

**Integrantes:**

Universidad de los Andes Ingeniería de Sistemas y Computación

ISIS 3301 Inteligencia de Negocios

**Proyecto 1**

**Machine Learning:**

**Procesamiento de Textos**

* Juan David Becerra – 201911588 - Regresión logística
* Juan Andrés Santiago – 201821950 - Random forest
* Nicolas Chalee Guerrero –201912737 - Naive Bayes

**Tabla de contenido**

1. [Comprensión del Negocio y Enfoque Analítico](#_bookmark0) [1](#_bookmark0)
2. [Comprensión y preparación de los datos](#_bookmark1) [2](#_bookmark1)
3. [Modelado y Evaluación](#_bookmark2) [4](#_bookmark2)  
   1. [Regresión Logística](#_bookmark3) [4](#_bookmark3)
   2. [Random Forest](#_bookmark4) [5](#_bookmark4)
   3. [Naive Bayes](#_bookmark5) [6](#_bookmark5)
4. [Resultados](#_bookmark6) [7](#_bookmark6)
5. [Trabajo en Equipo](#_bookmark7) [8](#_bookmark7)

# Comprensión del Negocio y Enfoque Analítico

|  |  |
| --- | --- |
| Oportunidad / Problema del negocio | El problema que el negocio quiere es lograr determinar la elegibilidad de un paciente para ensayos clínicos de cáncer a partir de texto descriptivo. |
| Descripción del requerimiento desde el punto de vista de machine learning | El negocio requiere de modelos de machine learning capaces de predecir si un paciente es elegible o no en base a los estudios y la condición que este tiene. En consecuencia, se va a hacer uso de algoritmos NLP (Natural Language Processing) con el fin de predecir la elegibilidad de un paciente dado sus estudios y condición. |



# Comprensión y preparación de los datos

En la Figura 1 se puede apreciar un fragmento del dataset que, en sí, consiste de 12000 filas y 2 columnas, dichas son ‘label y ‘study\_and\_condition. Con base a estas, se va a realizar la predicción de la elegibilidad de un paciente según su estudio y condición actual.

Texto, Carta

Descripción generada automáticamente

Figura 1. Fragmento dataset

Luego de realizar un conteo de todos los ‘label’, se pudo destacar que los datos están perfectamente balanceados.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Figura 2. Conteo de ‘label’

En las Figuras 3, 4 y 5 está el código utilizado para el preprocesamiento del texto de las ‘study\_and\_condition’. Dentro de este proceso se removió los caracteres non-ASCII, se pasó todo a minúscula, se limpió el texto de caracteres de puntuación, se eliminaron enteros y se eliminaron palabras ‘vacías’ como artículos o pronombres junto con contracciones del inglés. Por último, se realizó el proceso de stemming y lematización. Este proceso consiste en reducir las palabras a su raíz, eliminando sufijos o prefijos. Todos estos pasos son vitales para la analítica de textos y van a mejorar considerablemente el desempeño y métricas de las tareas realizadas por los modelos de machine learning utilizados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 3. Preprocesamiento del texto parte 1

Texto

Descripción generada automáticamente

Figura 4. Preprocesamiento del texto parte 2

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 5. Preprocesamiento del texto parte 3

Finalmente, después de todo el preprocesamiento realizado a los datos de todo el dataframe queda como resultado las dos variables de ‘label’ y ‘study\_and\_condition’ y una adicional ‘words’ que son las oraciones separadas por las palabras transformadas. Estas variables serán utilizadas para las tareas de los modelos de machine learning.

Tabla

Descripción generada automáticamente

Figura 6. Tabla después del procesamiento

# Modelado y Evaluación

## Regresión Logística

La Regresión Logística es un modelo estadístico que estudia la relacion entre una variavle cualitativa X y multiples variables cualitativas Yi. Ayuda a predecir la probabilidad de que ocurra un evento o no, de forma binaria, a partir de la optimizacion de los coeficientes de regresion.

Aplicada al caso de elegibilidad de un paciente para ensayos clínicos de cáncer

pesar de su nombre no es un algoritmo para aplicar en problemas de regresión en los que se busca un valor continuo, sino que es un método para problemas de clasificación, en los que se obtienen un valor binario entre 0 y 1, esto nos favorece debido a la transformación del puntaje en la nueva columna binaria “Rating”. A partir de matrices de pesos aleatorias ajustadas las cuales representan el pool de palabras con todas sus características y un vector de tokens los cuales se multiplican entre si basado en un punto dado para obtener un valor escalar, el cual es luego pasado a una función sigma en la cual se produce un valor binario para clasificar dicho punto.

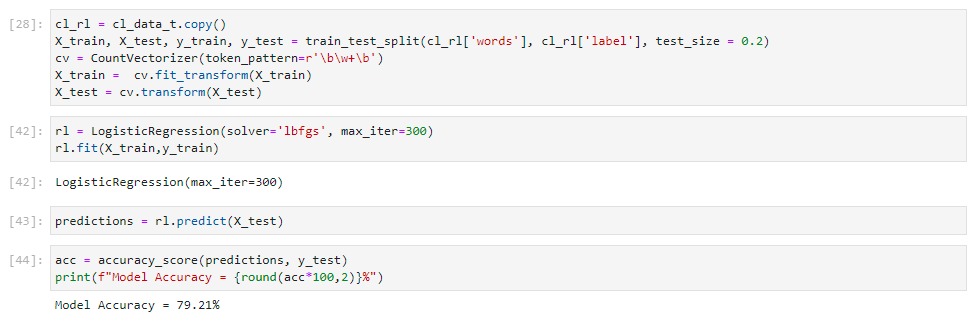


Figura 7. Implementación del modelo en JupyterLab

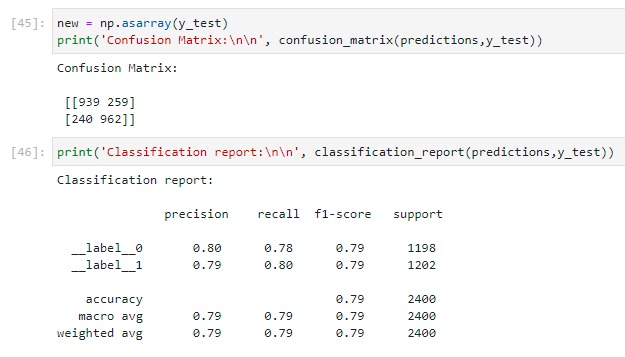


Figura 8. Resultados del modelo

El modelo de Regresión Logística tiene una precisión del 80% para pacientes elegibles y un 79% para pacientes no elegibles. El accuracy del modelo es de 79.21%.

En lo correspondiente a la matriz de confusión, se puede observar que la mayoría de datos se clasificaron de manera correcta. La cantidad de datos clasificados como pacientes elegibles correctamente fue de 939, y para no elegibles fue de 962.

## Random Forest

Un algoritmo de Random Forest consta de muchos árboles de decisión. En comparación a los otros algoritmos, Random Forest está más enfocado en tener la máxima exactitud posible. Finalmente, este algoritmo maneja muy bien los datos categóricos que en nuestro caso es mayoría entre los datos presentados.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 9. Implementación Random Forest

El modelo de Random Forest tiene una precisión del 80% para pacientes elegibles y un 81% para no elegibles, el accuracy total del modelo es del 81%. Como se presenta un balanceo exacto de los datos, los resultados son bastantes similares. Se presentan falsos positivos y falsos negativos, sin embargo, existe mucha más cantidad en los verdaderos negativos y verdaderos positivos.

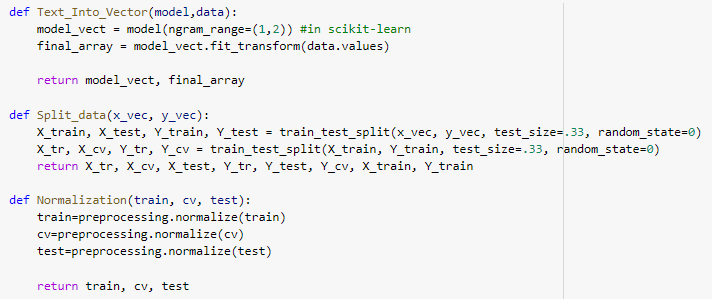
Gráfico

Descripción generada automáticamente

Figura 10. Resultados de Random Forest

## Naive Bayes

Para el algoritmo de Naive Bayes se debe tomar los párrafos que se tienen como datos y organizarlas en un vector de frecuencia, después de eso y de definir las diferentes funciones que se van a utilizar sobre los datos se llaman a esas mismas para que sea organizadas en la matriz de frecuencia, para después convertir esa misma matriz en una estructura de datos donde será implementada el algoritmo deseado, puede surgir problemas con el tamaño de la matriz, pero dado que las frases son cortas no se cree que se presente un problema.



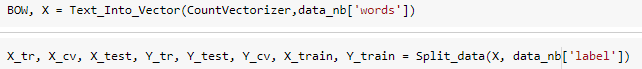


Figura 9. Conversión de los datos a vectores

Se aplica la implementación multinomial del algoritmo de Naive Bayes ya que los datos de texto son discretos es decir que son exactos en el sentido que son finitos y enteros. Se aplica el algoritmo para después obtener la matriz de confusión.

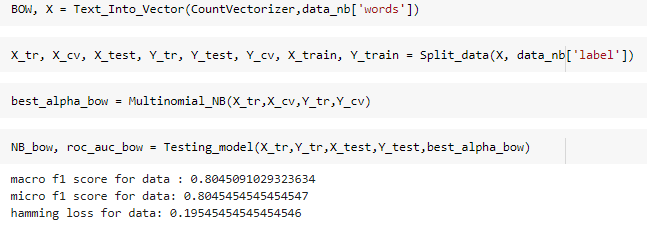


Figura 10. Aplicación de la ecuación de Naive Bayes

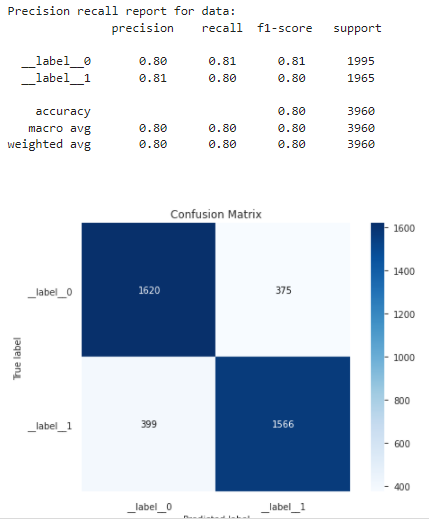


Figura 11. Resultados de Naive Bayes

Dando como resultado un total de 80,45% ya que gran parte de los datos se encuentran en los falsos que realmente son falsos y los verdaderos que realmente son verdaderos. Siendo los otros grupos de falsos positivos y falsos negativos una representación menor.

# Resultados

Para comenzar, en la fase de perfilamiento de los datos realizamos un wordcloud para ver las palabras más recurrentes del dataset, teniendo en cuenta la limpieza de datos realizada previamente. Esto con el objetivo de dar una idea del impacto que puede llegar a tener cada una al momento de aplicar cada uno de los modelos propuestos. El resultado se puede apreciar en la figura 12.

Texto, Escala de tiempo

Descripción generada automáticamente

Figura 12. Resultados Generales

Se realizó un wordcloud para cada conjunto (‘label 0’ / ‘label 1’) con el propósito de visualizar las palabras más recurrentes (figura 13 y figura 14 respectivamente). Con esto, se tendrá un panorama más claro sobre el impacto de las palabras al momento elegir un paciente para los ensayos clínicos cuando se apliquen los modelos propuestos.

Texto

Descripción generada automáticamente

Figura 13. Resultados pacientes elegibles

Texto

Descripción generada automáticamente

Figura 14. Resultados pacientes no elegibles

De las figuras presentadas previamente se puede apreciar la frecuencia de las diferentes palabras. En primera instancia se tiene todas las palabras relacionadas con ‘lymphoma’ que son muy frecuentes en las tres imágenes. Es por esto por lo que no es de mucha utilidad este tipo de palabras para determinar la elegibilidad de un paciente. De igual manera se tienen palabras como ‘study’, ‘diagnosis’ y ‘stag’ que se presentan en una gran mayoría en las tres figuras. Por otro lado, en el wordcloud de resultados de pacientes elegibles se presenta una buena cantidad de ‘study intervention antibody’ y ‘breast cancer diagnosis’ que podrían ser un indicio para ser elegible un paciente. Sin embargo, es importante mencionar que se presentan esas palabras en los resultados de pacientes no elegibles es por esto que no se logra una exactitud de 100%, sino un poco mas baja.

El mejor modelo de los 3 seleccionados es Naive Bayes, esto se debe a la forma en cómo funciona; donde divide el dataset en dos partes (en nuestro caso reviews positivas y negativas) y calcula la probabilidad de ver las palabras positivas en esos reviews, y lo mismo con las palabras negativas. Cuando le llegue un String determina la probabilidad de ver cada palabra de la cadena en uno de los dos grupos para así al final clasificar el nuevo texto como review mala o buena. Por otro lado, se presume que con una muestra más grande de reviews con 2 estrellas o menos en el dataset, la precisión en las predicciones de las reviews negativas subiría en los modelos de Regresión Lineal y Random Forest. Sin embargo, también es necesario tener en cuenta que varía en gran medida las palabras utilizadas para referirse a una review negativa en comparación a una positiva. Otro factor para considerar es que las reviews positivas tienden a ser mucho más genéricas, y la gran mayoría utiliza los mismos calificativos, mientras que las reviews negativas tienen descripciones mucho más detalladas dificultando el proceso de clasificación a partir de la analítica de textos. Debido a esto, tanto Regresión Lineal y Random Forest son muy eficientes para predecir cuándo un comentario va a representar una buena acogida para el producto, pero, puede llegar a tener un poco más de dificultades prediciendo cuando va a ser una mala recepción.

# Trabajo en Equipo

Cada uno de los miembros del equipo se encargó de la implementación de uno de los diferentes modelos y algoritmos de machine learning. La persona que asumió el rol de líder del proyecto fue Juan Andrés Santiago, el líder de negocio es Juan David Becerra y el líder de datos y analítica es Nicolas Chalee Guerrero.

Los mayores retos enfrentados en el desarrollo del proyecto fue en primer lugar la inexperiencia con el uso e implementación de los algoritmos NLP, en segundo lugar, debido al gran volumen de los datos fue difícil la implementación de un preprocesamiento óptimo de los datos y por la naturaleza del problema que se quería resolver era necesario hacer varias modificaciones a los datos para que los modelos funcionaran de forma óptima.

Adicionalmente los elevados tiempos de ejecución de los modelos hicieron que fuera muy costoso implementar correcciones y hacer comparaciones entre resultados para su optimización. Cada estudiante se encargó de la implementación de su algoritmo correspondiente y cada uno destine mayor tiempo a ciertas tareas según su rol asignado. Nicolás estuvo más pendiente al final en la logística de la totalidad del proyecto, sus entregables y que cada uno cumpliera con sus funciones. Juan José estuvo más activo en las partes que era necesario justificar al negocio los resultados y problemáticas. Finalmente, Juan Pablo aportó más a la parte de preprocesamiento de datos y la implementación de los algoritmos NLP. En promedio cada integrante aportó un promedio de 5.5 horas al desarrollo del proyecto.

Repartición de puntos:

* Juan Nicolas Bolaños: 35
* Juan José Ochoa: 31
* Juan Pablo Sarmiento: 34

Lo anterior tiene justificación en la proporción de horas invertidas en el proyecto y el grado de dificultad de cada una de las tareas asignadas.