**Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo. (Portafolio Análisis)**

Miguel Ángel Pérez López A01750145

**Regresión logística**

La regresión logística es un tipo de análisis de regresión para predecir una variable y. Sirve para estudiar las relaciones entre el conjunto x y la y. Necesita una o más entradas de variables independientes, una variable dependiente y una función de activación (por ejemplo la sigmoide). Sólo funciona con variables numéricas.

**Dataset**

Se utilizó el dataset winequality\_red que tiene 11 variables numéricas independientes de tipo flotante y una variable y binaria que define si el vino es de clase es de buena calidad o no.

El dataset no necesitó imputación de datos o cambio de variables categóricas a numéricas.

Escalamos todas las variables independientes X.

**Separación de datos**

Para el modelo de este dataset separamos los datos en train, test y validation de la siguiente forma:

Test size: 20%

Del train size se destinó el 25% para el validation set. Por lo tanto:

Train size: 60%

Validation size: 20%

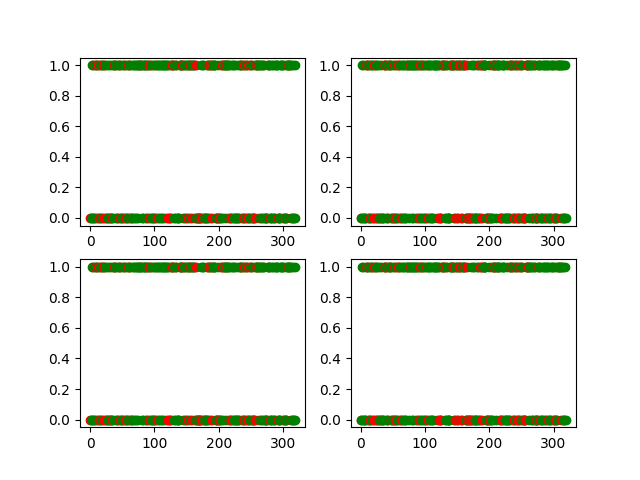
**Procedimiento**

Crearemos una lista con épocas y learning rates aleatorios. Vamos a entrenar el modelo y usaremos el conjunto de validación varias veces y las époccas y learning rates de la lista antes mencionada. Con esos hiper parámetros calcularemos la precisión de cada corrida para guardar los hiper parámetros que den la mejor precisión.

De esta forma tenemos hiper parámetros que podrían dar una buena precisión con el conjunto de test.

Para mejorar el desempeño del modelo utilizaremos una técnica de regularización llamada coarse to fine. Vamos a probar el modelo con un rango menor y mayor alrededor de los hiper parámetros obtenidos. Esto nos da una gran probabilidad de encontrar mejores hiper parámetros que resulten en una mejor precisión.

Entrenamiento con conjunto de validación

Gráfico, Forma

Descripción generada automáticamente

Predicciones correctas, Predicciones incorrectas

Como podemos observar en las 4 corridas, hay un gran porcentaje de puntos verdes (predicciones correctas).

El grado de bias o sesgo es bajo casi medio porque hay varios puntos verdes pero en el eje x en medio hay muchos puntos rojos. En general la mayoría son predicciones correctas alcanzando más del 70% de accuracy. La gráfica de costo nos indica que el costo no es bajo y que no va a llegar a 0. Por lo tanto el bias es un relativamente medio.

El grado de varianza es bajo porque las 4 corridas se ven muy similares. Esto nos indica que hay poca dispersión en los resultados y que las predicciones no cambian drásticamente dependiendo la corrida. La gráfica de costo muestra que hay poca dispersión porque el costo va disminuyendo y no muestra “picos”.

Predicción con conjunto de test

Interfaz de usuario gráfica

Descripción generada automáticamente

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

(Precisión, apha, épocas)

Estas gráficas son de las predicciones con valores reales y tenemos resultados muy similares a los del training e incluso mejores.

El grado de varianza es bajo. Esto nos indica que hay poca dispersión en los resultados. El conjunto de test es más grande y no parece que los errores hayan incrementaron proporcionalmente a los datos.

El nivel de ajuste del modelo final es **fit**. Definitivamente no es overfitting porque el error en el test no incrementó mucho. Debido a que el learning rate se calculó aleatorio con un rango entre 0 y 1, el learning rate fue alto y el costo no se pudo reducir más. Al tener un costo mayor a 0.5, tiene sentido que la precisión no sea mayor a 75% (como se ve en el screen de arriba).

En conclusión, el método de regresión lineal fue