Act10

Daniel Rojas

March 2025

1 Introducción

La regresión lineal multivariable es el caso generalizado de la regresión lineal simple. La único diferencia entre ambas es la cantidad de variables independientes, o features, de la función estimadora, para la simple solo se usa una variable, mientras que en la multivariable se usan 2 o más.

Al igual que la regresión lineal simple, la regresión lineal multivariable es un algoritmo supervisado del aprendizaje automátizado que destaca por su facil implementación y bajo coste computacional para las computadoras.

La ecuación de una función lineal con n variables independientes es la sigueinte:

$$y = m_1x_1 + m_2x_2 + ... + m_nx_n + b \tag{1}$$

Donde n es el número de variables independientes, $x_1, x_2, ..., x_n$ son las variables independientes, $m_1, m_2, ..., m_n$ los coeficientes de cada variable independiente, b la intersección con el eje y, y y es la variable dependiente.

Funciona de forma básicamente idéntica a la regresión lineal simple, se busca que el error cuadrático medio entre los valores reales y los predecidos sea el mínimo posible.

2 Metodología

Para completar la actividad, se usó de base el capítulo de regresión lineal del libro Aprende Machine Learning, adjuntado al final en las referencias.

De la siguiente forma se pueden cargar las librerías y dataset a utilizar:

```
# Imports necesarios
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (16, 9)
```

```
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

```
data = pd.read_csv("articulos_ml.csv")
```

Para empezar, hay que realizar un preprocesado de los datos. Primero hay que decidir que columna del dataset usaremos como las features, o las variable x_1 y x_2, y cual como nuestra etiqueta, o la variable y. En este caso, se eligió como features la cantidad de palabras de cada artículo y la suma de cantidad de enlaces, comentarios e imágenes en un atrículo; y como etiqueta la cantidad de veces que dicho artículo fue compartido.

Sin embargo, primero hay que filtrar algunos datos anómalos de la gráfica, específicamente los que tiene un conteo de palabras mayor a 3500 y los datos que tienen el número de veces compartido mayor a 80,000. Para filtrar los resultados, se puede ejecutar lo siguiente:

```
#Datos recortados con el m ximo de palabras restringido a 3500 y
    ↪ el m ximo de compartidos limitado a 80000
filtered_data = data[(data['Word count'] <= 3500) & (data['# Shares
    ↪ '] <= 80000)]
```

Después, hay que crear nuestro nuevo dataset con las features descritas anteriormente:

```
#Separar el dataset en valores de entrada y en las etiquetas.

#Crear la columna con la suma de la cantidad de links, comentarios
    ↪ e im genes en un art culo
suma = (filtered_data["# of Links"] + filtered_data['# of comments'
    ↪ ].fillna(0) + filtered_data['# Images video'])

#Crear el nuevo dataset y lo separa en valores de entrada y en
    ↪ etiquetas
dataX2 =  pd.DataFrame()
dataX2["Word count"] = filtered_data["Word count"]
dataX2["suma"] = suma
XY_train = np.array(dataX2)
z_train = filtered_data['# Shares'].values
```

Por último, hay que entrenar nuestro modelo de regresión lineal y observar su desempeño:

```
#Realizar la regresi n lineal multivariable

regr2 = linear_model.LinearRegression()

#Entrenar el modelo
regr2.fit(XY_train, z_train)

#Se realizan predicciones para posteriormente obtener el error
    ↪ cuadr tico medio, y el puntaje de varianza
z_pred = regr2.predict(XY_train)
```

```
10
11 # Los coeficientes
12 print('Coefficients: \n', regr2.coef_)
13 # Error cuadr tico medio
14 print("Mean squared error: %.2f" % mean_squared_error(z_train,
       ↪ z_pred))
15 # Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
16 print('Variance score: %.2f' % r2_score(z_train, z_pred))
```

EL modelo tiene de coeficientes $m_1 = 6.63$ y $m_2 = 483.41$, un error cuadrático medio de 352122816.48, y un valor de varianza de 0.11.

3 Resultados

El modelo de regresión lineal multivariable obtuvo un error cuadrático medio muy elevado y un valor de varianza demasiado bajo como para segurar que el modelo pueda ser útil para realizar predicciones. Aun así, puede ser útil para observar la tendencia de los datos. Por otro lado, comparado con los resultados mostrados en el libro, el modelo de regresión lineal múltiple tiene mejores resultados que el modelo de regresión lineal simple.

4 Conclusión

Por lo general, una regresión lineal múltiple brinda mejores resultados que una regresión lineal simple, siempre y cuando todas las features usadas no describan tan fuertemente el comportamiento de las demás. En otras palabras, si después de hacer un PCA aún tenemos 2 o más features, es mejor realizar una regresión lineal múltiple.

Por último, el único inconveniente de la regresión lineal multivariable es que en muchos casos no podremos observar el hiperplano que mejor se ajusta a los datos, por lo que la interpretación de este modelo es más complicada en comparación de la regresión lineal simple.

5 Referencias

Bagnato, J. I. (2020). Aprende Machine Learning.