

Regularización de Red Neuronal

Ejemplo

Importar Librerías

```
In [72]: import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model
from sklearn.linear_model import RidgeClassifier
from sklearn.linear_model import SGDClassifier
import warnings
warnings.filterwarnings('ignore')
```

Cargar Datos

```
In [42]: data = pd.read_csv('./water_potability.csv')
data = data.replace(np.nan, "0")

data.sample(10)
```

```
Out[42]:
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	T
2028	7.744125	203.396227	21403.002048	7.623242	333.210903	424.328740	16.949811	
722	8.860451	193.743105	24085.112140	6.183110	0	526.704924	9.901328	
1162	11.244507	169.403655	43533.657845	5.888460	341.0296	336.446655	16.089710	
444	7.70441	234.635951	22057.379165	8.124403	0	397.399660	14.897563	
816	5.452215	199.228468	14446.812390	7.689191	298.318418	469.131951	14.020507	
2483	8.759489	219.921390	33364.918123	5.687072	324.038506	443.548006	17.763343	
2941	4.27516	229.443109	26098.637956	6.525203	281.896827	508.792098	16.133203	
551	0	190.046661	18327.438429	7.898765	354.419249	445.928456	12.902213	
987	4.771491	200.374076	48002.084596	9.165239	340.310176	310.927835	18.768328	
2276	7.094802	222.501306	30045.367741	6.664010	0	531.959462	16.162941	

Definir Input/Output de la Red Neuronal

```
In [43]: #Entradas a La red
X=data.drop(columns="Potability")
#Salida
y = data.Potability
```

Escalamiento

```
In [45]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Entrenar Red Neuronal con Ridge

La Regresión Rigde, también denominada regresión contraída o Tikhonov regularization, regulariza el modelo resultante imponiendo una penalización al tamaño de los coeficientes de la relación lineal entre las características predictivas y la variable objetivo. En este caso, los coeficientes calculados minimizan la suma de los cuadrados de los residuos penalizada al añadir el cuadrado de la norma L2 del vector formado por los coeficientes donde λ es un parámetro que controla el grado de penalización: cuanto mayor éste, los coeficientes serán menores resultando más robustos a la colinealidad

```
In [70]: modelo = RidgeClassifier(alpha=0.1, max_iter=500, solver='lbfgs', tol=0.00000001, random
modelo.fit(X_train, y_train)
predictions=modelo.predict(X_test)
```

Revisamos la Red Neuronal con Ridge

```
In [71]: accuracy= modelo.score(X_test, y_test)
print('Accuracy: ', accuracy)
```

Accuracy: 0.6074481074481074

Entrenar Red Neuronal con Laso

Es un modelo lineal que penaliza el vector de coeficientes añadiendo su norma L1 (basada en la distancia Manhattan) a la función de coste.

Lasso tiende a generar "coeficientes dispersos": vectores de coeficientes en los que la mayoría de ellos toman el valor cero. Esto quiere decir que el modelo va a ignorar algunas de las características predictivas, lo que puede ser considerado un tipo de selección automática de características. El incluir menos características supone un modelo más sencillo de interpretar que puede poner de manifiesto las características más importantes del conjunto de datos. En el

caso de que exista cierta correlación entre las características predictivas, Lasso tenderá a escoger una de ellas al azar.

Esto significa que, aunque Ridge es una buena opción por defecto, si sospechamos que la distribución de los datos viene determinada por un subconjunto de las características predictivas, Lasso podría devolver mejores resultados.

```
In [67]: modelo = linear_model.Lasso(alpha=0.1)
modelo.fit(X_train,y_train)
predictions= modelo.predict(X_test)
```

Revisamos la Red Neuronal con Lasso

```
In [68]: accuracy= modelo.score(X_test, y_test)
print('Accuracy: ',accuracy)
```

Accuracy: -5.629996622036337e-05

Entrenar Red Neuronal con Elastic Net

Es un modelo de regresión lineal que normaliza el vector de coeficientes con las normas L1 y L2. Esto permite generar un modelo en el que solo algunos de los coeficientes sean no nulos, manteniendo las propiedades de regularización de Ridge.

El parámetro λ regula el peso dado a la regularización impuesta por Ridge y por Lasso. Desde este punto de vista Elastic Net es un superconjunto de ambos modelos.

En el caso de que exista cierta colinealidad entre varias características predictivas, Elastic Net tenderá a escoger una o todas (aun con coeficientes menores) en función de cómo haya sido parametrizado.

```
In [76]: modelo = SGDClassifier(alpha=0.1,max_iter=500,tol=0.00000001,loss='squared_error',per
modelo.fit(X_train,y_train)
predictions= modelo.predict(X_test)
```

Revisamos la Red Neuronal con Elastic Net

```
In [77]: accuracy= modelo.score(X_test, y_test)
print('Accuracy: ',accuracy)
```

Accuracy: 0.608058608058608

Conclusion

Para este conjunto de datos a pesar de no dar resultados mejores al 60% los algoritmos de regularización L2 (Ridge) y Elastic Net son los indicados superando el resultado de Lasso.

