**Universidad de los Andes**

**Departamento de Ingeniería**

**Facultad de Ingeniería de Sistemas y Computación**

**Inteligencia de Negocios**

**Informe de Proyecto #1:**

**Análisis de textos**

**Determinar la elegibilidad de un paciente para ensayos clínicos de cáncer a partir de texto descriptivo**

**Por:**

**Carlos Figueredo - 201813445**

**Carlos Ballén - 200821488**

**Simón Guzmán - 201533711**

**Tabla de contenidos:**

**Parte 1:** Comprensión del negocio y enfoque analítico. **Página 3**

**Parte 2:** Comprensión de los datos y preparación de los datos**. Página 3**

**Parte 3:** Modelado y evaluación. **Página 4**

**Parte 4:** Análisis de los resultados.  **Página 5**

**Parte 5:** Trabajo en equipo.  **Página 6**

**Parte 6:** Referencias  **Página 7**

**Comprensión del negocio y enfoque analítico:**

| Oportunidad/problema Negocio | El negocio tiene datos con textos que describen los estudios y las condiciones de los pacientes. Estos textos tienen asignados si los pacientes son elegibles para ensayos clínicos de cáncer. | |
| --- | --- | --- |
| Descripción del requerimiento desde el punto de vista de aprendizaje de máquina | Analizar los textos de forma que se relacione la ocurrencia de las palabras en los estudios y condiciones con la etiqueta de elegibilidad. | |
| Detalles de la actividad de minería de datos | | |
| Tipo aprendizaje | Tarea de aprendizaje | Algoritmo e hiper-parámetros utilizados (con la justificación respectiva) |
| Supervisado | Clasificación | Random-forest: Con n-estimadores igual a 1000. No se hizo búsqueda debido a la demora en la ejecución del proceso. |
| Supervisado | Clasificación | SVM con los hiper-parámetros de c=100, gamma = 1 y kernel: rbf los cuales se encontraron con un grind search para c= [0.1, 1, 10,1000], gamma =[1,0.1,0.01, 0.001, 0.0001] para kernel: rbf |
| Supervisado | Clasificación | Regresion Logisitca, Parámetros de label 0 y 1 en función del TF-IDF de las palabras analizadas. |

**Comprensión de los datos y preparación de los datos:**

Los datos tienen dos columnas, una con el nombre label y otra con el nombre study\_and\_conditions. La columna label tiene \_\_label\_\_1 si el paciente es elegible para diagnóstico y \_\_label\_\_0 de lo contrario. La columna study\_and\_conditions tiene el texto con la información que se tiene respecto a cada paciente.

A los datos se les realizó una eliminación de nulos y duplicados. Se les removieron los caracteres que no son de tipo ASCII, los signos de puntuación, las palabras que son consideradas stopwords. Se reemplazaron los dígitos por su representación textual y se pasaron todas las letras a minúsculas.

Posteriormente, se realizó la tokenización, pero primero unas correcciones en las contracciones de las palabras y eliminación del ruido.

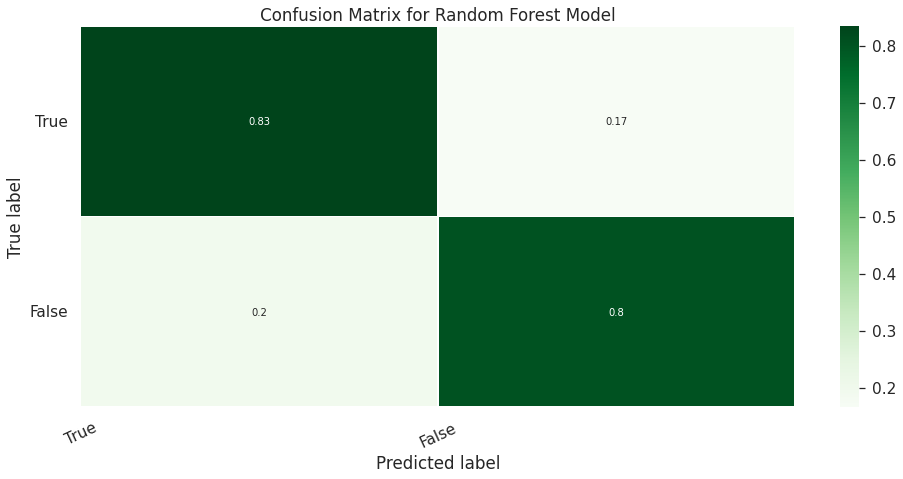
Ya después se pasó a la normalización de los textos, eliminando así prefijos y sufijos y realizando una lematización.

Para finalizar se convirtieron los valores de la columna label de \_\_label\_\_0 a 0 y de \_\_label\_\_1 a 1. Luego se realizó un Term-frecuency times inverse Document-frecuency (TF-IDF) a los datos para contar las ocurrencias de las palabras en los textos.

**Modelado y evaluación:**

1. **Random-Forest (Carlos Figueredo):**

El algoritmo Random Forest es un meta estimador que ajusta un número de clasificadores de árboles de decisión en varias submuestras del conjunto de datos y usa el promedio para mejorar la precisión predictiva y controlar el sobreajuste. El tamaño de las submuestras es controlado con el parámetro max\_samples si bootstrap=True(default), de otra manera el conjunto de datos completo es usado para construir cada árbol. Para la tarea de aprendizaje se decidió que el hiperparámetro n estimadores fuera igual a 1000, no se buscó el más óptimo debido a la complejidad temporal de la tarea. Al aplicar el algoritmo sobre los datos se obtuvo una exactitud del 81%



1. **Support Vector Machines (Simón Guzmán Londoño)**

El algoritmo de máquinas de vectores de soporte es llamado de esta forma ya que se genera un margen máximo entre los vectores de soporte que pasan por los puntos que definen la máxima separación entre las clases. Para este modelo, se tienen los parámetros C y Gamma que se desean optimizar, ya que en el caso de un kernel lineal solo es necesario optimizar C pero con un kernel RBF es necesario optimizar ambos. Asi, al usar grind search se obtiene que los parametros optimos son C = 100 y gamma = 1 para kernel:rbf

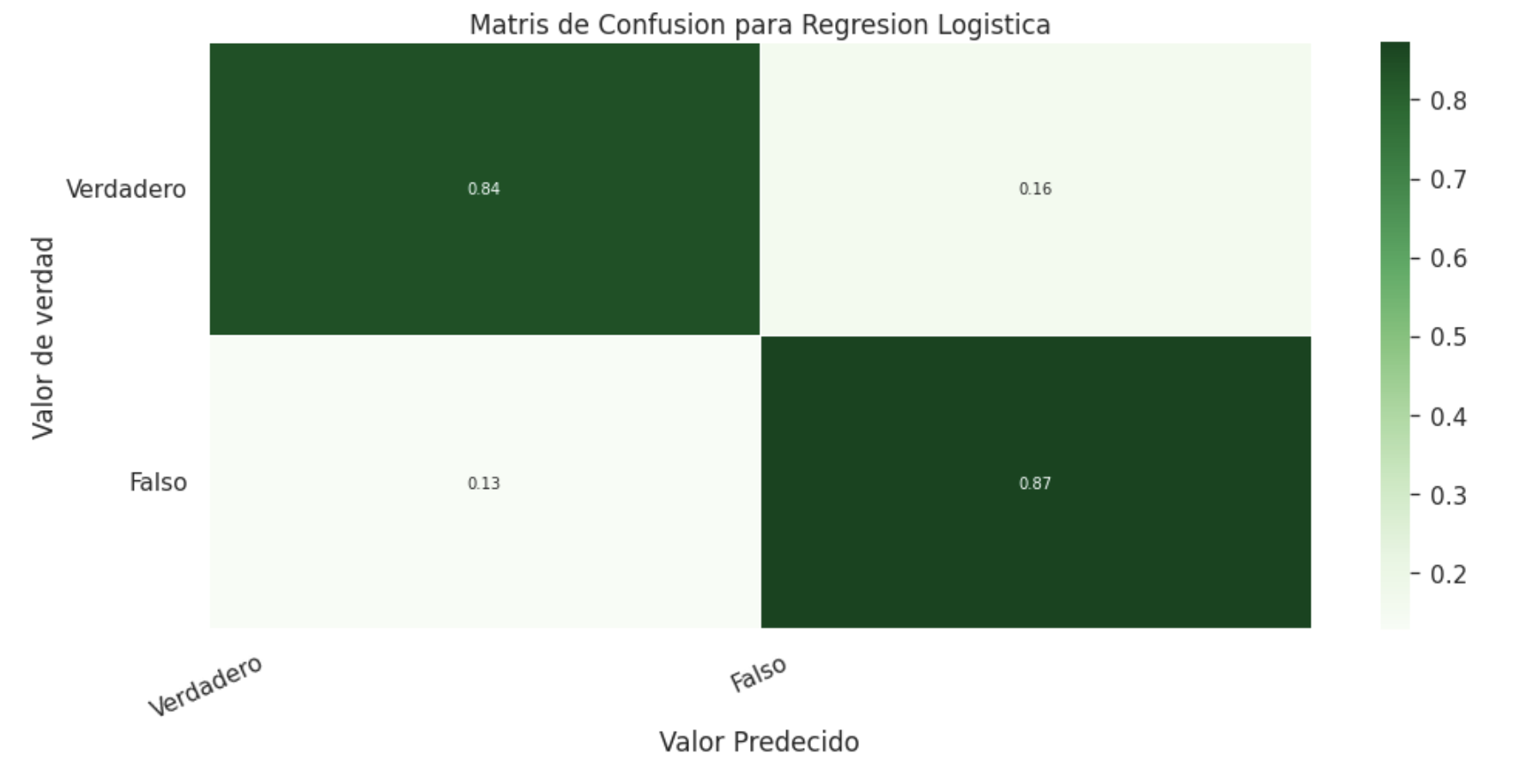


Con esto en mente, se prueba el modelo haciendo uso de los parámetros encontrados y se obtiene una precisión de 83% para saber si el paciente no es elegible y de 81% para el caso de que si sea elegible.

1. **Regresión Logística (Carlos Ballen)**

La regresión logística es un tipo de algoritmo de clasificación categórica, como lo puede ser en este caso el label 0 y 1 de los datos para determinar si un paciente es elegible o no para diagnóstico de cáncer. Por lo general esta variable se contrasta con múltiples variables independientes que para este caso son los datos de las palabras del diagnóstico de la columna study\_and\_conditions. Se utilizó este algoritmo ya que nos permite determinar probabilísticamente el evento de si un paciente es elegible con base en los factores determinados por su estudio y condición médica.

Con los datos trabajados el algoritmo logró una exactitud del 86% con una precision del 84% de que sea si elegible para diagnóstico de cáncer.

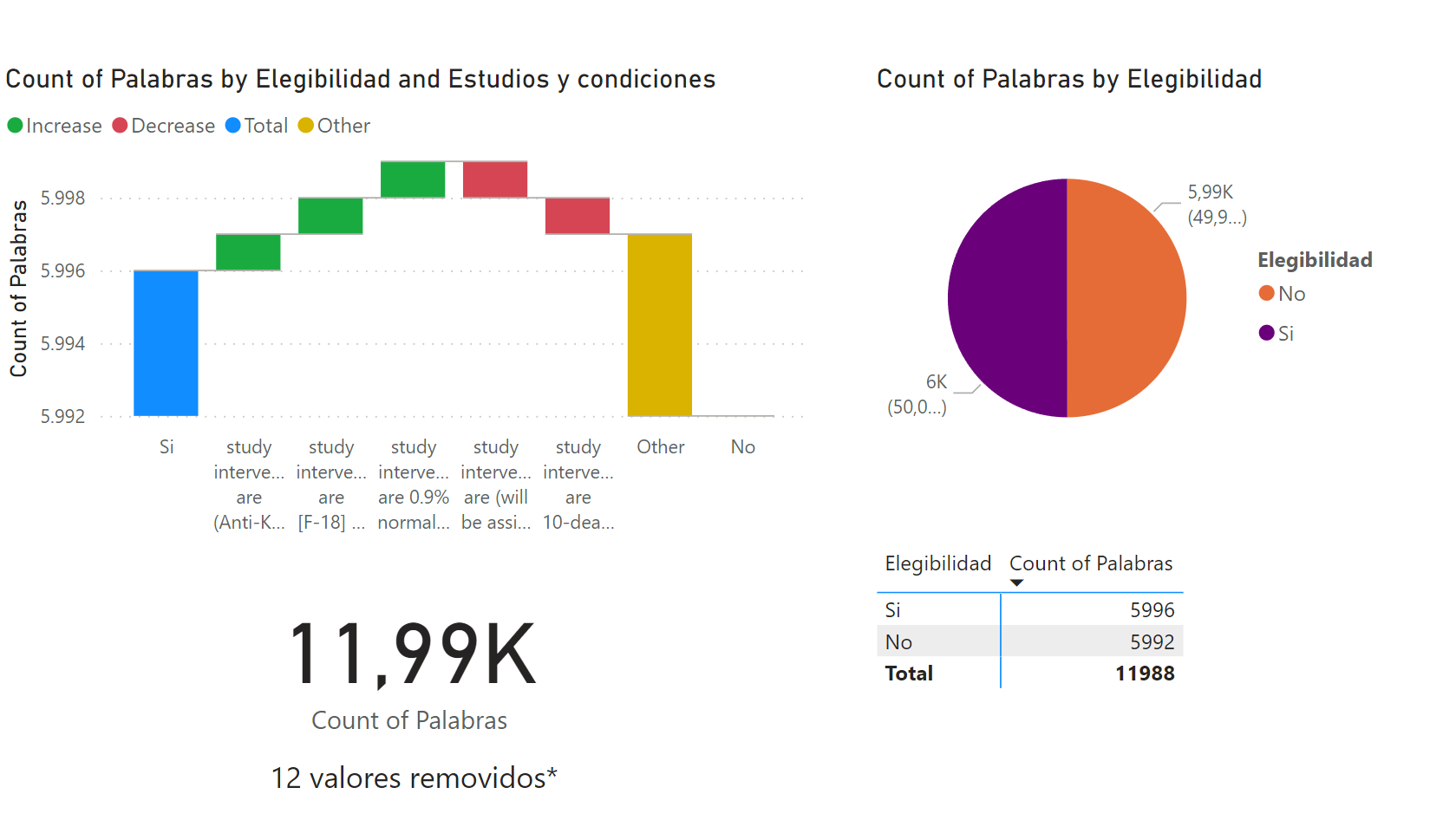


**Resultados:**

*Descripción y visualización en un tablero de control, de los resultados obtenidos, que permita a la organización comprenderlos, haciendo énfasis en el análisis de las medidas arrojadas por los modelos utilizados y cómo aportan en la consecución de los objetivos del negocio. Incluir posibles estrategias que la organización debe plantear relacionadas con los resultados obtenidos en los modelos y una justificación de por qué esa información es útil para ellos. Este resultado debe estar en el documento y deben generar un video y una presentación de máximo 7 minutos explicando su proyecto y resultados.*

Después de obtener los resultados de los tres algoritmos, se consideró que para el negocio lo más conveniente sería el uso del algoritmo de regresión logística, puesto que obtuvo la mayor exactitud. Una alta exactitud es importante ya que, según la naturaleza del negocio, se debe fallar lo menos posible a la hora de diagnosticar a un paciente.

Asimismo, considerando el contexto se considera que un falso negativo es más perjudicial ya que conlleva a tener una elegibilidad negativa cuando esta es necesaria para el paciente. Mientras que, si bien un falso positivo no es deseable, se le recomendaría al negocio implementar un paso adicional para filtrar estos casos.



**Trabajo en equipo:**

Durante el desarrollo del proyecto tuvimos que investigar sobre diferentes métodos para poder implementar limpieza a los datos, algunos como stemming, lemmatization, entre otros.

**Puntajes asignados (100 para asignar):**

Carlos Figueredo (Líder de Proyecto y Líder de Datos)

Tareas: Limpieza de datos, alistamiento de documentos, algoritmo Random Forest.

Horas destinadas al proyecto: 11 horas

Puntaje: 40

Carlos Ballén (Líder de Negocio)

Tareas: Algoritmo Regresión Logística, Grabación de Video, Edicion de Video documentación.

Horas destinadas al proyecto: 6 horas

Puntaje: 30

Simón Guzmán (Líder de Analitica)

Tareas: Algoritmo SVM, Grabación de Video, documentación, Tablero de control.

Horas destinadas al proyecto: x horas

Puntaje: 30

**Repositorio de trabajo:**

<https://github.com/simonguzm997/Proyecto_1_BI>

**Referencias:**

Python – Lemmatization Approaches with Examples. Geeksforgeeks. Recuperado de: <https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/>

Stemming and Lemmatization in Python. Datacamp. Recuperado de: <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>

Text Classification with Python and Scikit-Learn. Stackabuse. Recuperado de: <https://stackabuse.com/text-classification-with-python-and-scikit-learn/>

TF-IDF | TF-IDF Python Example. Stackabuse. Recuperado de: <https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76>

Regresión Logística para Python. Recuperado de:

<https://www.aprendemachinelearning.com/regresion-logistica-con-python-paso-a-paso/>