y evaluación

En este paso se entrena un **modelo de regresión lineal múltiple** utilizando únicamente las variables seleccionadas por el método de selección hacia adelante. Se busca evaluar la capacidad predictiva del modelo sobre el conjunto de prueba, usando la métrica R^2 como indicador de qué proporción de la varianza de la calidad del vino es explicada por las variables seleccionadas.

[ ]:
```
from sklearn.metrics import r2_score

selected_features = list(sfs_forward.k_feature_names_)

X_train = train[selected_features]
y_train = train["calidad"]

X_test = test[selected_features]
y_test = test["calidad"]

# Entrenar
lr.fit(X_train, y_train)

# Predecir
y_pred = lr.predict(X_test)

# Calcular R^2
r2 = r2_score(y_test, y_pred)

print("\nVariables seleccionadas (hacia adelante):")
print(selected_features)

print("\nR² en datos de prueba (modelo hacia adelante): ")
print(r2)
```

```
Variables seleccionadas (hacia adelante):
['acidezVolatil', 'cloruros', 'dioxidoAzufreLibre', 'dioxidoAzufreTotal', 'pH',
'sulfatos', 'alcohol']

R² en datos de prueba (modelo hacia adelante):
0.3554446360037674
```

El modelo entrenado con las variables `acidezVolatil`, `cloruros`, `dioxidoAzufreLibre`, `dioxidoAzufreTotal`, `pH`, `sulfatos` y `alcohol` obtuvo un R^2 de 0.355, lo que indica que aproximadamente el 35.5% de la variabilidad en la calidad del vino puede explicarse mediante estas características. Aunque este valor no es extremadamente alto, confirma que estas variables aportan información significativa y permite validar la utilidad de la selección hacia adelante.

Estos resultados sirven como base para el siguiente paso: aplicar la **selección hacia atrás** a partir de estas mismas variables, con el fin de determinar si un subconjunto más reducido puede mantener o mejorar la capacidad predictiva del modelo, optimizando aún más su eficiencia.

5 Selección hacia atrás de características

A continuación se aplica la **selección hacia atrás** sobre el subconjunto de variables previamente seleccionado mediante el método hacia adelante. Este procedimiento elimina de manera iterativa las variables menos relevantes para maximizar la métrica R^2, con el objetivo de reducir aún más el número de predictores sin comprometer significativamente la capacidad explicativa del modelo.

[33]:
```
sfs_backward = SequentialFeatureSelector(
    estimator=lr,
    k_features=(2, 5),        # rango entre 2 y 5 variables
    forward=False,            # hacia atrás
    scoring='r2',
    cv=10
)

sfs_backward = sfs_backward.fit(df[selected_features], y)

print("\nÍndices de características seleccionadas (hacia atrás):")
print(list(sfs_backward.k_feature_idx_))

print("\nNombres de características seleccionadas (hacia atrás):")
print(list(sfs_backward.k_feature_names_))
```

```
Índices de características seleccionadas (hacia atrás):
[0, 1, 3, 5, 6]

Nombres de características seleccionadas (hacia atrás):
['acidezVolatil', 'cloruros', 'dioxidoAzufreTotal', 'sulfatos', 'alcohol']
```

El proceso identificó como variables más relevantes `acidezVolatil`, `cloruros`, `dioxidoAzufreTotal`, `sulfatos` y `alcohol`. Comparado con la selección hacia adelante, se eliminan `dioxidoAzufreLibre` y `pH`, lo que sugiere que su contribución a la explicación de la calidad del vino es menor cuando se consideran conjuntamente las otras variables. Este subconjunto optimizado permite entrenar un modelo más eficiente, reduciendo complejidad y potencial sobreajuste, al tiempo que mantiene las características más influyentes sobre la variable objetivo.

6 Comparación de modelos y evaluación de R^2

En esta etapa se construyó un nuevo modelo de regresión lineal empleando únicamente las variables seleccionadas mediante el método de eliminación hacia atrás. El objetivo es comparar el desempeño de este modelo reducido con el modelo completo del paso 4, utilizando como métrica principal el coeficiente de determinación (R^2) en el conjunto de prueba.

[ ]:
```
selected_features_backward = list(sfs_backward.k_feature_names_)

X_train_b = train[selected_features_backward]
y_train_b = train["calidad"]

X_test_b = test[selected_features_backward]
y_test_b = test["calidad"]

# Entrenar
lr.fit(X_train_b, y_train_b)

# Predecir
y_pred_b = lr.predict(X_test_b)

# Calcular R^2
r2_b = r2_score(y_test_b, y_pred_b)

print("\nVariables seleccionadas (hacia atrás):")
print(selected_features_backward)

print("\nR² en datos de prueba (modelo hacia atrás):")
print(r2_b)
```

```
Variables seleccionadas (hacia atrás):
['acidezVolatil', 'cloruros', 'dioxidoAzufreTotal', 'sulfatos', 'alcohol']

R² en datos de prueba (modelo hacia atrás):
0.3568560100040997
```

La comparación entre el modelo completo y los modelos reducidos obtenidos por selección de variables muestra un comportamiento interesante.

- El **modelo hacia atrás**, con solo **5 variables** (`acidezVolatil`, `cloruros`, `dioxidoAzufreTotal`, `sulfatos`, `alcohol`), alcanza un (R^2) en datos de prueba de **0.3569**.

- El **modelo hacia adelante**, que incluye **7 variables** (`acidezVolatil`, `cloruros`, `dioxidoAzufreLibre`, `dioxidoAzufreTotal`, `pH`, `sulfatos`, `alcohol`), obtiene un (R^2) muy similar: **0.3554**.

Al comparar estos resultados con el modelo completo, se observa que los valores de (R^2) se mantienen prácticamente iguales, lo que indica que tanto el modelo hacia atrás como el modelo

hacia adelante logran **reducir la complejidad** sin sacrificar de manera significativa la capacidad explicativa.

En particular, el modelo hacia atrás resulta más atractivo: con **menos predictores (5 vs 7)** logra un (R^2) incluso ligeramente superior al del modelo hacia adelante. Esto refleja un mejor balance entre **parsimonia y poder predictivo**, lo cual hace que sea el modelo más eficiente de los tres comparados.

Aunque ambos métodos de selección son válidos, el **modelo hacia atrás es preferible** porque conserva el poder explicativo del modelo original al mismo tiempo que reduce de forma más efectiva el número de variables.