MDS7104 - Aprendizaje de maquinas

Tarea 1

Cristopher Urbina Herrera

2024-09-08

Tabla de contenidos

1 Regresión lineal y descomposición de la sesgo-varianza

1.1 a)

El modelo lineal $h(z)$ tendrá un sesgo elevado ya que intenta estimar una función cuadrática con una función lineal.

1.2 b)

Para calcular el coeficiente de regresión θ se puede utilizar la fórmula:

$$\theta = (X^T X)^{-1} X^T y$$

Donde X es la matriz de características y y es el vector de etiquetas.

Bajo el supuesto que se entreno con solo un punto de datos (x_1, y_1), y que $x \sim \text{Uniform}(a, b)$ y $y = g(x) = 2x^2$, se tiene que:

$$x = x_1$$

Por lo que:

$$\begin{aligned} \hat{\theta} &= (X^T X)^{-1} X^T y \\ &= (x_1^2)^{-1} x_1 2x_1^2 \\ &= 2x_1 \end{aligned} \tag{1}$$

El coeficiente de regresión θ es igual a $2x_1$, cuando se entrena con un solo punto de datos (x_1, y_1).

1.3 c)

El sesgo de la predicción $\hat{h}(z_\star) = \hat{\theta} z_\star$ se puede calcular como:

$$\text{Bias}(h(z_\star)) = E[h(z_\star)] - g(z_\star)$$

Donde $g(z) = 2z^2$ es la función real que se intenta estimar.

$$\begin{aligned} \text{Bias}(h(z_\star)) &= E[h(z_\star)] - g(z) \\ &= E[\theta z_\star] - 2z_\star^2 \\ &= 2E[x z_\star] - 2z_\star^2 \\ &= 2E[x]E[z_\star] - 2z_\star^2 \\ &= -2z_\star^2 \end{aligned} \tag{2}$$

El sesgo de la predicción es igual a $-2z_\star^2$.

1.4 d)

La varianza de la predicción $\hat{h}(z_\star) = \hat{\theta} z_\star$ se puede calcular como:

$$\begin{aligned} \text{Var}(h(z_\star)) &= E[E[h(z_\star)] - h(z_\star)]^2 \\ &= E[-2x z_\star]^2 \\ &= 0 \end{aligned} \tag{3}$$

La varianza es cero ya que se entreno con un solo punto de datos, por lo que no hay variabilidad en la predicción.

1.5 e)

El error cuadrático medio (MSE) se puede calcular como:

$$\begin{aligned} \text{MSE} &= \text{Bias}^2 + \text{Var} + \sigma^2 \\ &= (-2z_\star^2)^2 + 0 + 0 \\ &= 4z_\star^4 \end{aligned} \tag{4}$$

Para $z = 1$, el MSE es igual a 4.

1.6 f)

Para mejorar el modelo se puede utilizar un modelo polinomial de grado 2, ya que la función real es cuadrática. Además, se puede utilizar un conjunto de datos más grande para entrenar el modelo y reducir el sesgo y la varianza.

1.7 g)

El fenómeno que se podría modelar con el enunciado es la relación entre la altura de un objeto y el tiempo que tarda en caer al suelo. La regresión lineal podría aportar al estudio del fenómeno al proporcionar una estimación de la relación entre las variables. Sin embargo, la regresión lineal puede no ser suficiente para modelar la relación entre la altura y el tiempo, ya que la relación es cuadrática. En este caso, un modelo polinomial de grado 2 sería más adecuado. Además, la baja capacidad computacional necesaria y la interpretación de los parámetros también podrían influir en la elección de un modelo simple como la regresión lineal para un problema complejo, ya que la regresión lineal es fácil de interpretar y de implementar.

2 Regresión lineal bayesiana

3 Regresión lineal caso aplicado

3.1 a)

Implementación de la clase `LinearRegression` en Python:

```
class LinearRegression:
  def __init__(self, features, labels):
  # Agregar columna de unos para el término de intercepto
      self.X = np.c_[np.ones((features.shape[0], 1)), features.to_numpy()]
      self.y = labels.to_numpy()
      self.theta = None
  def fit(self):
  # Calcular el coeficiente de regresión theta
      self.theta = np.linalg.inv(self.X.T.dot(self.X))\
                              .dot(self.X.T.dot(self.y))
  def predict(self, X_new):
  # Agregar columna de unos para el término de intercepto al valor a predecir
      X_new_b = np.c_[np.ones((X_new.shape[0], 1)), X_new]
      return X_new_b.dot(self.theta)
```

3.2 b)

La distribución de las calificaciones es la siguiente:

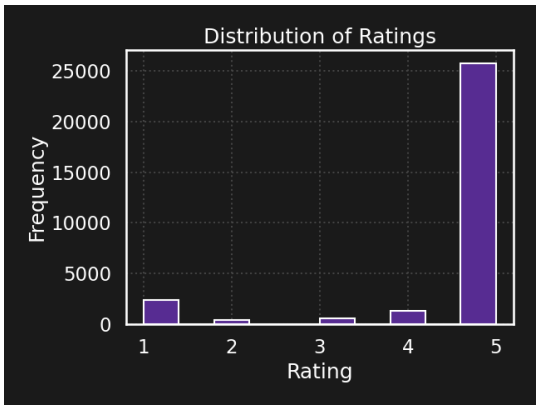

Figure 1: Distribución de las calificaciones de los usuarios

Se observa que la maoiría de las calificaciones se encuentran en 5, lo que indica que los usuarios tienden a calificar positivamente.

3.3 c)

Al entrenar el modelo de regresión lineal con las características **verified_purchase**, **helpful_votes** y **length_of_review** se obtienen los siguientes coeficientes:

Parameter	Coefficient
Intercepto	4.22
verified_purchase	0.6
helpful_votes	1.42e-04
length_of_review	-1.4e-02

Table 1: Coeficientes de regresión de la regresión lineal

Esto indica que las características **verified_purchase** y **helpful_votes** tienen un impacto positivo en la calificación, mientras que la característica **length_of_review** tiene un impacto negativo. En concreto, si la reseña es verificada, la calificación aumenta en 0.6 puntos. Por cada voto útil, la calificación aumenta en 1.42e-04 puntos. Por cada palabra adicional en la reseña, la calificación disminuye en 1.4e-02 puntos.

El error cuadrático medio (MSE) obtenido al entrenar el modelo y evaluarlo en el conjunto de prueba y entrenamiento:

Dataset	MSE
Train	1.16
Test	1.15

Table 2: Error cuadrático medio de la regresión lineal

3.4 d)

A continuación se muestra el MSE y la norma de los parametros para diferentes valores de ρ en regresión Lasso y Ridge:

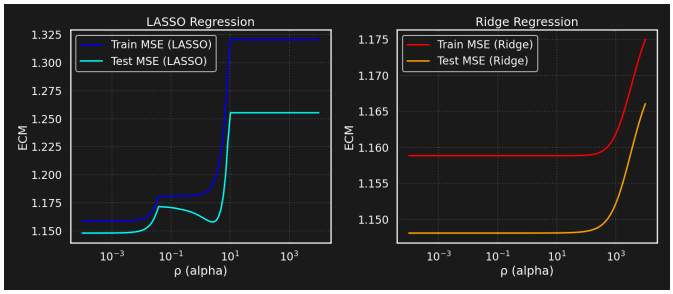

Figure 2: MSE para diferentes valores de ρ en regresión Lasso y Ridge

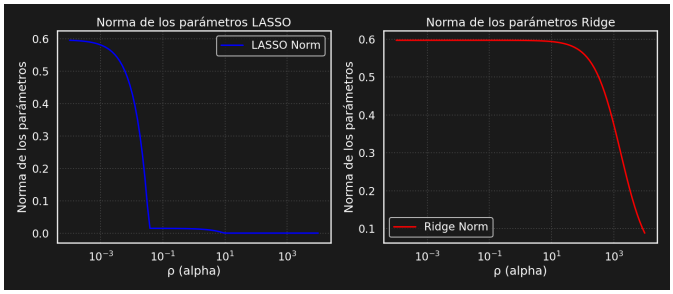

Figure 3: Norma de los parámetros para diferentes valores de ρ en regresión Lasso y Ridge

Para valores grandes de ρ, la regresión Lasso tiende a reducir los coeficientes a cero, mientras que la regresión Ridge tiende a reducir los coeficientes a valores pequeños.

3.5 e)

Ahora al esacalar las características antes de entrenar el modelo, se obtienen los siguientes resultados:

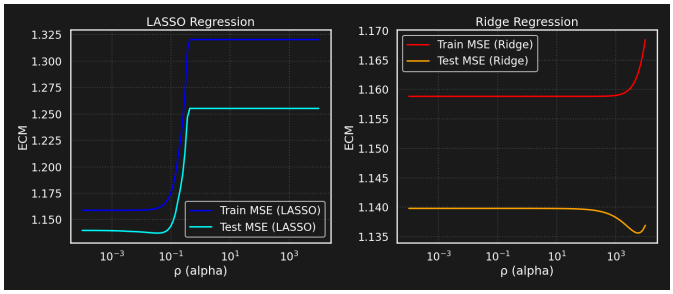

Figure 4: MSE para diferentes valores de ρ en regresión Lasso y Ridge

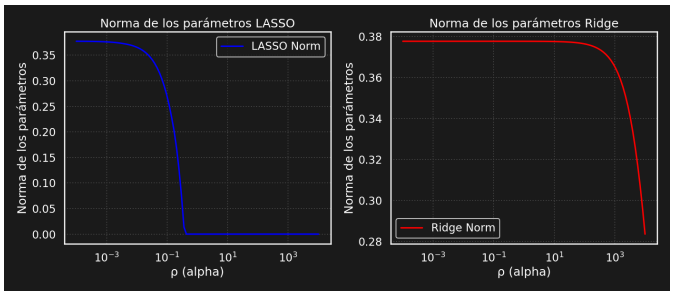

Figure 5: Norma de los parámetros para diferentes valores de ρ en regresión Lasso y Ridge

Las metricas calculadas son comparables, ya que los estimadores estan estimando caracteristicas en la misma escala. Además se puede observar una disminucion en la norma de los parametros producto de esto, luego se ven curvas mucho más suaves, incluso en el caso de ridge se ve una mejora en los valores de errores a mayor valor de ρ. Ridge generaliza mejor en terminos de error a observaciones fuera de muestra.

3.6 f)

Consideremos un problema de regresión lineal donde queremos incorporar características polinómicas. La formulación matemática de la regresión polinómica se puede expresar como:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_d x^d + \epsilon$$

Donde:

- y es la variable dependiente.
- x es la variable independiente.
- $\beta_0, \beta_1, \beta_2, ..., \beta_d$ son los coeficientes a estimar.

- d es el grado del polinomio.
- ϵ es el término de error.

Al implementar la regresión polinómica en Python, se puede utilizar la clase `PolynomialFeatures` de la librería `sklearn.preprocessing` para generar las características polinómicas. Obteniendo los siugientes resultados tanto en test como en train:

Dataset	MSE
Train	1.11
Test	1.142

Table 3: Error cuadrático medio de la regresión polinómica

3.7 g)

De todos los metodos, basadome en resultados, recomendaria una implementación de regresion Ridge con caracteristicas escaladas, ya que demostro dar unos resultados muy buenos en el conjunto de test, dando a entender que resulta un muy buen modelo para generalizar a observaciones fuera de muestra.