

Benemérita Universidad Autónoma de Puebla - FCC

Minería de Datos

# Avance III.

## Implementación del Ensamble



SEMANA DEL 20 - 24 DE ABRIL 2020

Emiliano Carrillo Moncayo

[Enlace al Jupyter Notebook de esta entrega](#)

# Introducción y Objetivos

Esta vez mi objetivo era comenzar a experimentar con la implementación del ensamble y la agrupación de mis tres modelos de clasificación: K Nearest Neighbors, Red Neuronal, y Naive Bayes.

Decidí ocupar como ensamble un método de máximo de votaciones. Quiere decir que tomo la moda de los resultados de cada data-point para cada uno de mis modelos. Con esto saco mi predicción final, la predicción del ensamble.

## Experimentos y Desarrollo

Los resultados que había obtenido con cada uno de mis clasificadores por separados los siguientes:

\*\*\*\*\* CLASIFICADOR KNN \*\*\*\*\*

Matriz de confusión:

```
[[11  1  0  0  0  0]
 [ 0 40  0  0  0  1]
 [ 0  1  9  0  2  0]
 [ 0  0  0  4  0  0]
 [ 0  0  0  0  6  2]
 [ 0  0  0  0  0 20]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	12
1	0.95	0.98	0.96	41
2	1.00	0.75	0.86	12
3	1.00	1.00	1.00	4
4	0.75	0.75	0.75	8
5	0.87	1.00	0.93	20
accuracy			0.93	97
macro avg	0.93	0.90	0.91	97
weighted avg	0.93	0.93	0.93	97

Puntaje de precisión:

0.9278350515463918

\*\*\*\*\* CLASIFICADOR NEURAL NET\*\*\*\*\*

Matriz de confusión:

```
[[11  1  0  0  0  0]
 [ 0 41  0  0  0  0]
 [ 0  0 12  0  0  0]
 [ 0  0  0  4  0  0]
 [ 0  0  0  0  8  0]
 [ 0  0  0  0  0 20]]
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	12
1	0.98	1.00	0.99	41
2	1.00	1.00	1.00	12
3	1.00	1.00	1.00	4
4	1.00	1.00	1.00	8
5	1.00	1.00	1.00	20

accuracy			0.99	97
macro avg	1.00	0.99	0.99	97
weighted avg	0.99	0.99	0.99	97

Puntaje de precisión:

0.9896907216494846

RESULTADOS OBTENIDOS CON LOS MODELOS KNN Y LA RED NEURONAL

\*\*\*\*\* CLASIFICADOR NAIVE BAYES \*\*\*\*\*

Matriz de confusión:  
[[ 9 1 0 0 0 2]  
[ 0 40 0 0 0 1]  
[ 2 2 8 0 0 0]  
[ 0 0 0 4 0 0]  
[ 0 1 0 0 5 2]  
[ 0 0 0 0 0 20]]

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.82	0.75	0.78	12
1	0.91	0.98	0.94	41
2	1.00	0.67	0.80	12
3	1.00	1.00	1.00	4
4	1.00	0.62	0.77	8
5	0.80	1.00	0.89	20
accuracy			0.89	97
macro avg	0.92	0.84	0.86	97
weighted avg	0.90	0.89	0.88	97

Puntaje de precisión:  
0.8865979381443299

## RESULTADOS OBTENIDOS CON NAIVE BASES

Después de haber ensamblado éstos, obtuve los resultados totales de mi ensamble:

\*\*\*\*\* ENSAMBLE \*\*\*\*\*

Matriz de confusión:  
[[11 1 0 0 0 0]  
[ 0 41 0 0 0 0]  
[ 2 1 9 0 0 0]  
[ 0 0 0 4 0 0]  
[ 0 0 0 0 6 2]  
[ 0 0 0 0 0 20]]

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.85	0.92	0.88	12
1	0.95	1.00	0.98	41
2	1.00	0.75	0.86	12
3	1.00	1.00	1.00	4
4	1.00	0.75	0.86	8
5	0.91	1.00	0.95	20
accuracy			0.94	97
macro avg	0.95	0.90	0.92	97
weighted avg	0.94	0.94	0.94	97

Puntaje de precisión:  
0.9381443298969072

## RESULTADOS OBTENIDOS INTEGRANDO EL ENSAMBLE TOTAL

Como podemos observar la red neuronal operó mejor que el ensamble final. Deduzco que ocurre esto porque fue a la red neuronal a la única que le hice un grid search para obtener la mejor configuración de sus parámetros de entrada.

Para la siguiente entrega planeo que mi código de la opción al usuario de buscar automáticamente la mejor configuración de parámetros para cada uno de los modelos automáticamente o éste los pueda ingresar a su disposición.