Análisis y reporte sobre el desempeño del modelo.

Cutberto Arizabalo Nava. A01411431.

Introducción.

En este documento se muestra el proceso de análisis de desempeño de un modelo de regresión lineal con la intención de tomar decisiones para la mejora del desempeño de este.

Nuestro set de datos está relacionado directamente con las mediciones de temperaturas tomadas durante la Segunda Guerra Mundial.

Específicamente hablando de variables, se definió que la Temperatura Máxima será nuestro valor para predecir. Asimismo, la Temperatura mínima y media serán las variables independientes con las cuales trataremos de predecir la temperatura máxima.

Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation).

Para el entrenamiento del modelo y su posterior validación, se dividió el set de datos de la siguiente forma:

70% - Datos para entrenamiento

30% - Datos para pruebas

```
# Se crea un conjunto para train y uno para test para cada set de datos.
# Se toma el 30% de los datos para train
# Random state nos permite pasar una semilla para la aleatoreidad
# en la separación
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=1)
```

Aunado a esto, se implementó la técnica de Cross-Validation para hacer una validación del comportamiento del modelo.

Desempeño del modelo

El MSE de train es 0.9221266984191295

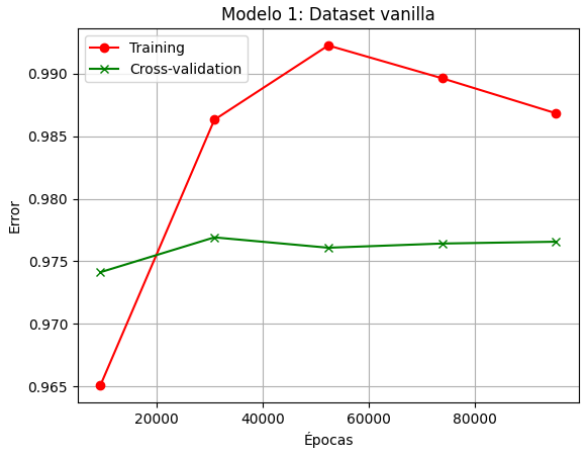
El MSE de test es 1.1099640074438633

En cuanto a la validación cruzada, tenemos el siguiente resultado:

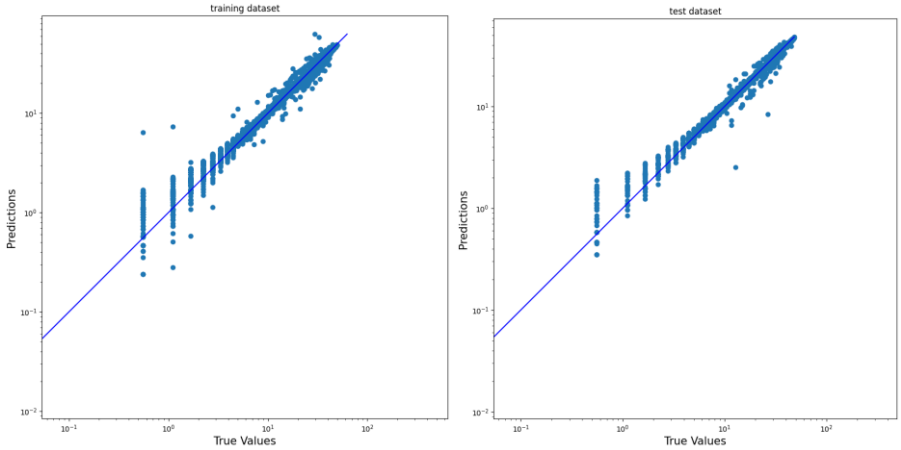
```
scoring = "neg_root_mean_squared_error"
linscores = cross_validate(model, X_train, y_train, scoring=scoring, return_estimator=True)
print("Linear regression score:", linscores["test_score"].mean())
```

✓ 0.9s

```
Linear regression score: -0.9546874893823597
```



En base a esta información, podemos comentar que no se observa la presencia de overfitting en este modelo, ya que el error es muy similar en los datasets de train y test. Asimismo, no se observa un underfitting, ya que el error de train no es elevado.

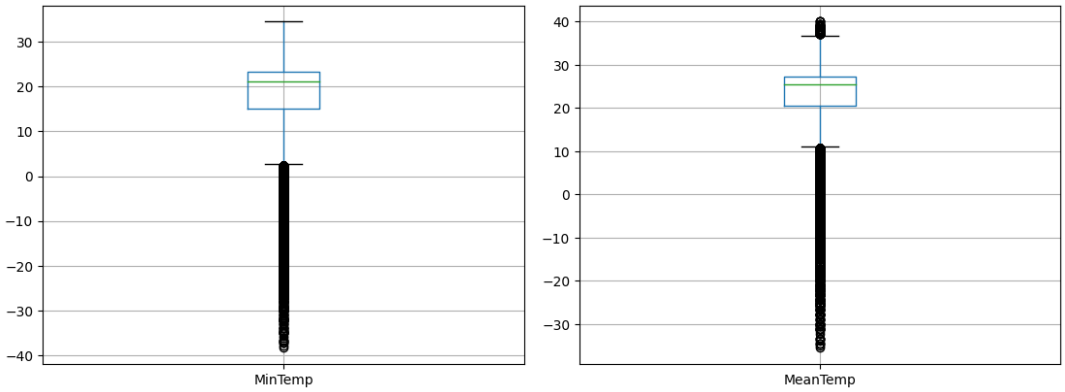


Algo interesante a observar es que la varianza es **baja**, ya que nuestra predicción se vio muy poco afectada por el cambio del dataset de train al de test. Sin embargo, es evidente que existe un grado de bias **medio**, ya que en el dataset de test (lado derecho), notamos que la predicción ya no cruza por el centro de los datos que coinciden en la misma línea, cosa que sí sucedía en el dataset de train.

Podemos ver que tenemos en todos los casos un error muy cercano a 1.

Vamos a intentar reducir este error en el modelo utilizando algunas técnicas de regularización de datos y ajustando algunos hiper parámetros.

Para empezar, vamos a analizar los datos de entrada.



La presencia de valores extremos es innegable. Basándonos en esto, vamos a aplicar una winsorización a los datos, lo que nos permitirá deshacernos de una parte de estos valores extremos, asignándole a estos un valor menos extremo. Dicha winsorización se hará al 5% de los outliers en cada extremo.

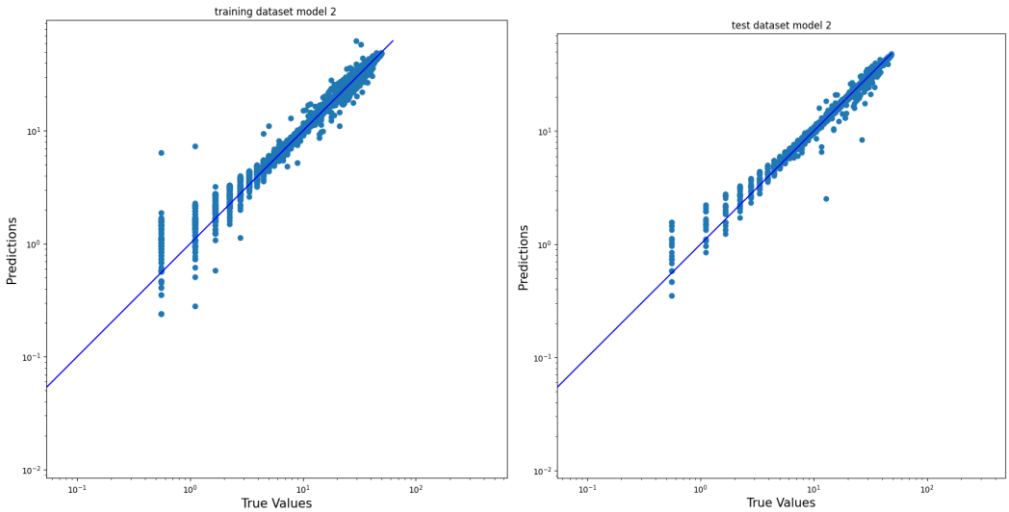
Asimismo, ajustaremos el tamaño del Split para que se use el 80% de los datos para entrenar y el 20% para probar.

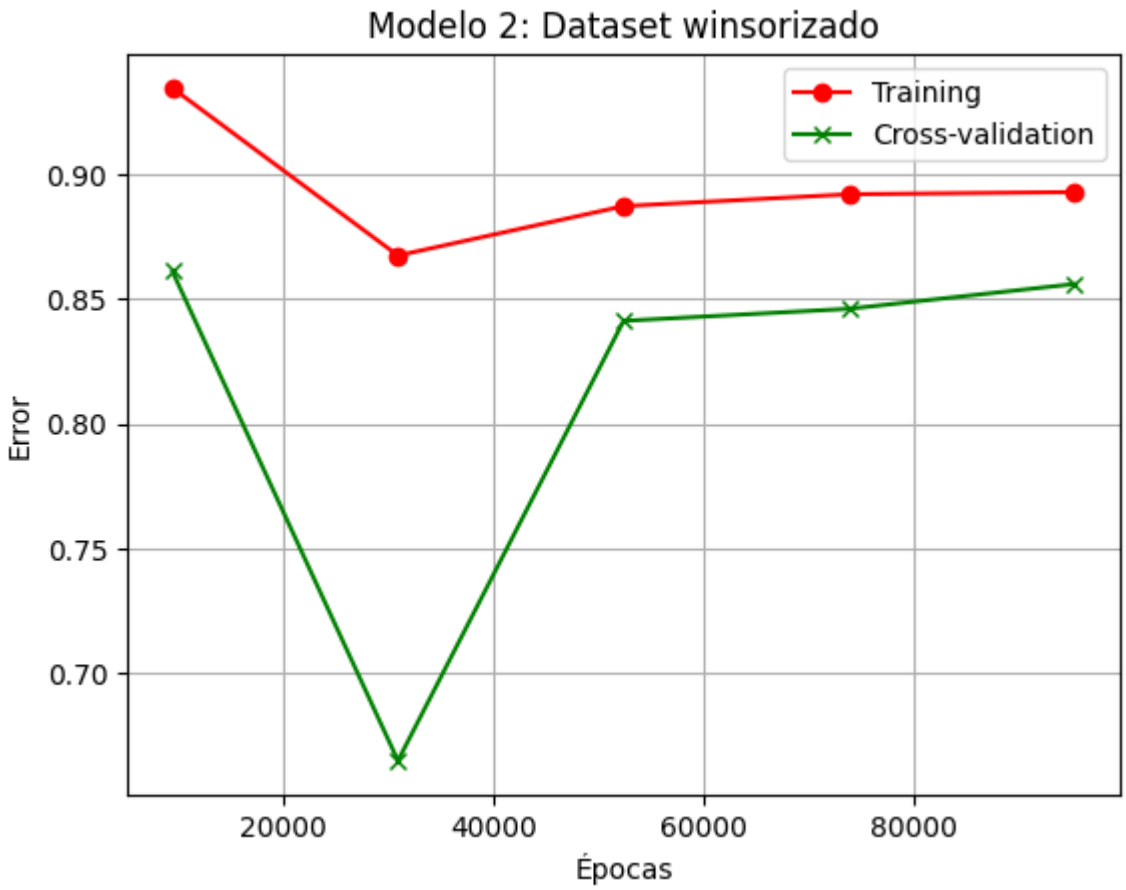
Resultados del nuevo modelo:

El MSE de train es 0.9637526721514528

El MSE de test es 1.0373787670269368

Podemos notar que el error en train es ligeramente mayor, pero ahora el error en test es menor. De esta forma, la diferencia entre ambos errores se ha reducido, y hemos obtenido mejores resultados en el dataset de pruebas.





En este modelo la diferencia que se obtiene al final del entrenamiento entre el error en training y el error en cross validation es menor, y se puede apreciar claramente al comparar esta gráfica con la del modelo anterior.

Además, en las gráficas de comparación entre las predicciones y los datos reales para train y test, podemos notar que el bias se redujo ligeramente en el dataset de test, ya que la línea de predicción pasa un poco más cercano al centro que en el modelo anterior. Sin embargo, seguimos contando con un nivel **medio** de bias y un nivel bajo de varianza.

Conclusiones:

Podemos ver que nuestro modelo tuvo una ligera mejora gracias a la regularización de los datos y al ajuste en un hiper parámetro. Sin embargo, seguimos observando un grado de bias. Para solucionar esto existen varias opciones, siendo la más drástica cambiar nuestro modelo por uno que se adapte mejor al set de datos.

En lo particular me resulta satisfactorio el rendimiento del modelo, ya que cuenta con un error bastante bajo y fue entrenado de forma rápida y con pocos recursos computacionales.