**Sistemas Inteligentes TC2011.1**

Semestre Febrero- Junio 2021

# Proyecto Final: Agentes basados en conocimiento

---


**Profesor Tutor:**
Candy Yuridiana Alemán Muñoz


Fecha de entrega: martes 08 de junio del año 2021
Presentación: miércoles 09 de mayo - 4:00 PM


**Integrantes:**
Juan Diego Bastidas Santivañez - A01423502

Kimberly Atara Lopez Vazquez - A01423052

Equipo 2

# Reporte

## Introducción

El propósito de este reporte es exponer los resultados que obtuvimos de implementar algoritmos para el entrenamiento de modelos y así poder resolver un problema clasificando el valor de una instancia objetivo.

Problema: La xenofobia ha sido un comportamiento social presente en las personas desde los inicios de la humanidad. El miedo y el rechazo a lo diferente han llevado a muchas personas a desconfiar, menospreciar e incluso odiar a otras personas que pertenecen a un entorno social diferente. Este comportamiento xenófobo es tan fuerte que ha logrado incorporarse a internet a través de las distintas redes sociales. Por lo tanto, este proyecto trata sobre el análisis de clasificadores capaces de detectar Tweets xenófobos utilizando el texto de algunos tweets.

También decidimos usar tres algoritmos supervisados para analizar su eficiencia en cada experimento con los diferentes conjuntos de características obtenidas gracias a el preprocesamiento de los textos.

## Descripción del conjunto de datos

Nosotros decidimos utilizar el conjunto de datos de tweets para la detección de xenofobia o no xenofobia, misma que es su clase. El conjunto de textos de entrada que se encuentran en esta base de datos es en el idioma inglés. Contábamos con dos archivos .csv uno de Training y otro de Testing. El archivo de Testing no contenía tweets con clase, por lo que, decidimos utilizar 1000 tweets de Training para general el modelo y 500 para clasificar. Obtenemos la matriz de confusión, porcentaje de instancias bien colocadas y mal colocadas, matriz de recuerdo y matriz de precisión.

Cada tweet, por medio de la clase, define si contiene un sentimiento xenófobo (1) o no (0).

Posteriormente, analizando los resultados obtenidos asignamos una clase a los tweets del archivo de “Testing”.

## Preprocesamiento

Una razón adicional de ser un requerimiento en el proyecto de utilizar pre procesamientos, es el hecho de que ayuda que los algoritmos puedan analizar los textos de forma más fácil. Nos ayuda a limpiar los textos de símbolos extraños o diferentes al idioma, a quitar palabras vacías, a tener los textos normalizados a un formato específico y que son fáciles de procesar en los algoritmos inteligentes.

En el preprocesamiento de los textos nosotros nos ayudamos de la librería de nltk y re (regex).

En nuestro caso decidimos utilizar el vocabulario de Training & Testing como nuestro conjunto de características. Con el objetivo de obtener las características más relevantes y minimizar el número de las mismas,iniciamos con un regex que nos ayudó a quitar todos los hashtags #, los correos y menciones en cada tweet @. Segundo, quitamos todas las contracciones, pasamos el texto a minúsculas y removimos signos de puntuación. Tercero, quitamos las palabras con el menor número de frecuencias simples (número de apariciones) en los textos, ya que estas palabras podrían no ayudarnos mucho porque pueden ser inventadas, palabras irrelevantes o quizás escribieron alguna palabra mal escrita en un tweet. Delimitamos a 10 el número mínimo de frecuencia por palabra en los diccionarios.

```
Vocabulario Lematizados

['trump', 'supporters', 'need', 'to', 'say', 'the', '4', 'democrats', 'socialist', 'squad', 'be', 'racist', 'america', 'have', 'freedom
548
Vocabulario Stop words

['trump', 'supporters', 'say', '4', 'democrats', 'socialist', 'squad', 'racist', 'america', 'freedom', 'speech', 'rally', 'democratic',
442
Vocabulario Steeming

['trump', 'support', 'need', 'to', 'say', 'the', '4', 'democrat', 'socialist', 'squad', 'are', 'racist', 'america', 'ha', 'freedom', 'o
570
Numero de instancias (Training): 1500
Numero de instancias (Testing): 500
```

Luego, realizamos los diferentes tipos de pre-procesamientos seleccionados, los cuales son:

**Lematización**

Este preprocesamiento recoge todos los textos, y pasa las palabras de cada oración a su forma de “diccionario”. Esto hace que muchas palabras se pasen a una forma que pueda resumir todos los posibles cambios en la palabra (por ejemplo, “decir” reemplazaría las palabras “dije, dijéramos, dijo, decimos, decirnos”, etc).

```
def lemmatize_word(text):
    word_tokens = word_tokenize(text)
    lemmas = [lemmatizer.lemmatize(word, pos ='v') for word in word_tokens]
    expanded_text = ' '.join(lemmas)
    return expanded_text
```

**Eliminación de Stopwords**

Estas palabras son palabras vacías, palabras que si se ponen solas carecen de algún significado. Usualmente son conjunciones, artículos, preposiciones o adverbios.

```
def remove_stopwords(text):
    stop_words = set(stopwords.words("english"))
    word_tokens = word_tokenize(text)
    filtered_text = [word for word in word_tokens if word not in stop_words]
    expanded_text = ' '.join(filtered_text)
    return expanded_text
```

**Stemming**

Este tipo de preprocesamiento es una forma de quitar sufijos de las palabras, e intentar llevarlas a una palabra raíz. No necesariamente tienen que ser palabras raíz para el lenguaje, pero pueden ser palabras que si le cambias algún sufijo, puede aparecer una nueva palabra. Por ejemplo: “curr”, puede ser una palabra preprocesada de las palabras “currency”, o “currently”, o “curry”, hay varias posibilidades, por la cual es el preprocesamiento que más nos ayudó a reducir el tamaño de las oraciones.

```
def stemming_word(input_str):
    stemmer= PorterStemmer()
    expanded_words = []
    input_str=word_tokenize(input_str)
    for word in input_str:
        expanded_words.append(stemmer.stem(word))
    expanded_text = ' '.join(expanded_words)
    return(expanded_text)
```

Separamos las oraciones por palabras e invocamos cada función para hacer esta etapa.

Y finalmente, usamos la librería de pandas para guardar en archivos diferentes el conjunto de textos preprocesados de cada tipo.

```
Longitud promedio de textos por clase 1 / Lemas= 150.27586206896552 clase 0= 146.08804347826086
Longitud promedio de textos por clase 1 / Stopwords= 100.26379310344828 clase 0= 101.06086956521739
Longitud promedio de textos por clase 1 / Steeming= 143.22413793103448 clase 0= 140.1478260869565
Numero de palabras por clase 0 Lemas= 26050 clase 1= 16854
Numero de palabras por clase 0 Stopwords= 14126 clase 1= 8796
Numero de palabras por clase 0 Steeming= 26050 clase 1= 16854
```

## Clasificación

Los algoritmos que utilizamos son: Árbol de decisión, Naive Bayes y Random Forest Para el desarrollo de estos algoritmos utilizamos el lenguaje de programación python y scikit-learn.

Los “Árboles de decisión” son un método de aprendizaje supervisado no paramétrico que se utiliza para clasificación y regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo aprendiendo reglas de decisión simples inferidas de las características de los datos. Un árbol puede verse como una aproximación constante por partes.

Los métodos de “Naive Bayes” son un conjunto de algoritmos de aprendizaje supervisados basados en la aplicación del teorema de Bayes con la suposición “ingenua” de independencia condicional entre cada par de características dado el valor de la variable de clase.

Nosotros implementamos MultinomialNB implementa el algoritmo ingenuo de Bayes en datos distribuidos multinomialmente, y es una de las dos variantes clásicas ingenuas de Bayes utilizadas en la clasificación de texto porque usualmente es para los datos se representan típicamente como recuentos de vectores de palabras.

El algoritmo de “Random Forest” es una forma de elevar el algoritmo de árboles de decisión. Esto lo hace mediante la creación de diversos árboles diferentes, las cuales al final se unen para poder conseguir un resultado con muchas más probabilidades de ser la decisión correcta.

Cada árbol en el bosque de árboles (forest) escoge una clase como resultado. Para determinar la mejor respuesta, se analiza cada árbol y se escoge la clase que tenga más votos a favor en el bosque.

## Evaluación

### Base de datos “Training” - Salidas

Estos resultados salen de la base de datos de Training. Utilizamos 1000 de los 1500 originales para que sean el “entrenamiento” y los 500 restantes para que sean las pruebas de verificación de la asignación de clases. Recordemos que cada preprocesamiento tiene su diccionario diferente, por lo que cada resultado tiene un vector de características de diferente tamaño.

**Árbol de decisión resultados**:

Lematización

```
Matriz de confusión
                |  1  |  0
 Xenófobico     |  160  |  40
 No Xenofóbico |  47  |  253

Porcentaje correcto=  82.6

Porcentaje incorrecto=  17.4

Precisión=  0.8

Recuerdo=  0.7729468599033816

F-score=  0.7862407862407863
```

Stemming

```
Matriz de confusión
                |  1  |  0
 Xenófobico     |  159  |  31
 No Xenofóbico |  48  |  262

Porcentaje correcto=  84.2

Porcentaje incorrecto=  15.8

Precisión=  0.8368421052631579

Recuerdo=  0.7681159420289855

F-score=  0.801007556675063
```

Stop words

```
Matriz de confusión
                |  1  |  0
 Xenófobico     |  162  |  39
 No Xenofóbico |  45  |  254

Porcentaje correcto=  83.2

Porcentaje incorrecto=  16.8

Precisión=  0.8059701492537313

Recuerdo=  0.782608695652174

F-score=  0.7941176470588236
```

Análisis: Comparando los resultados de los diferentes pre procesamientos, determinamos que al utilizar **Stemming** nos fue mucho mejor en predecir las clases correctas, según el archivo original. Además, tiene el mejor F-score, y mientras más se acerque al valor de 1, el algoritmo es considerado mejor que el resto. Seguido por el de lematización y finalmente eliminando stop words. no obstante, si sólo queremos enfocarnos en el análisis de la pregunta principal de si es la oración xenófoba o no, el preprocesamiento de **Lematización** tiene una mejor precisión, seguido por stemming y de nuevo al final con stop words.

No obstante, analizando todos los algoritmos, nos percatamos que este tipo de preprocesamiento ha sido el que arroja mejores resultados en la sección de **recuerdo**, siendo más precisos el que utiliza la **eliminación de stop words**.

**Naive Bayes resultados**:

Lematización

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 148 | 72 |
| No Xenofóbico | 59 | 221 |

Porcentaje correcto= 73.8

Porcentaje incorrecto= 26.200000000000003

Precisión= 0.6727272727272727

Recuerdo= 0.714975845410628

F-score= 0.6932084309133489

Stemming

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 150 | 66 |
| No Xenofóbico | 57 | 227 |

Porcentaje correcto= 75.4

Porcentaje incorrecto= 24.6

Precisión= 0.6944444444444444

Recuerdo= 0.7246376811594203

F-score= 0.7092198581560285

Stop words

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 140 | 60 |
| No Xenofóbico | 67 | 233 |

Porcentaje correcto= 74.6

Porcentaje incorrecto= 25.4

Precisión= 0.7

Recuerdo= 0.6763285024154589

F-score= 0.6879606879606879

Análisis: En Bayes se ve una reducción en la precisión y en el recuerdo en los 3 tipos de pre procesamientos. En esta ocasión, al usar **Stop words** se consiguió la mejor precisión, pero el peor recuerdo. No obstante, no es el que tiene el mayor porcentaje de predicciones correctas, ese va para el de **Lematización**. El que tiene un mejor “promedio” entre la precisión y el recuerdo (F-score) es el que utiliza **Stemming**. Esto implica que, abarcando tanto precisión como recuerdo, es el que tiene una mejor armonía con los datos ingresados y las predicciones del algoritmo.

**Random Forest resultados**

Lematización

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 142 | 21 |
| No Xenofóbico | 65 | 272 |

Porcentaje correcto= 82.8

Porcentaje incorrecto= 17.2

Precisión= 0.8711656441717791

Recuerdo= 0.6859903381642513

F-score= 0.7675675675675675

Stemming

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 147 | 26 |
| No Xenofóbico | 60 | 267 |

Porcentaje correcto= 82.8

Porcentaje incorrecto= 17.2

Precisión= 0.8497109826589595

Recuerdo= 0.7101449275362319

F-score= 0.7736842105263159

Stop words

Matriz de confusión

| | 1 | 0 |
|---|---|---|
| Xenófobico | 157 | 33 |
| No Xenofóbico | 50 | 260 |

Porcentaje correcto= 83.39999999999999

Porcentaje incorrecto= 16.6

Precisión= 0.8263157894736842

Recuerdo= 0.7584541062801933

F-score= 0.7909319899244334

Análisis: Mediante Random Forest, el **lematizado** tiene la mayor diferencia entre la precisión y el recuerdo. Es el algoritmo con mayor precisión entre todos los probados incluyendo con los algoritmos anteriores de Bayes y árbol de decisión. Por lo que si quisiéramos buscar un algoritmo que nos arroje menos cantidad de resultados pero estos sean los más confiables, en **Random forest con lematización** sería la mejor opción.

## Base de datos “Testing” - Salidas

**Árbol de decisión resultados**:

Lematización

| | Text | Class |
|---|---|---|
| 0 | ye bhi illegal immigrant lag rahi hai nrc regi... | 1 |
| 1 | yeah i ask someone about her skirt today and s... | 1 |
| 2 | you need to clean your selective hear cobwebs ... | 1 |
| 3 | it easy to complain and criticize with no offe... | 1 |
| 4 | guess the sendthemback policy of your friend h... | 0 |
| ... | ... | ... |
| 495 | by the way chucky when obama be in there you s... | 1 |
| 496 | exactly obama use strong language about border... | 0 |
| 497 | if corporations can be sue and punish for the ... | 1 |
| 498 | lara trump accuse critics of send her back of ... | 1 |
| 499 | trump reverse again say crowd chant | 0 |

Stemming

| | Text | Class |
|---|---|---|
| 0 | ye bhi illeg immigr lag rahi hai nrc regist me... | 1 |
| 1 | yeah i ask someon about her skirt today and sh... | 1 |
| 2 | you need to clean your select hear cobweb out ... | 1 |
| 3 | it easi to complain and critic with no offer o... | 1 |
| 4 | guess the sendthemback polici of your friend h... | 1 |
| ... | ... | ... |
| 495 | by the way chucki when obama wa in there you s... | 1 |
| 496 | exactli obama use strong languag about border ... | 0 |
| 497 | if corpor can be su and punish for the opioid ... | 1 |
| 498 | lara trump accus critic of send her back of pu... | 1 |
| 499 | trump revers again say crowd chant | 0 |

Stop words

| | Text | Class |
|---|---|---|
| 0 | ye bhi illegal immigrant lag rahi hai nrc regi... | 1 |
| 1 | yeah asked someone skirt today said asos like ... | 1 |
| 2 | need clean selective hearing cobwebs ears tell... | 1 |
| 3 | easy complain criticize offering solution clos... | 1 |
| 4 | guess sendthemback policy friend hit little cl... | 1 |
| ... | ... | ... |
| 495 | way chucky obama seem want deport illegal alie... | 1 |
| 496 | exactly obama used strong language borders rap... | 0 |
| 497 | corporations sued punished opioid epidemic sta... | 1 |
| 498 | lara trump accuses critics send back pushing b... | 1 |
| 499 | trump reverses says crowd | 1 |

Análisis: Relacionando los resultados anteriores, asumimos que la versión de **Stemming** es la más acertada, ya que en la fase de training, fue la que obtuvo un mayor porcentaje de precisión. No obstante, los resultados se parecen mucho al obtenido en la lematización.

**Naive Bayes resultados**:

## Lematización

| | Text | Class |
|---|---|---|
| 0 | ye bhi illeg immigr lag rahi hai nrc regist me... | 1 |
| 1 | yeah i ask someon about her skirt today and sh... | 0 |
| 2 | you need to clean your select hear cobweb out ... | 1 |
| 3 | it easi to complain and critic with no offer o... | 1 |
| 4 | guess the sendthemback polici of your friend h... | 1 |
| ... | ... | ... |
| 495 | by the way chucki when obama wa in there you s... | 1 |
| 496 | exactli obama use strong languag about border ... | 0 |
| 497 | if corpor can be su and punish for the opioid ... | 1 |
| 498 | lara trump accus critic of send her back of pu... | 0 |
| 499 | trump revers again say crowd chant | 0 |

## Stemming

| | Text | Class |
|---|---|---|
| 0 | ye bhi illeg immigr lag rahi hai nrc regist me... | 1 |
| 1 | yeah i ask someon about her skirt today and sh... | 0 |
| 2 | you need to clean your select hear cobweb out ... | 1 |
| 3 | it easi to complain and critic with no offer o... | 1 |
| 4 | guess the sendthemback polici of your friend h... | 1 |
| ... | ... | ... |
| 495 | by the way chucki when obama wa in there you s... | 1 |
| 496 | exactli obama use strong languag about border ... | 0 |
| 497 | if corpor can be su and punish for the opioid ... | 1 |
| 498 | lara trump accus critic of send her back of pu... | 0 |
| 499 | trump revers again say crowd chant send her ba... | 0 |

## Stop words

| | Text | Class |
|---|---|---|
| 0 | ye bhi illegal immigrant lag rahi hai nrc regi... | 1 |
| 1 | yeah asked someone skirt today said asos like ... | 0 |
| 2 | need clean selective hearing cobwebs ears tell... | 0 |
| 3 | easy complain criticize offering solution clos... | 1 |
| 4 | guess sendthemback policy friend hit little cl... | 1 |
| ... | ... | ... |
| 495 | way chucky obama seem want deport illegal alie... | 1 |
| 496 | exactly obama used strong language borders rap... | 0 |
| 497 | corporations sued punished opioid epidemic sta... | 1 |
| 498 | lara trump accuses critics send back pushing b... | 0 |
| 499 | trump reverses says crowd | 0 |

Análisis: En esta prueba, se logró observar que en la muestra de 10 oraciones distintas, se obtuvieron resultados iguales, excepto en un texto del preprocesamiento **stop words**. De igual manera, no podemos decir cuál fue mejor, debido a que tuvimos que probar los algoritmos sin ninguna respuesta oficial a las clases asignadas, pero podemos acercarnos gracias a los algoritmos programados.

## Conclusiones

Mediante los 3 tipos de pre procesamientos pudimos conseguir diversos resultados tanto para la precisión y el recuerdo. Gracias a las matrices de confusión pudimos también resaltar cómo variaron los resultados en la sección de “Training”. Al aplicar los algoritmos probados en la base de datos de Testing pudimos obtener resultados confiables hasta cierto punto. Recordando que el que ofrecía mejor precisión de todos era **Random forest con lematización**, y el que ofrecía mejor nivel de

recuerdo era el **árbol de decisión preprocesado con la eliminación de stop words**.

Juan Diego Bastidas Santivañez

La clase abarcó temas interesantes, en especial en el último tramo del semestre, y justo fue en la parte donde empezamos a programar nuestros algoritmos. Ahí empecé a sentir lo potente que pueden ser los algoritmos de machine learning. Mi forma de entender mejor los temas es realizando los ejercicios de forma tangible, y me agradó poder codificar un algoritmo con un cierto nivel de inteligencia para poder determinar las clases correspondientes de algún vector de características. Siento que si profundizo más en el tema, podría ser una herramienta muy útil en mi futuro, ya que cada vez más la inteligencia artificial se adentra en todas las áreas, incluso en lo más pequeño, por lo que podría serme de utilidad en algún momento, y con los conocimientos adquiridos en este semestre, siento que podría defenderme al inicio, y después yo seguir por mi propio camino, y agradezco haber aprendido de esto a tan temprana edad.

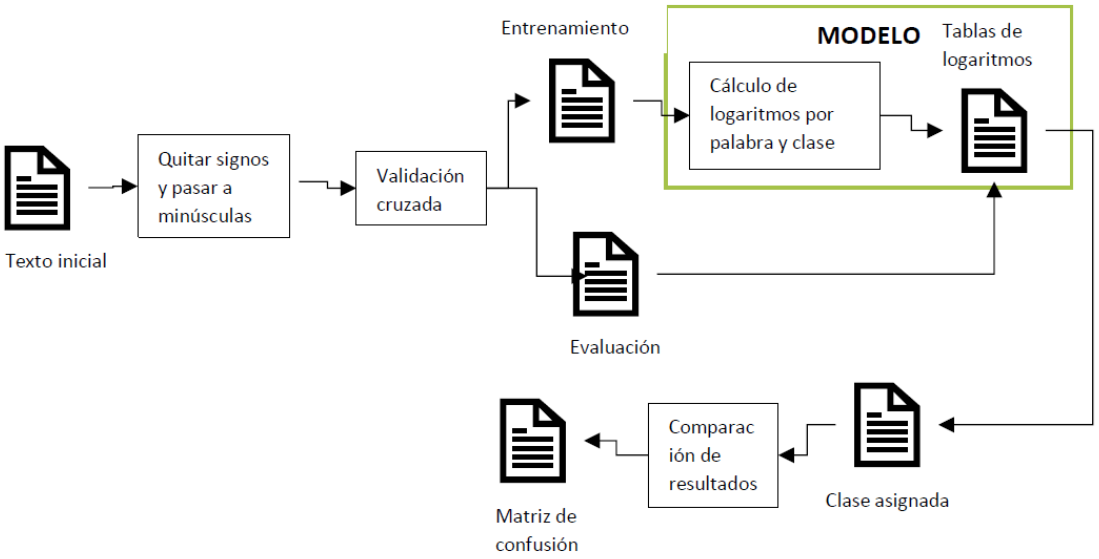
Kimberly Atara López Vazquez

En lo personal me pareció muy interesante la materia, no me imaginé que podría ser capaz de programar algoritmos inteligentes, capaces de predecir una clase. Siempre había pensado que era una materia muy complicada, sin embargo logramos un proyecto muy interesante implementando distintos algoritmos. Cada una de las actividades que hicimos en clase y tareas nos hicieron aprender gradualmente, fue muy divertido ver los resultados. La investigación y leer documentación también fue muy importante para llegar al resultado. En conclusión ha sido un proyecto interesante y considero que se ha usado casi todo lo que hemos aprendido durante el semestre para implementar varias cosas. Tanto me gustó la materia que ahora estoy en un curso de Data Science en BEDU por medio de una beca de Santander.

## Análisis del proyecto

### Defensa/Explicación del proyecto con el docente en las fechas acordadas

Básicamente lo que nosotros hicimos fue utilizar diferentes tipos de algoritmos y diferentes tipos de preprocesamiento para analizar la eficiencia de los mismos. Pasando por cada una de las distintas etapas mencionadas en este diagrama.




## Referencias

Scikit-learn. Naive Bayes. Recuperado de: scikit-learn.org/stable/modules/naive_bayes.html

Scikit-learn. Decision Trees. Recuperado de: https://scikit-learn.org/stable/modules/tree.html

Scikit-learn. Random Forest Classifier. Recuperado de: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Candy Yuridiana. Naive Bayes. Recuperado de: experiencia.tec.mx