# Documento Proyecto 1

**Inteligencia de Negocios**
**ISIS-3301**

**Grupo 13**
**Simón Rendón - 201820112**
**Juan Díaz - 201729408**
**Manuel Sosa - 201815393**

**28 marzo 2022**

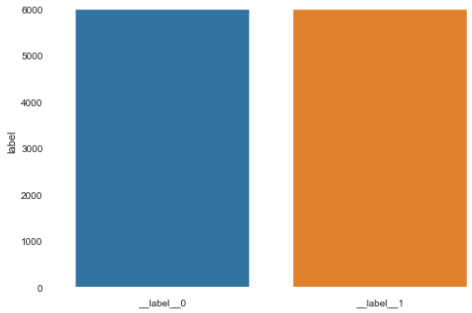
## 1. Comprensión del negocio y enfoque analítico

| | | |
|---|---|---|
| **Oportunidad/problema del negocio** | Se busca automatizar el proceso de identificación de pacientes con cáncer elegibles para ensayos clínicos a partir de textos descriptivos. | |
| **Descripción del requerimiento desde el punto de vista de aprendizaje de máquina** | Se requiere usar análisis de texto para darle solución al problema planteado por el negocio. En primer lugar, se debe utilizar el procesamiento de lenguaje natural para la extracción de conocimiento y posteriormente implementar un modelo de aprendizaje de máquina supervisado de clasificación. | |
| Detalles de la actividad de minería de datos | | |
| **Tipo de aprendizaje** | **Tarea de aprendizaje** | **Algoritmo e hiper-parámetros utilizados (con la justificación respectiva)** |
| Supervisado | Clasificación | Support Vector Machine |
| Supervisado | Clasificación | Naive Bayes |
| Supervisado | Clasificación | Logistic Regression |

## 2. Comprensión y preparación de los datos

Lo primero que podemos observar al cargar los datos es que estos solo cuentan con dos columnas (‘label’, ‘study_and_condition’) con un total de 12000 registros (filas). No se evidencia la existencia de nulos o vacíos en la base de datos. Adicionalmente, en la columna ‘label’ solo contamos con dos opciones categóricas de tipo string: __label__0 y __label__1 que hacen referencia a si son elegibles o no, siendo __label__1 la que corresponde a ser elegible. Por otro lado, en la columna ‘study_and_condition’ contamos con cadenas de textos que hacen referencia al estudio y condición del paciente; estos separados entre sí por un punto.

En toda la base de datos solo contamos con 12 duplicados, como son tan pocos decidimos no elimnarlos. También podemos ver que los datos se encuentran perfectamente distribuidos entre __label__0 y __label__1, cada uno con 6000 datos.




El promedio de palabras que tienen tiene un campo que es tipo __label__0 es de 23.132 y para el __label__1 es de 25.289. Mientras que el promedio de caracteres para __label__0 es de 165.273 y para __label__1 es de 182.292.

En cuanto a la preparación de los datos consideramos importante crear dos nuevas columnas separando el estudio y la condición. Esto con el fin de poder saber si los modelos son efectivos con solo el estudio o con solo la condición o con ambos.

Ahora procedemos a hacer el preprocesamiento de texto, para esto pasamos por 3 etapas: limpieza de datos, tokenización y normalización.

Para la limpieza de los datos pasamos todas las palabras a minúsculas, pasamos todos los números a su representación en texto (ejemplo: 6 pasa a ser seis), removemos los signos de puntuación, removemos los caracteres no ASCII y eliminamos los stopwords.

La tokenización permite dividir frases u oraciones en palabras. Con el fin de desglosar las palabras correctamente para el posterior análisis.

En la normalización de los datos se realiza la eliminación de prefijos y sufijos, además de realizar una lematización.

## 3. Modelado y evaluación

### 3.1 Logistic Regression (Juan David Díaz)

Uno de los algoritmos seleccionados fue Regresión Logística, el cual es similar a Regresión Lineal puesto que también se usa para predecir la ausencia o presencia de una característica según un conjunto de valores predictores. Algo esencial del algoritmo escogido es que se usa cuando la variable dependiente es binaria. Es precisamente por esta última razón que se escogió en esta ocasión el modelo de regresión logística, ya que las variables objetivo o categorías son dos. Esto permite adaptar de una mejor forma la base de datos proveída por el negocio al modelo y así garantizar un mejor desempeño.

Tras documentarnos al respecto, hallamos que la calidad de la clasificación puede variar dependiendo de cómo se divida la columna a estudiar. Por esta razón se realizaron tres casos de prueba para encontrar en qué caso existía un mejor desempeño y sobre eso obtener hallazgos más precisos y de mejor calidad.

**Columna 'study'**
Qué tan bien se puede predecir la variable objetivo con el componente study del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.54 | 0.46 | 0.50 | 1204 |
| __label__1 | 0.53 | 0.61 | 0.56 | 1196 |
| accuracy | | | 0.53 | 2400 |
| macro avg | 0.53 | 0.53 | 0.53 | 2400 |
| weighted avg | 0.53 | 0.53 | 0.53 | 2400 |

**Columna 'condition'**
Qué tan bien se puede predecir la variable objetivo con el componente condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.80 | 0.79 | 0.80 | 1214 |
| __label__1 | 0.79 | 0.80 | 0.80 | 1186 |
| accuracy | | | 0.80 | 2400 |
| macro avg | 0.80 | 0.80 | 0.80 | 2400 |
| weighted avg | 0.80 | 0.80 | 0.80 | 2400 |

**Columna 'study_condition'**
Qué tan bien se puede predecir la variable objetivo con el componente study_condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.81 | 0.79 | 0.80 | 1225 |
| __label__1 | 0.79 | 0.80 | 0.79 | 1175 |
| accuracy | | | 0.80 | 2400 |
| macro avg | 0.80 | 0.80 | 0.80 | 2400 |
| weighted avg | 0.80 | 0.80 | 0.80 | 2400 |

### 3.2 Naive Bayes (Manuel Sosa)

Naive Bayes es una técnica de clasificación estadística basada en el Teorema de Bayes. Es uno de los algoritmos de aprendizaje supervisado más simples. El clasificador Naive Bayes es un algoritmo rápido, preciso y confiable. Los clasificadores Naive Bayes tienen alta precisión y velocidad en grandes conjuntos de datos.

Tras documentarnos al respecto, hallamos que la calidad de la clasificación puede variar dependiendo de cómo se divida la columna a estudiar. Por esta razón se realizaron tres casos de prueba para encontrar en qué caso existía

un mejor desempeño y sobre eso obtener hallazgos más precisos y de mejor calidad.

**Columna 'study'**
Qué tan bien se puede predecir la variable objetivo con el componente study del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.52 | 0.67 | 0.58 | 1177 |
| __label__1 | 0.56 | 0.41 | 0.47 | 1223 |
| accuracy | | | 0.54 | 2400 |
| macro avg | 0.54 | 0.54 | 0.53 | 2400 |
| weighted avg | 0.54 | 0.54 | 0.53 | 2400 |

**Columna 'condition'**
Qué tan bien se puede predecir la variable objetivo con el componente condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.78 | 0.77 | 0.78 | 1156 |
| __label__1 | 0.79 | 0.80 | 0.79 | 1244 |
| accuracy | | | 0.79 | 2400 |
| macro avg | 0.79 | 0.79 | 0.79 | 2400 |
| weighted avg | 0.79 | 0.79 | 0.79 | 2400 |

**Columna 'study_condition'**
Qué tan bien se puede predecir la variable objetivo con el componente study_condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.81 | 0.77 | 0.79 | 1223 |
| __label__1 | 0.78 | 0.81 | 0.79 | 1177 |
| accuracy | | | 0.79 | 2400 |
| macro avg | 0.79 | 0.79 | 0.79 | 2400 |
| weighted avg | 0.79 | 0.79 | 0.79 | 2400 |

**3.3 Support Vector Machine (Simón Rendón)**
El último algoritmo elegido para la tarea de clasificación es Support Vector Machine (SVM). Este algoritmo consiste en organizar los datos de entrenamiento en un plano de 2 dimensiones y posteriormente encontrar un hiperplano que separe las dos categorías a clasificar permitiendo así identificar la categoría de nuevos datos. Este algoritmo es, normalmente, utilizado únicamente en clasificación binaria, lo cual se ajusta de buena manera al problema que queríamos resolver y por dicho motivo lo escogimos.

Tras documentarnos al respecto, hallamos que la calidad de la clasificación puede variar dependiendo de cómo se divida la columna a estudiar. Por esta

razón se realizaron tres casos de prueba para encontrar en qué caso existía un mejor desempeño y sobre eso obtener hallazgos más precisos y de mejor calidad.

**Columna 'study'**

Qué tan bien se puede predecir la variable objetivo con el componente study del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.55 | 0.52 | 0.53 | 1204 |
| __label__1 | 0.54 | 0.58 | 0.56 | 1196 |
| accuracy | | | 0.55 | 2400 |
| macro avg | 0.55 | 0.55 | 0.55 | 2400 |
| weighted avg | 0.55 | 0.55 | 0.55 | 2400 |

**Columna 'condition'**

Qué tan bien se puede predecir la variable objetivo con el componente condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.79 | 0.79 | 0.79 | 1185 |
| __label__1 | 0.79 | 0.80 | 0.80 | 1215 |
| accuracy | | | 0.79 | 2400 |
| macro avg | 0.79 | 0.79 | 0.79 | 2400 |
| weighted avg | 0.79 | 0.79 | 0.79 | 2400 |

**Columna 'study_condition'**

Qué tan bien se puede predecir la variable objetivo con el componente study_condition del texto. Se halló lo siguiente:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.77 | 0.80 | 0.78 | 1151 |
| __label__1 | 0.81 | 0.78 | 0.79 | 1249 |
| accuracy | | | 0.79 | 2400 |
| macro avg | 0.79 | 0.79 | 0.79 | 2400 |
| weighted avg | 0.79 | 0.79 | 0.79 | 2400 |

**Columna 'condition' con hiperparametros ajustados**

Una vez vimos que el modelo funcionaba mejor con la columna condition decidimos usar GridSearch para calcular los hiperparametros más convenientes. El resultado fue :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| __label__0 | 0.82 | 0.82 | 0.82 | 1185 |
| __label__1 | 0.82 | 0.83 | 0.82 | 1215 |
| accuracy | | | 0.82 | 2400 |
| macro avg | 0.82 | 0.82 | 0.82 | 2400 |
| weighted avg | 0.82 | 0.82 | 0.82 | 2400 |

**4. Resultados**

-Para asegurar un buen desempeño en los algoritmos se debe garantizar ciertas condiciones de preparación de los datos. Estas condiciones ayudan a minimizar el 'ruido' y a aumentar la pureza y precisión de los datos. Algunas de estas condiciones son: eliminación de stopwords, eliminación puntuación, eliminación de caracteres que no son ASCII, normalización y tokenización.

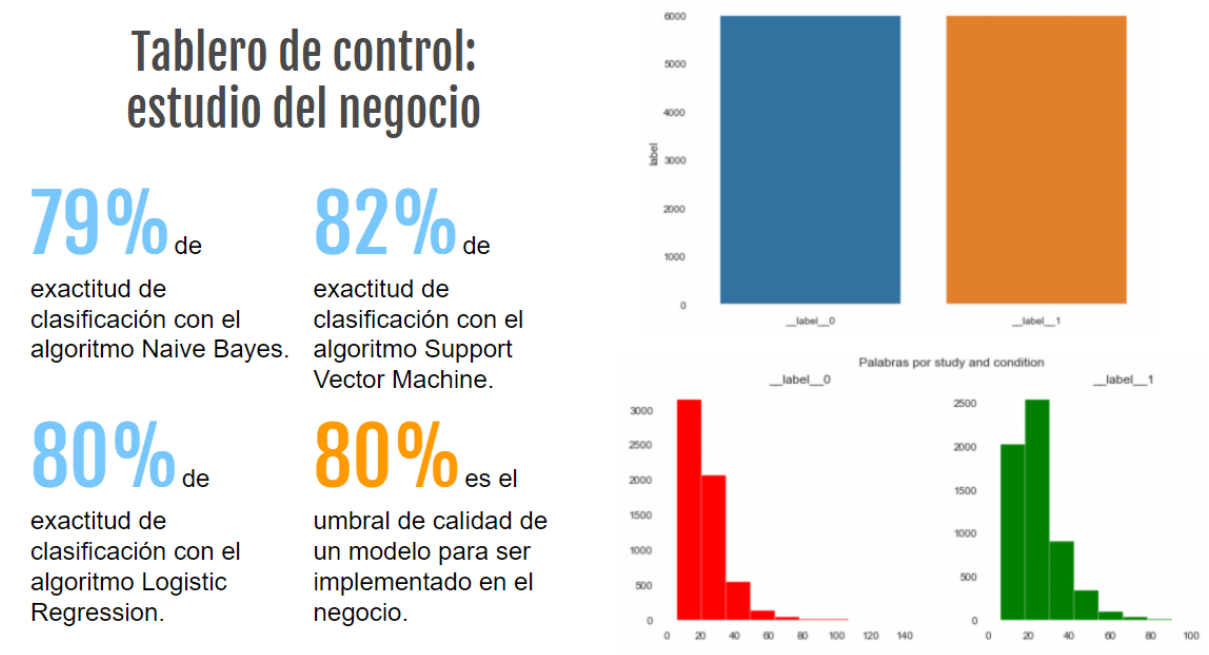
-El mejor resultado de cada algoritmo se dio usando la columna combinada study_condition. En términos de accuracy, la Regresión Logística obtuvo un desempeño del 80%, Naive Bayes del 79% y Support Vector Machine del 82%.

-Entre los tres algoritmos usados, el de mayor utilidad, dada la naturaleza de los datos y la preparación realizada, corresponde a Support Vector Machine con una exactitud del 84%.

-Para asegurar un agregación de valor al negocio se estableció un umbral de calidad de desempeño del modelo de clasificación para poder ser acoplado al negocio. Este umbral de calidad es que la exactitud sea igual o mayor a 80%. Bajo esta condición, se le sugiere al negocio implementar el modelo de clasificación usando el algoritmo de Support Vector Machine.

-Dada las consecuencias de una posible mala clasificación, sugerimos implementar un segundo modelo que reclasifique los pacientes elegibles del SVM, para evitar los falsos positivos

---




**Trabajo en equipo**

<u>Retos</u>

Algunos de los retos fueron: primero, la acumulación de entregas de otras materias junto a esta, lo cual impedía una dedicación absoluta al proyecto. Segundo, puesto

que durante semana de receso cada estudiante tiene un horario distinto al usual, no era fácil acordar una hora de reunión y cumplirla. Tercero, fue un reto positivo el hecho de tener que buscar qué algoritmos usar, cómo preparar los datos para un desempeño óptimo y todo esto por nuestra cuenta. Cuarto, el hecho de que en muchas ocasiones teníamos que tomar decisiones de diseño de la entrega.

Aporte por integrante
-Simón Rendón (líder de proyecto, líder de datos): presentación, Support Vector Machine, conclusiones, informe. Tiempo requerido 6 horas.
-Manuel Sosa (líder de negocio, líder de datos):preparación de los datos, presentación, Naive Bayes, conclusiones, tablero de control, informe. Tiempo requerido 6 horas.
-Juan David Díaz (líder de analítica, líder de datos): preparación de los datos, presentación, regresión logística, resultados, informe. Tiempo requerido 7 horas.

Distribución sobre 100 por estudiante (por acuerdo común)
-Simón Rendón: 33
-Manuel Sosa: 33
-Juan David Díaz: 34

| Reunión | Objetivo | Fecha |
|---|---|---|
| Lanzamiento y planeación | Definir roles y forma de trabajo del grupo. Establecer la comprensión y enfoque analítico. | 22 marzo 2022 |
| Ideación | En consenso, determinar las estrategias de preparación de los datos así como los modelos a utilizar por cada integrante. | 22 marzo 2022 |
| Seguimiento | Revisar el avance de los modelos y los resultados. (Reunión Zoom) | 24 marzo 2022 |
| Finalización | Consolidar los modelos y resultados en un solo Notebook así como definir el documento y la presentación. | 28 marzo 2022 |

**Link del repositorio en GitHub:**

https://github.com/mr-sosa/BI-Proyecto1.git

## 5. Referencias


- **Rani, Vijaya (2021). NLP-text-classification-model (commit 007433). https://github.com/vijayaiitk/NLP-text-classification-model/blob/main/NLP%20text%20classification%20model%20Github.ipynb**
- ***IBM Docs*. (s. f.). IBM - Deutschland | IBM. https://www.ibm.com/docs/es/spss-statistics/beta?topic=regression-binary-logistic**
- **Roman, V. (2019, April 25). Algoritmos Naive Bayes: Fundamentos e Implementación. Medium; Ciencia y Datos. https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f**