CLASIFICADOR DE COMENTARIOS TÓXICOS

Santiago Ariza Briceño
Yezith Fernando Rincón Guevara
Nicolas Tarazona Moncada

PROBLEMA

- Hoy en día, cualquier vídeo alojado en la plataforma Youtube, cuenta con personas que suelen dejar comentarios negativos de distintas índoles que realmente no aportan ningún tipo de valor adicional o constructivo, es decir, no aportan nada bueno y la mayoría solo buscan generar odio/polémica.

SOLUCIÓN

- Es por eso que decidimos realizar este clasificador de comentarios, en donde se busca separar estos comentarios tóxicos para llevar a cabo alguna acción hacia estos, por ejemplo, que sean eliminados por completo.
- Para desarrollar el modelo de clasificación se usarán varias de las técnicas de clasificación vistas durante el semestre.

Decision Tree Classifier

Random Forest Classifier

Support Vector Machine

Naive Gaussian Bayes

Redes Neuronales
DNN

DATASET

	CommentId	VideoId	Text	IsToxic	IsAbusive	IsThreat	IsProvocative	IsObscene	IsHatespeech	IsRacist	IsNationalist	IsSexist
0	Ugg2KwwX0V8-aXgCoAEC	04kJtp6pVXI	If only people would just take a step back and...	False	False	False	False	False	False	False	False	False
1	Ugg2s5AzSPioEXgCoAEC	04kJtp6pVXI	Law enforcement is not trained to shoot to app...	True	True	False	False	False	False	False	False	False
2	Ugg3dWTOxryFfHgCoAEC	04kJtp6pVXI	\nDont you reckon them 'black lives matter' ba...	True	True	False	False	True	False	False	False	False
3	Ugg7Gd006w1MPngCoAEC	04kJtp6pVXI	There are a very large number of people who do...	False	False	False	False	False	False	False	False	False
4	Ugg8FfTbbNF8IngCoAEC	04kJtp6pVXI	The Arab dude is absolutely right, he should h...	False	False	False	False	False	False	False	False	False

El dataset que se usó contaba con una columna ‘Text’ que contenía el comentario extraído tal cual de youtube y con distintas columnas en donde el comentario ya venía clasificado según su tipo de toxicidad. Para el desarrollo del proyecto trabajaremos con la columna ‘Text’ y la columna ‘IsToxic’.

PRE-PROCESAMIENTO

01

Se codifican los valores 'True' y 'False' del dataset por '1' y '0' respectivamente

	Text	IsToxic	IsAbusive	IsThreat
0	If only people would just take a step back and...	0	0	0
1	Law enforcement is not trained to shoot to app...	1	1	0
2	\nDont you reckon them 'black lives matter' ba...	1	1	0
3	ere are a very large number of people who do...	0	0	0
4	The Arab dude is absolutely right, he should h...	0	0	0

02

Se hace una limpieza de signos en los comentarios

```
'if', 'only', 'people', 'would', 'just', 'take', 'a',
```

03

Se mapean las palabras de los comentarios a números

```
'if': 4934, 'only': 8698, 'people': 6996, 'would': 7986,
```

PRE-PROCESAMIENTO

04

Se crea un nuevo DataFrame con las palabras ya mapeadas a números y se elige lo que será nuestro ground-truth, la columna 'IsToxic'.

	0	1	2	3	4	5	6	7	8	9	...	810	811	812	813	814	815	816	817	818	819
0	4934	8698	6996	7986	4844	4080	6213	1868	6157	6119	...	0	0	0	0	0	0	0	0	0	0
1	2094	7596	8793	8828	5280	1132	1985	1132	8993	5925	...	0	0	0	0	0	0	0	0	0	0
2	1205	3940	8641	2921	3262	4083	5372	3232	4826	2375	...	0	0	0	0	0	0	0	0	0	0
3	6915	3719	6213	8820	5001	9198	8598	6996	2349	4060	...	0	0	0	0	0	0	0	0	0	0
4	1221	3390	5479	8793	7728	6205	3491	3368	2292	8828	...	0	0	0	0	0	0	0	0	0	0

IMPLEMENTANDO LOS MODELOS DE CLASIFICACIÓN

Naive Gaussian Bayes

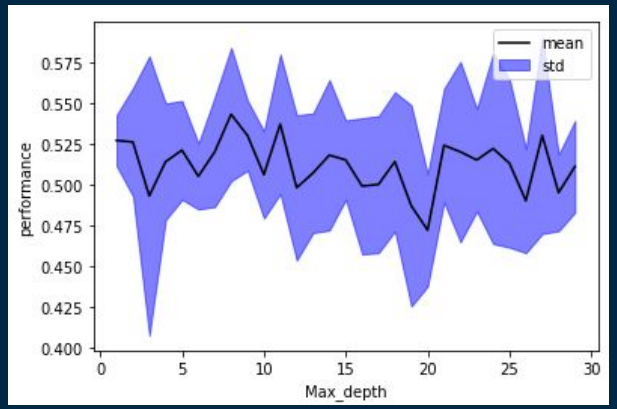
- Métrica de accuracy en: 0.50%

Decision Tree Classifier

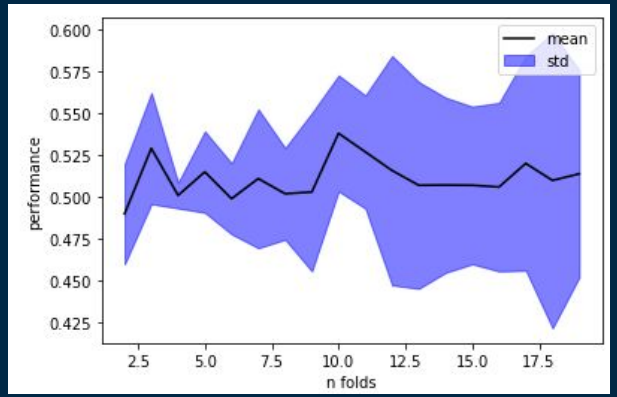
0.545% (DTC)

Métrica de accuracy en: 0.515%
(+/- 0.04019)

Decision Tree Classifier - tunning (max_depth)



Decision Tree Classifier - tunning (Cross-Validation)

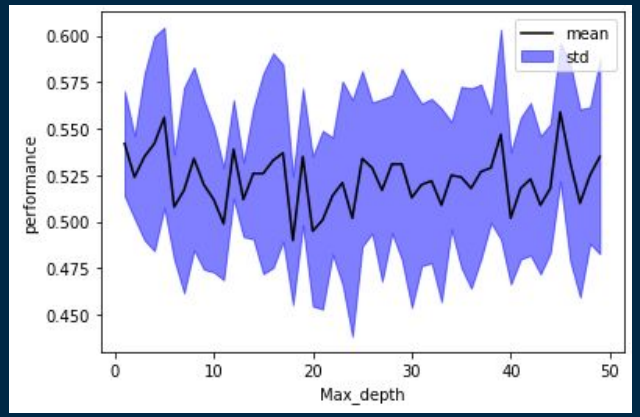


IMPLEMENTANDO LOS MODELOS DE CLASIFICACIÓN

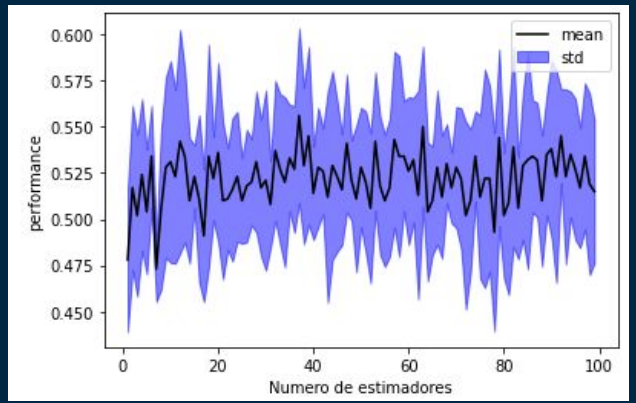
Random Forest Classifier

- Métrica de accuracy en: 0.516% (+/- 0.042)

Random Forest Classifier - tunning (max_depth)



Random Forest Classifier - tunning (n_estimators)



IMPLEMENTANDO LOS MODELOS DE CLASIFICACIÓN

Support Vector Machine (rbf)

- Métrica de accuracy en: 0.55%

Red neuronal - DNN

- Métrica de accuracy en: 0.52%

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation=tf.nn.tanh),
    tf.keras.layers.Dense(128, activation=tf.nn.tanh),
    tf.keras.layers.Dense(64, activation=tf.nn.tanh),
    tf.keras.layers.Dense(2, activation=tf.nn.softmax)
])

model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(X_train, y_train, epochs=40)
```

IMPLEMENTANDO LOS MODELOS DE CLASIFICACIÓN

Análisis de componente principal (PCA)

- Se redujo la dimensionalidad de 820 a 500 y con los datos transformados se obtuvo:

Random Forest Classifier

- Métrica de accuracy en: 0.543 (+/- 0.042)

Red neuronal - DNN

- Métrica de accuracy en: 0.5566666722297668%

CONCLUSIONES