Prueba de evaluación de la práctica de regresión

Grado en Ingeniería en Tecnologías de Telecomunicación

Nombre y apellidos: _____

1. Se dispone de un conjunto de datos que ha sido dividido en 3 matrices `X_train, X_val, X_test` (para entrenamiento, validación y test, respectivamente), que deben normalizarse para obtener las matrices de datos siguientes:

   ```
   X_train_s : Conjunto de entrenamiento
   X_val_s : Conjunto de validación
   X_test_s : Conjunto de test
   ```

   ¿Cual es la forma correcta de normalizar?

   ◯
   ```
   from sklearn.preprocessing import StandardScaler
   scaler = StandardScaler()
   X_train_s = scaler.transform(X_train_val)
   X_val_s = scaler.transform(X_val)
   X_test_s = scaler.transform(X_test)
   ```

   ◯
   ```
   from sklearn.preprocessing import StandardScaler
   scaler = StandardScaler()
   X_train_s = scaler.fit_transform(X_train_val)
   X_val_s = scaler.fit_transform(X_val)
   X_test_s = scaler.fit_transform(X_test)
   ```

   √
   ```
   from sklearn.preprocessing import StandardScaler
   scaler = StandardScaler()
   X_train_s = scaler.fit_transform(X_train_val)
   X_val_s = scaler.transform(X_val)
   X_test_s = scaler.transform(X_test)
   ```

   ◯
   ```
   from sklearn.preprocessing import StandardScaler
   scaler = StandardScaler()
   X_train_s = scaler.fit(X_train_val)
   X_val_s = scaler.transform(X_val)
   X_test_s = scaler.transform(X_test)
   ```

2. Tras completar la normalización de la pregunta anterior, se calculan los valores siguientes:

```
m_train = np.mean(X_train_s, axis=0)
m_val = np.mean(X_val_s, axis=0)
std_train = np.std(X_train_s, axis=0)
std_test = np.mean(X_test_s, axis=0)
```

Indique la opción correcta

- √ `m_train[0] = 0`
- ◯ `m_val[0] = 0`
- ◯ `std_train[0] = 0`
- ◯ `std_test[0] = 1`

3. Se dispone de un conjunto de datos que ha sido dividido en 3 subconjuntos (entrenamiento, validación y test), guardados en tres arrays bidimensionales de numpy (`X_train`, `X_val` y `X_test`, respectivamente), donde cada fila representa una muestra.

Se desea ajustar un modelo de Ridge regression, eligiendo el valor del parámetro `alpha` por validación, entre los valores en

```
alpha_list = [0.1, 0.2, 0.4, 0.8, 1.6]
```

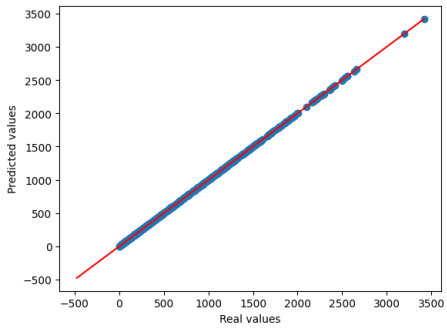
Recorriendo esta lista de valores, para cada valor de `alpha` se ha ajustado el modelo y se ha calculado el coeficiente R^2 con los datos de entrenamiento y con los de validación, obteniendo, respectivamente, las listas de valores

```
R2_train = [-1, 0.9, 0.3, 0.8, 0]
R2_val = [-0.9, 0.1, 0.8, 0.6, 0.5]
```

De acuerdo con los valores obtenidos, debe elegirse el valor

- ◯ `alpha = 0.1`
- ◯ `alpha = 0.2`
- √ `alpha = 0.4`
- ◯ `alpha = 0.8`

4. A la hora de representar las etiquetas de los datos de test frente a las predicciones hechas por el modelo, los puntos aparecen perfectamente alineados en la diagonal como indica la figura. ¿Qué podemos afirmar sobre las prestaciones del modelo?



◯ Tiene un R2 igual a 0
√ **Tiene un R2 igual a 1**
◯ Tiene un RMSE igual a 1
◯ Su R2 es inferior a su RMSE

5. Se dispone de un conjunto de datos que ha sido dividido en 3 subconjuntos (entrenamiento, validación y test), guardados en tres arrays bidimensionales de numpy (`X_train`, `X_val` y `X_test`, respectivamente) con sus respectivos arrays de las variables objetivo (`y_train`, `y_val` e `y_test`). Los datos ya están normalizados.

   Con estos datos, se ha aplicando un procedimiento estándar de validación para determinar los parámetros `alpha` y `gamma` de un modelo Kernel Ridge con kernel rbf, y se han obtenido los valores `alpha=0.3` y `gamma=1`.

   Se desea obtener el modelo final con estos parámetros, y estimar sus prestaciones mediante el RMSE. Para ello, se construye el objeto

```
kernel_ridge = KernelRidge(kernel='rbf', alpha=0.3, gamma=1)
```

   Indique cuál es la secuencia de instrucciones apropiada.

◯
```
kernel_ridge.fit(X_train, y_train)
y_pred = kernel_ridge.predict(X_val)
rmse_kernel = mean_squared_error(y_val, y_pred, squared=False)
```

√
```
kernel_ridge.fit(X_train, y_train)
y_pred = kernel_ridge.predict(X_test)
rmse_kernel = mean_squared_error(y_test, y_pred, squared=False)
```

◯
```
kernel_ridge.fit(X_val, y_val)
y_pred = kernel_ridge.predict(X_val)
rmse_kernel = mean_squared_error(y_test, y_pred, squared=False)
```

◯
```
kernel_ridge.fit(X_test, y_test)
y_pred = kernel_ridge.predict(X_test)
rmse_kernel = mean_squared_error(y_test, y_pred, squared=False)
```

6. Indique cual de las siguientes afirmaciones es verdadera.
   - ◯ Para un mismo dataset, el parámetro `alpha` es común a los modelos Lasso, Ridge, y KernelRidge: una vez obtenido el mejor valor para un modelo, podemos asegurar que es también el mejor valor para los otros dos.
   - ◯ Todos los modelos utilizados en la práctica (LinearRegression, Lasso, Ridge y KernelRidge) requieren optimizar un hiperparámetro: `alpha`, `gamma` o ambos.
   - ◯ El número de parámetros de un modelo de regresión semilineal no depende del grado del polinomio
   - √ **El modelo Kernel Ridge es no lineal respecto a las observaciones.**

7. Se va a utilizar un algoritmo de aprendizaje automático que tiene 3 parámetros para validar llamados alpha, beta y gamma. Para cada uno de estos parámetros se quieren validar los siguientes valores:

```
alpha = [1e-5, 1e-4, 0.001, 0.01, 0.1,  1, 10, 100, 200]
beta = [1, 2, 3, 4, 5]
gamma = [1, 10, 100, 1000]
```

   ¿Cuantos modelos se crearán para encontrar el mejor valor?
   - ◯ 18
   - ◯ 9
   - ◯ 900
   - √ **180**

   La técnica de analizar los pesos de un modelo para identificar qué variables son las más importantes, ¿en qué algoritmo no podría utilizarse?

8.
   - ◯ Regresor lineal
   - ◯ Regresor Ridge
   - ◯ Regresor Lasso
   - √ **Regresor KernelRidge**

9. ¿Cual de las siguientes afirmaciones sobre los resultados de la practica es FALSA?
   - √ **Al crear un modelo utilizando únicamente la mejor variable, se obtienen mejores prestaciones que al utilizar todas las variables.**
   - ◯ El modelo Kernel Ridge ha obtenido las mejores prestaciones.
   - √ **Los modelos polinómicos lineal y LASSO han obtenido mejores prestaciones que sus equivalentes no polinómicos.**
   - ◯ Ridge Regression y LASSO han obtenido prestaciones muy similares.

10. Indique qué modelo de regresión tiende a dar valor cero a los coeficientes asociados a las características de entrada menos relevantes:

    ◯ `LinearRegression`
    ◯ `Ridge`
    √ `Lasso`
    ◯ `KernelRidge`

11. Al utilizar la métrica R^2 indique qué implica el haber tenido un resultado:

    - Igual a 1
    - Igual a 0
    - Negativo

12. Se dispone de un conjunto de datos que ha sido dividido en 3 subconjuntos (entrenamiento, validación y test) y estandarizado correctamente. El nombre de los conjuntos de datos es el siguiente:

```
X_train_s, y_train : Conjunto de entrenamiento
X_val_s, y_val : Conjunto de validación
X_test_s, y_test : Conjunto de test
```

    Indique cual de estas 6 variables debe pasarse como parámetro en cada uno de los fragmentos que están subrayados en el siguiente código para entrenar correctamente un modelo de regresión Lasso, obteniendo el valor del parámetro alpha y mostrando finalmente el MSE del modelo.

```
from sklearn.linear_model import Ridge

v_alpha = [1e-5, 1e-4, 0.001, 0.01, 0.1, 1, 10, 100, 200]

r2_alpha = np.zeros(v_alpha.shape)

for ind_alpha in range(len(v_alpha)):

   ridgemodel = Ridge(alpha=v_alpha[ind_alpha])

   ridgemodel.fit(PARAMETRO 1, PARAMETRO 2)

   r2 = ridgemodel.score(PARAMETRO3, PARAMETRO 4)

   r2_alpha[ind_alpha] = r2

r2_min = np.min(r2_alpha)
pos=np.where(r2_alpha==r2_min)

alpha_best = v_alpha[pos[0][0]]

print('\nThe best value of alpha is: ',alpha_best )

ridge_best = Ridge(alpha=alpha_best)

ridge_best.fit(PARAMETRO 5, PARAMETRO 6)

y_pred_ridge = ridge_best.predict(PARAMETRO 7)
mse_ridge = round(mean_squared_error(PARAMETRO 8, y_pred_ridge),2)
print('The MSE for ridge regression model is:',mse_ridge)
```

13. Se desea ajustar un modelo de regresión $f(x) = w_0 + w_1x_1 + w_2x_2$ que tiene un hiperparámetro α, mediante un procedimiento de evalidación con un dataset que se ha dividido en tres subconjuntos, (de entrenamiento, validacion y test). Como medida de prestaciones se utiliza el coeficiente R^2.

    Indique qué consecuencia indeseada podría darse en cada uno de los casos siguientes:

    - Que el conjunto de entrenamiento sea demasiado pequeño.
    - Que el conjunto de validación sea demasiado pequeño.
    - Que el conjunto de test sea demasiado pequeño.

14. Para optimizar los hiperparámetros del modelo KernelRidge, ha sido necesario explorar todos los pares de valores (`alpha`, `beta`) mediante un doble bucle, y guardar el valor el mejor par en dos variables (digamos `alpha_best` y `beta_best`). Indique qué procedimiento ha seguido para seleccionar el mejor par.

    Para responder a esta pregunta, puede utilizar fragmentos de código, o pseudocódigo, o una simple descripción en lenguaje natural, del procedimiento seguido. Pero intente ser preciso en la explicación.

15. Explique por qué puede resultar interesante hacer un scatter plot de predicciones del modelo (en el eje vertical) respecto a los valores reales de la variable objetivo (en el eje horizontal). ¿Que información puede proporcionar esta representación? Puede apoyarse en ejemplos para su respuesta.