Instituto Tecnológico y de Estudios Superiores de Monterrey

TC3006C

Reporte módulo 2: Machine Learning
Grupo 101

Evidencia 2:

Análisis y Reporte sobre el desempeño del modelo

Docente:
Ivan Mauricio Amaya Contreras

Alumno:

Alan Mondragón Rivas

A01734565

Campus Monterrey

Fecha de entrega:

16 de septiembre 2022

Fish Market Dataset

Para esta evidencia se utilizó un dataset de Kaggle el cual nos proporciona información acerca de 7 especies de peces que se comercializan. El fin de este dataset es poder realizar un modelo de machine learning que pueda predecir con base a sus características un cierto peso y con este posteriormente poder asignarle un cierto peso arbitrario para comercializarlos.

Entre las características de los pescados podemos encontrar lo siguiente:

- **Species:** Clase del pez
- **Weight:** Peso del pez en gramos (g)
- **Length1:** Longitud vertical (cm)
- **Length2:** Longitud diagonal (cm)
- **Length3:** Longitud cruzada (cm)
- **Height:** Altura del pez (cm)
- **Width:** Anchura diagonal del pez (cm)

Análisis previo de los datos

Antes de escoger un modelo de regresión para hacer nuestras predicciones, primero debemos de conocer cuál es la relación de los datos. Para ello podemos ver la matriz de correlaciones de nuestros datos para ver si existe linealidad entre dos o más variables, teniendo lo siguiente:

	Weight	Length1	Length2	Length3	Height	Width
Weight	1.000000	0.915712	0.918618	0.923044	0.724345	0.886507
Length1	0.915712	1.000000	0.999517	0.992031	0.625378	0.867050
Length2	0.918618	0.999517	1.000000	0.994103	0.640441	0.873547
Length3	0.923044	0.992031	0.994103	1.000000	0.703409	0.878520
Height	0.724345	0.625378	0.640441	0.703409	1.000000	0.792881
Width	0.886507	0.867050	0.873547	0.878520	0.792881	1.000000

Gracias a esta matriz de correlaciones podemos darnos cuenta de que todas las longitudes están fuertemente relacionadas. Para el caso de la altura vemos que realmente no está tan fuertemente relacionada ni con el peso ni con las longitudes por lo que puede que no sea una característica que nos sea útil para implementar nuestro modelo. Por otro lado, vemos que la anchura diagonal se relaciona correlaciona más con los demás datos que la altura, por lo que podemos usar esta característica para poder realizar nuestro modelo.

Modelo de regresión utilizado

Para nuestro conjunto de datos tenemos que las longitudes 1,2 y 3 están fuertemente relacionadas con el peso, pero sobre todo entre sí, por lo que haremos una nueva característica donde se incluyan las tres longitudes de tal manera que siga siendo útil para nuestro modelo y sobre todo reducir el número de dimensiones. Lo que haremos será multiplicar las 3 longitudes y sacarle la raíz cúbica a dicha operación para no tener un valor tan grande para los coeficientes del modelo. Esta nueva característica será la primera de nuestro modelo. Mientras que la anchura diagonal por sí sola será nuestra segunda característica y con buscaremos un modelo de regresión lineal de segundo orden que se pueda ajustar a nuestras características para encontrar un modelo para predecir el peso de los peces. Por lo que buscaremos que nuestro modelo se ajuste a estas dos características.

RStudio

En módulo de estadística utilizamos la herramienta de RStudio, la cual tiene una sintaxis muy simple al igual que python con la diferencia que R es más especializada para trabajar con herramientas estadísticas. Por lo que para analizar cómo sería nuestro modelo de regresión lineal tomando en cuenta la anchura y nuestra característica creada de las longitudes como variables independientes una de la otra es decir θ1 y θ2, así como la interacción de ambas características, lo cual nos ayuda para ver qué tan significante es tomar en cuenta la interacción entre estas. El script de R se encuentra con el nombre de **Fish_test.Rmd**.

```
Call:
lm(formula = y ~ Fs$Width * L3)

Residuals:
     Min       1Q   Median       3Q      Max
-217.758  -39.504   -3.285   27.637  299.713

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  71.1874    40.7643   1.746  0.08274 .
Fs$Width    -53.4969    10.9824  -4.871 2.71e-06 ***
L3           -6.7464     2.0812  -3.242  0.00146 **
Fs$Width:L3   5.3301     0.3324  16.033  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 78.16 on 155 degrees of freedom
Multiple R-squared:  0.9532,    Adjusted R-squared:  0.9523
F-statistic:  1053 on 3 and 155 DF,  p-value: < 2.2e-16
```

Lo que podemos ver en la imagen anterior es un resumen del modelo generado en R. La parte de coeficientes es la que más nos interesa analizar ya que en la columna de *Estimate* tenemos los valores de θ, θ1, θ2 y θ3; donde θ3 corresponde a la interacción de x1 y x2. Sabemos que esta interacción es significativa ya que tenemos un valor t mucho mayor de 0 que nos indica que la interacción es

significativa para hallar nuestra H0 por lo que nuestro modelo quedaría de la siguiente manera: $h_\theta = \theta + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2$

Sustituyendo los coeficientes y nuestras variables tenemos lo siguiente:

$$F_{\text{weight}} = 71.1874 - 53.4969 F_{\text{width}} - 6.7464\text{L} + 5.3301 F_{\text{width}}\text{L}$$

Python

El script de Python se encuentra con el nombre de **Evi1_pt2.py**.

Gracias a esta nueva característica que creamos a partir de una relación entre las longitudes, podemos utilizar esta información para implementarla en python con sklearn y probar con diferentes modelos de regresión con diferentes características para saber cuáles de esas características son más relevantes para poder hacer nuestra estimación del peso de los peces. Por lo que vamos a probar 4 modelos de distintas características y comparar sus desempeños y para ello necesitamos usar las siguientes librerías.

Librerías Utilizadas

- **Pandas:** Para importar y visualizar el dataset
- **Matplotlib:** Para poder graficar los resultados obtenidos
- **Sklearn:** Librería para el uso de herramientas de Machine Learning, pero solo usaremos las siguientes:
    - **Train_test_split:** Nos permite separar datos de entrenamiento y prueba de manera más sencilla
    - **Linear_model:** Para usar el modelo ya implementado de regresión lineal
    - **Metrics:** Para usar métricas estadísticas que nos ayuden a evaluar el desempeño de los modelos

Añadimos las nuevas características al modelo

- **L123:** Contiene la raíz cúbica del producto de las longitudes 1,2 y 3
- **WidthXL123:** La interacción entre Anchura la nuestra nueva característica

```
# Creamos nuestras caracteristicas nuevas para nuestro modelo
data['L123']= (data['Length1'] * data['Length2'] * data['Length3'])**(1/3)
data['WidthXL123'] = data['L123']*data['Width']
```

Al agregar estas nuevas columnas quiere decir que vamos a tener más interacciones por lo que hay que checar la matriz de correlación con estos cambios y ver qué tan relevantes pueden ser para nuestro modelo lineal.

	Weight	Length1	Length2	Length3	Height	Width	L123	WidthXL123
Weight	1.000000	0.915712	0.918618	0.923044	0.724345	0.886507	0.920534	0.971972
Length1	0.915712	1.000000	0.999517	0.992031	0.625378	0.867050	0.998814	0.957006
Length2	0.918618	0.999517	1.000000	0.994103	0.640441	0.873547	0.999491	0.958637
Length3	0.923044	0.992031	0.994103	1.000000	0.703409	0.878520	0.996907	0.953540
Height	0.724345	0.625378	0.640441	0.703409	1.000000	0.792881	0.656937	0.693852
Width	0.886507	0.867050	0.873547	0.878520	0.792881	1.000000	0.874339	0.938106
L123	0.920534	0.998814	0.999491	0.996907	0.656937	0.874339	1.000000	0.957927
WidthXL123	0.971972	0.957006	0.958637	0.953540	0.693852	0.938106	0.957927	1.000000

Implementación de diferentes modelos

Para poder comparar que se use el modelo con las características óptimas para poder predecir nuestra y que es el peso del pez, se probarán **6 modelos** donde cada uno posee distintas características. Además, se creó un arreglo donde contiene todos las **X** y este arreglo se mete en una función donde se evalúa cada modelo para ver el rendimiento y poder decir cuál es la mejor opción para predecir el peso del pez.

```
# Caracteristicas para los distintos modelos
x1 = data[['Length1','Length2','Length3','Width','Height','L123','WidthXL123']]
x2 = data[['Length1','Length2','Length3','Width','L123','WidthXL123']]
x3 = data[['Length1','Length2','Length3','Height','Width']]
x4 = data[['Height','Width','L123','WidthXL123']]
x5 = data[['Height','Width','WidthXL123']]
x6 = data[['Height','Width','L123']]

chars_x = [x1,x2, x3, x4, x5, x6]
```

Se creó una función donde se crea el modelo de regresión lineal múltiple y se evalúa su rendimiento dependiendo de los conjuntos de entrenamiento y prueba y de las características de X_n.

```
# Función de evaulación de desempeño del modelo de regresión lineal
def evalMLR(X_train, X_test, y_train, y_test):
  # Declaración de modelo de regresión lineal
  mlr = LinearRegression()
  # Ajuste de nuestro modelo
  mlr.fit(X_train, y_train)
  #Predicciónes del modelo
  y_hat_train = mlr.predict(X_train)
  y_hat_test = mlr.predict(X_test)
  # Validación
  sc_train = r2_score(y_train, y_hat_train)
  sc_test = r2_score(y_test, y_hat_test)
  mse = mean_squared_error(y_test, y_hat_test)
  print("R2_Train = "+ "{:.6}".format(sc_train*100)+ "%\tR2_Test = "+
        "{:.6}".format(sc_test*100)+"%\tECM = "+ "{:.6}".format(mse)+"\n")
```

Métricas de desempeño

Para evaluar el desempeño de cada modelo usamos dos métricas estadísticas para evaluar que tan bien se ajustó el conjunto de entrenamiento para hacer las predicciones con el conjunto de prueba.

- **Coeficiente de determinación (r^2):** Nos indica que tan bien se ajustan los datos con nuestro modelo mediante los residuos entre los valores reales y los estimados
- **Error cuadrático medio (ECM):** Es la suma de la varianza y el cuadrado sesgo de las predicciones

Predicciones de prueba

- **Entradas**: Conjuntos de entrenamiento y prueba para nuestros modelos
- **Valor esperado:** Valores de los pesos de los peces reales de nuestro dataset
- **Valor obtenido:** Valores de los pesos de los peces predichos por los modelos

Se usaron de manera aleatoria **5** números que son un hiperparámetro del modelo de regresión lineal de Sklearn para que se escogieron de manera aleatoria distintas muestras para los conjuntos de prueba y entrenamiento y estas muestras aleatorias se evaluaron en los 6 diferentes modelos creados, donde los resultados fueron los siguientes:

	MUESTRA 1			MUESTRA 3			MUESTRA 5		
Modelo 1	R2_Train = 97.6622%	R2_Test = 96.6109%	ECM = 4270.07	R2_Train = 97.8638%	R2_Test = 94.7471%	ECM = 3936.63	R2_Train = 97.2638%	R2_Test = 98.0484%	ECM = 1534.45
Modelo 2	R2_Train = 96.9786%	R2_Test = 96.1907%	ECM = 4799.51	R2_Train = 97.4301%	R2_Test = 93.4178%	ECM = 4932.81	R2_Train = 96.6881%	R2_Test = 97.5643%	ECM = 1915.08
Modelo 3	R2_Train = 88.7399%	R2_Test = 87.0841%	ECM = 16273.2	R2_Train = 89.583%	R2_Test = 80.2878%	ECM = 14772.7	R2_Train = 89.3725%	R2_Test = 81.624%	ECM = 14448.0
Modelo 4	R2_Train = 97.5444%	R2_Test = 96.624%	ECM = 4253.56	R2_Train = 97.8158%	R2_Test = 94.4075%	ECM = 4191.11	R2_Train = 97.1935%	R2_Test = 97.9089%	ECM = 1644.08
Modelo 5	R2_Train = 96.8633%	R2_Test = 96.4446%	ECM = 4479.61	R2_Train = 97.3724%	R2_Test = 93.1055%	ECM = 5166.86	R2_Train = 96.8572%	R2_Test = 96.2721%	ECM = 2931.02
Modelo 6	R2_Train = 88.2887%	R2_Test = 86.4876%	ECM = 17024.7	R2_Train = 89.1186%	R2_Test = 80.0402%	ECM = 14958.2	R2_Train = 88.8925%	R2_Test = 81.6013%	ECM = 14465.7
	MUESTRA 2			MUESTRA 4					
Modelo 1	R2_Train = 97.7428%	R2_Test = 95.7405%	ECM = 5822.7	R2_Train = 97.717%	R2_Test = 95.895%	ECM = 4662.03			
Modelo 2	R2_Train = 97.2289%	R2_Test = 94.863%	ECM = 7022.18	R2_Train = 97.0988%	R2_Test = 95.4591%	ECM = 5157.13			
Modelo 3	R2_Train = 88.3097%	R2_Test = 87.134%	ECM = 17587.6	R2_Train = 88.636%	R2_Test = 86.6419%	ECM = 15170.8			
Modelo 4	R2_Train = 97.6376%	R2_Test = 95.7447%	ECM = 5816.99	R2_Train = 97.5779%	R2_Test = 95.965%	ECM = 4582.56			
Modelo 5	R2_Train = 96.9434%	R2_Test = 95.6066%	ECM = 6005.7	R2_Train = 97.0759%	R2_Test = 95.2547%	ECM = 5389.18			
Modelo 6	R2_Train = 87.9496%	R2_Test = 86.4986%	ECM = 18456.3	R2_Train = 88.2658%	R2_Test = 86.0365%	ECM = 15858.3			

Podemos ver que para tenemos dos modelos que poseen el coeficiente de determinación y la menor cantidad de residuos, pero en algunas pruebas un modelo es ligeramente mejor que el otro. Por lo que lo que haremos será quedarnos con el modelo más simple que en este caso es el **modelo 4** al tener menos características en cuenta que el modelo 1. Por lo que nuestro mejor modelo de regresión lineal múltiple nos quedó de la siguiente manera:

$$h_{\text{weight}} = \theta + \theta_1 x_{\text{height}} + \theta_2 x_{\text{width}} + \theta_3 x_{\text{L123}} + \theta_3 x_{\text{widthXL123}}$$

Evaluación del modelo con conjuntos de entrenamiento y prueba

Se corrieron 5 muestras aleatorias para poder entrenar, probar y validar el modelo donde se obtuvieron los siguientes rendimientos.

Muestra	r^2 (Train)	r^2 (Test)	ECM
1	97.5444%	96.6240%	4253.56
2	97.6376%	95.7447%	5816.99
3	97.8158%	94.4075%	4191.11
4	97.5779%	95.9650%	4582.56
5	97.1935%	97.9089%	1644.08

Podemos ver que los datos predichos por el modelo realmente se acercan a los valores reales; sin embargo, esta no es la única forma en la que se tiene para decir si nuestro modelo es lo suficientemente bueno como para poder confiar en las estimaciones que nos arroja de los valores. Por lo que vamos a tener que hacer un análisis sobre el sesgo o bias y la varianza de los datos obtenidos del modelo y observar el tipo de ajuste que tiene nuestro modelo.

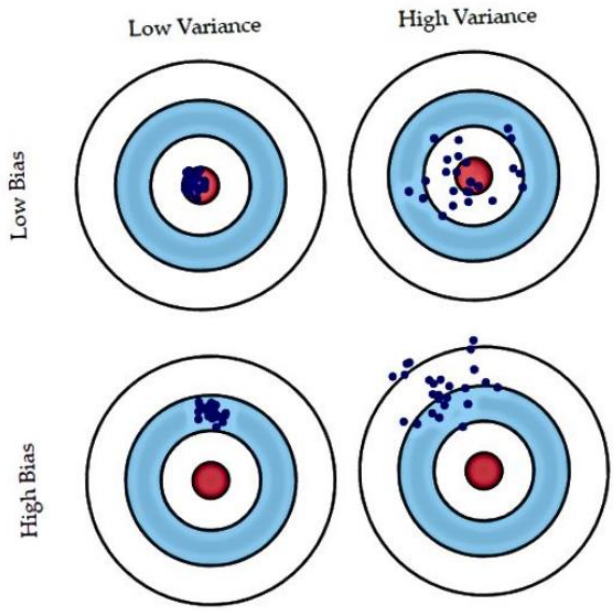
Análisis del grado de sesgo o bias y varianza

SESGO O BIAS

- **Bajo:** Supuestos débiles con respecto a la forma funcional de la asignación de entradas a salidas
- **Alto:** Fuertes supuestos con respecto a la forma funcional del mapeo de entradas a salidas

VARIANZA

- **Baja:** Pequeños cambios en el modelo con cambios en el conjunto de datos de entrenamiento
- **Alta:** Grandes cambios en el modelo con cambios en el conjunto de datos de entrenamiento



Estos fueron los promedios de sesgo y varianza de cada una de las muestras para entrenar y probar el modelo:

Muestra	Sesgo prom.	Varianza prom.
1	4408.923	223.173
2	5774.135	346.702
3	4257.437	220.319
4	4593.347	437.855
5	1588.349	223.119

Podemos ver que en promedio se tiene menos varianza entre las diferentes pruebas y que puede ser una varianza. Por otro lado, el sesgo promedio entre muestras vemos que varía un poco más por lo que puede tratarse de que nuestro modelo tiene un sesgo un poco alto.

Nivel de ajuste del modelo

- **Subajustado:** alto sesgo y poca varianza → error de entrenamiento y prueba alto
- **Sobreajustado:** bajo sesgo y alta varianza → error de entrenamiento bajo y error de prueba alto
- **Balanceado:** bajo sesgo y poca varianza (ideal) → error de entrenamiento y prueba bajo

Utiliza técnicas de regularización para mejorar el desempeño del modelo

En la siguiente tabla podemos encontrar como es que mejoro o no el modelo (*RL*) con las técnicas de regularización y ver el desempeño del modelo usando los métodos de *sklearn* de **Lasso** y **Rigde** para las técnicas de **L1** y **L2** respectivamente para las 5 muestras aleatorias que tomamos.

Muestra	Modelo	r^2(Train)	r^2(Test)	ECM
1	RL	97.54437019%	96.62397438%	4253.560749
1	L1	97.54436962%	96.62332606%	4254.377584
1	L2	97.54436983%	96.62342038%	4254.258749
2	RL	97.63755155%	95.74467018%	5816.989804
2	L1	97.63755099%	95.74364174%	5818.395669
2	L2	97.63755124%	95.74383973%	5818.125017
3	RL	97.81581284%	94.40751838%	4191.107047
3	L1	97.81581240%	94.40784534%	4190.862020
3	L2	97.81581250%	94.40791485%	4190.809927
4	RL	97.57792952%	95.96497547%	4582.558173
4	L1	97.57792897%	95.96604281%	4581.345995
4	L2	97.57792902%	95.96611975%	4581.258615
5	RL	97.19346661%	97.90892849%	1644.081232
5	L1	97.19346615%	97.90913019%	1643.922649
5	L2	97.19346625%	97.90928554%	1643.800507

Los scripts utilizados para esta evidencia se encuentran en la carpeta de **Codes**.

Podemos ver que no hay una gran diferencia en cuanto al uso de técnicas de regularización tanto los método de L1 y L2 varían entre ellas y nuestro modelo original en a partir de 6 cifras decimales por lo que podemos decir que no hay una mejora tan significativa usando técnicas de regularización específicamente para nuestro, por lo que podemos concluir que nuestro modelo si es muy bueno para

predecir datos de nuestro dataset y que por lo tanto es un modelo balanceado ya que posee poca varianza y poco sesgo. En conclusión, es muy importante escoger siempre un modelo adecuado para cada aplicación y siempre analizar las variables que tenemos y probar con diferentes combinaciones para poder obtener mejores resultados posible para hacer nuestras predicciones, así como usar técnicas de regularización para mejorar el desempeño de nuestros modelos.