## Correcciones Implementación sin framework

Elisa María Bonilla Martín A01028576

Instituto Tecnológico de Estudios Superiores de Monterrey Campus Estado de México

## 1 Correcciones

60 ptos. 30 ptos. 0 ptos. Algoritmo Es claro que el Utiliza más del No cumple o implementado algoritmo es 5% del código no entregado manualmente y implementado con librerías corriendo de al menos el ya manera directa 95% implementada desde el manualmente compilador.

En mi primera entrega utilizaba pandas, numpy y sklearn. Específicamente utilizaba las siguientes funciones

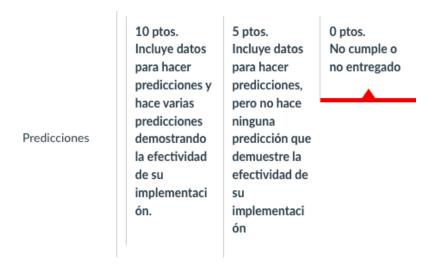
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Para reducir mi uso de las librerías utilice las siguientes líneas de código para hacer el train test split:

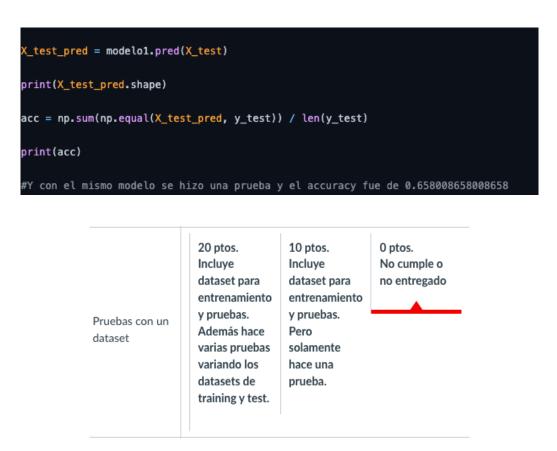
```
n_train = math.floor(s_f * feat.shape[0])
n_test = math.ceil((1-s_f) * feat.shape[0])
X_train = feat[:n_train]
y_train = target[:n_train]
X_test = feat[n_train:]
y_test = target[n_train:]
```

Y para el accuracy score hice lo siguiente:

```
acc = np.sum(np.equal(X_train_pred, y_train)) / len(y_train)
print(acc)
```



En la primera entrega no había realizado predicciones por lo que añadí la siguiente parte:



En la primera entrega no había proporcionado el dataset que estaba usando pero ya lo añadí al repositorio de github y aquí está la prueba.



Análisis del nivel de accuracy y/o error (loss)

10 a > 0 pts
Realiza un breve documento en el que analiza el accuracy y/o error de su implementación variando algunos hiperparámetros

Y por último realice lo siguiente para encontrar los mejores parámetros

```
#Se crea una función que prueba los diferentes learning rates
learning_rates = [0.1, 0.15, 0.5, 0.04]
models = \{\}
for i in learning_rates:
   print ("learning rate es: ",i)
   models[i+1] = RegLogistica(learning_rate = i, num_i=1000)
   models[i+1].fit(X_train, y_train)
   X_train_pred = models[i+1].pred(X_train)
   acc = np.sum(np.equal(X_train_pred, y_train)) / len(y_train)
   print("El accuracy es: ",acc)
   print ("-----
'''Se obtuvo que para
learning rate es: 0.1
El accuracy es: 0.6685288640595903
learning rate es: 0.15
El accuracy es: 0.6741154562383612
learning rate es: 0.5
El accuracy es: 0.6536312849162011
learning rate es: 0.04
El accuracy es: 0.6480446927374302
#Podemos ver que para este problema no se debe de usar un learning rate tan bajo
```