Proyecto 1 - Entrega 1: Analítica de textos

Laura Rodriguez - 201816069
Jaime Carvajal - 201632567
John Guzmán - 201713338

El vídeo se puede consultar en este enlace y el repositorio de GitHub en este.

Introducción

Los tratamientos interventores de cáncer tienden a ser demasiado restrictivos para muchos pacientes y muchos de estos además son excluidos por comorbilidades, tratamientos anteriores, o por edad. Esto quiere decir que la seguridad de estos pacientes al recibir estos tratamientos no está definida. Este proyecto se realiza con la idea de obtener una herramienta de predicción ideal que, utilizando documentos clínicos cortos, predijera si un paciente sería incluido o excluido de estos tratamientos, y así expeditar el proceso de tratamiento de cáncer.

Trabajo en Equipo

En la tabla a continuación se presenta las distribución de los roles por cada uno de los integrantes:

Roles de los Integrantes del Equipo		
Nombre del Integrante	**Tipo de Rol**	**Importancia y Justificacion del Rol**
John Alexander Guzman	Líder de Datos	Se encarga de gestionar los datos que se van a usar en el proyecto y de las asignaciones de tareas sobre datos. Debe dejarlos disponibles para todo el grupo.
Jaime Andres Carvajal	Líder de Analítica	Se encarga de gestionar las tareas de analítica del grupo. Se encarga de verificar que los entregables cumplen con los estándares de análisis y que se tiene el "mejor modelo" según las restricciones existentes.
Laura Andrea Rodriguez	Líder de Negocio	Es responsable de velar por resolver el problema o la oportunidad identificada y estar alineado con la estrategia del negocio para el cual se plantea el proyecto. Debe garantizar que el producto se puede comunicar de forma apropiada.
	Líder de Proyecto	Está a cargo de la gestión del proyecto. Define las fechas de reuniones, pre-entregables del grupo y verifica las asignaciones de tareas para que la carga sea equitativa. Se encarga de subir la entrega del grupo. Si no hay consenso sobre algunas decisiones, tiene la última palabra.

Comprensión del Negocio y Enfoque Analítico

En términos de aprendizaje de máquina, lo que se requiere obtener es la clasificación de pacientes entre dos categorías: Incluido y excluido. Se plantea que el objetivo que se esta buscando con el análisis de Machine Learning, es determinar la elegibilidad de un paciente para ensayos clínicos de cáncer a partir del texto descriptivo, esto se debe a que se busca optimizar y acompañar el programa de selección de los pacientes, de forma que se ayude a determinar ciertos criterios que ayuden a definir la eficacia y seguridad de nuevos tratamientos para pacientes de estas características.

Por lo tanto, se plantea que este es un problema de aprendizaje supervisado y de clasificación. Esto se debe a que ya de por si los datos están etiquetados, lo que permite hacer el entrenamiento más fácil y eficiente, y por el otro lado, es un problema de clasificación binaria, pues vamos a definir si es el paciente es elegible o no.

A continuación se presenta una tabla que explica de mejor forma el objetivo del negocio:

Oportunidad/Problema del Negocio		Los ensayos clínicos de cáncer intervencionista suelen ser demasiado restrictivos, y algunos pacientes suelen ser excluidos en función de la comorbilidad, los tratamientos previos o concomitantes, o el hecho de que son mayores de cierta edad. La eficacia y seguridad de nuevos tratamientos para pacientes de estas características no están, por tanto, definidas
Descripcion del requerimiento desde el punto de vista de aprendizaje de maquina		Se plantea que el objetivo que se esta buscando con el analisis de Machine Learning, es determinar la elegibilidad de un paciente para ensayos clínicos de cáncer a partir del texto descriptivo, esto se debe a que se busca optimizar y acompañar el programa de selección de los pacientes, de forma que se ayude a determinar ciertos criterios que ayuden a definir la eficacia y seguridad de nuevos tratamientos para pacientes de estas características. Por lo tanto, se plantea que este es un problema de aprendizaje supervisado y de clasificacion. Esto se debe a que ya de por si los datos están etiquetados, lo que permite hacer el entrenamiento más fácil y eficiente, y por el otro lado, es un problema de clasificación binaria, pues vamos a definir si es el paciente es elegible o no.
Detalles de la actividad de mineria de datos		
Tipo de Aprendizaje	**Tarea de Aprendizaje**	**Algoritmo e hiper-parametros utilizados (poner justificacion respectiva)**
Aprendizaje Supervisado	Clasificación	El algoritmo utilizado es arboles de clasificación, este algoritmo se escogió pues es ayuda a definir las caracteristicas mas importantes al momento de hacer la predicción, en este sentido, se va a usar el arbol de clasificación para poder entender que palabras son posiblemente las más relevantes al momento de tomar una decisión si el paciente es elegible o no. Los hiperparametros utilizados, fueron encontrados haciendo uso de validacion cruzada con GridSearchCV
Aprendizaje Supervisado	Clasificación	El algoritmo utilizado es support vector machine. Las ventajas de utilizar una SVM lineal incluyen la velocidad de entrenamiento y el hecho de que el único parámetro que se debe optimizar es la C-regularización. Además, cuando se comparan con otros algoritmos de clasificación, como redes neuronales, tienden a ser más rápidos y a tener mejores resultados cuando la cantidad de datos es relativamente pequeña (No pasa de los 20,000). Teniendo en cuenta que se está trabajando con 12,000 datos, una SVM lineal podría ser la forma ideal de clasificación para este problema.
Aprendizaje Supervisado	Clasificación	El algoritmo utilizado es K-Nearest Neighbors. Se decidió poner a prueba este algoritmo bajo el contexto especifico de este problema, con el fin de identificar si es posible determinar acertadamente si un determinado paciente es elegible o no. Con el fin de determinar los mejores hiperparámetros del modelo se utilizó la búsqueda con validación cruzada con GridSearchCV.
Aprendizaje Supervisado	Regresión	El algoritmo utilizado es la Regresión Logística. Pero por la forma específica de la regresión logística es de clasificación. Es fácil de implementar y se puede utilizar como base para cualquier problema de clasificación binaria. Y para este caso en específico da muy buenos resultados.

Comprensión y preparación de los datos

En cuanto al perfilamiento de los datos, primero se hizo una separación entre los datos del estudio y los datos de tipo de cáncer de la columna study_and_condition, esto para poder identificar como influia el estudio de los pacientes y sus diferentes tipos de cancer. De aquí se hizo una agrupación por el tipo de estudio, y se

buscó cual era la distribución de los pacientes elegidos por estudio y se encontró lo siguiente:

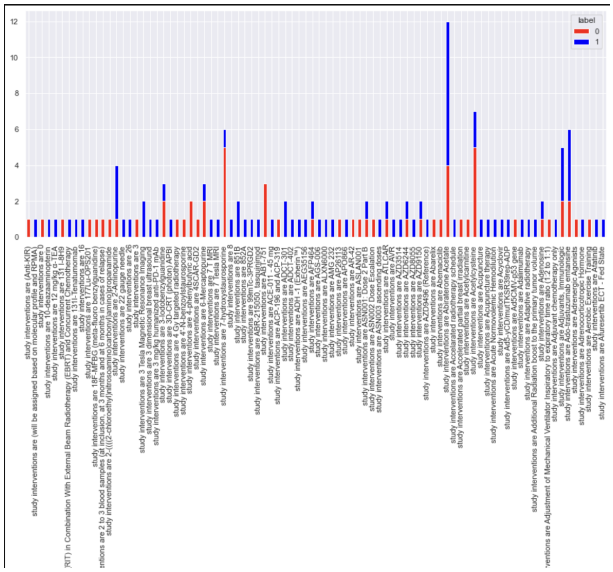

Figura 1: Distribución de acuerdo al estudio

De igual forma se dividieron los datos por dos tipos diferentes de cancer *clintrial_lymphoma* y *clintrial_breast*, y se encontró que las palabras que posiblemente podían tener mayor influencia eran las siguientes:recurrent, stage_ii, stage_iii, stage_iv, follicular, diffuse, hodgkin.

Ahora bien, en cuanto a la preparacion de los datos, dado que estos estaban en formato texto, tuvo que pasar por tres etapas distintas:

- Limpieza de los datos: para esto se definieron diferentes metodos que removian caracteres que no pertenecian a ASCII, se removio la puntuacion, se reemplazaron los numeros, y se removieron las llamadas "stopwords".
- Tokenización: En este punto se aplicaron los metodos que se hicieron en limpieza, y se hizo la division del texto en un arreglo de tokens.
- Normalización: finalmente, se realiza la eliminación de prefijos y sufijos, además de realizar una lemmatización

Modelado y Evaluación de Resultados

Se realizó la aplicación de cuatro algoritmos diferentes para la tarea de aprendizaje de máquina seleccionada. En esta parte se van describir los algoritmos utilizados, haciendo una justificación del porque se uso cada uno de estos modelos, así como se describirá el algoritmo para la construcción de los modelos y se hará una presentación de los resultados de la evaluación cuantitativa

1. Árbol de decisión

1.1. Justificación

Los arboles de decisión, son uno de los algoritmos de aprendizaje supervisado más utilizados en machine learning y pueden realizar tareas de clasificación o regresión. En este sentido, los arboles de decisión son representaciones gráficas de posibles soluciones a una decisión basadas en ciertas condiciones.

Para obtener el árbol óptimo y valorar cada subdivisión entre todos los árboles posibles y conseguir el nodo raíz y los subsiguientes, el algoritmo deberá medir de alguna manera las predicciones logradas y valorarlas para comparar de entre todas y obtener la mejor. Para medir y valorar, utiliza diversas funciones, siendo las más conocidas y usadas los "Índice gini" y "Ganancia de información" que utiliza la denominada "entropía". La división de nodos continuará hasta que lleguemos a la profundidad máxima posible del árbol ó se limiten los nodos a una cantidad mínima de muestras en cada hoja

Teniendo en cuenta lo anterior, este algoritmo se escogió, pues es ayuda a definir las caracteristicas mas importantes al momento de hacer la predicción, en este sentido, se va a usar el árbol de clasificación para

poder entender que palabras son posiblemente las más relevantes al momento de tomar una decisión si el paciente es elegible o no.

1.2. Modelo

Ahora bien, para lograr el objetivo anteriormente mencionado, haciendo uso de una validación cruzada, que divide en k particiones disjuntas el conjunto de datos de entrenamiento. Luego, se fija un valor de hiper parámetro y se toma como conjunto de validación la primera partición y el resto para entrenar el modelo. Este proceso se repite para cada partición. Al finalizar, se determina el valor promedio de las métricas seleccionadas. Este ciclo se repite sobre diferentes valores del hiper parámetro y se seleccionan aquel que ofrezcan el mejor rendimiento.

Scikit-learn ofrece algunos métodos que automatizan el proceso de buscar los valores de los hiper parámetros. Uno de ellos es GridSearchCV, el cual se basa en la validación cruzada de k-particiones. Por lo tanto, a continuación, se hace uso de este algoritmo, en función de encontrar los mejores hiper parámetros para la construcción de nuestro árbol de decisión. De esto, se obtuvo que el mejor modelo está definido por:

$arbol\_final = DecisionTreeClassifier('criterion' :' gini','max\_depth' : 6,'min\_samples\_split' : 5)$

1.3. Evaluación de Resultados

Ahora bien, como resultados se encontró la siguiente matriz de confusión para los datos de prueba:

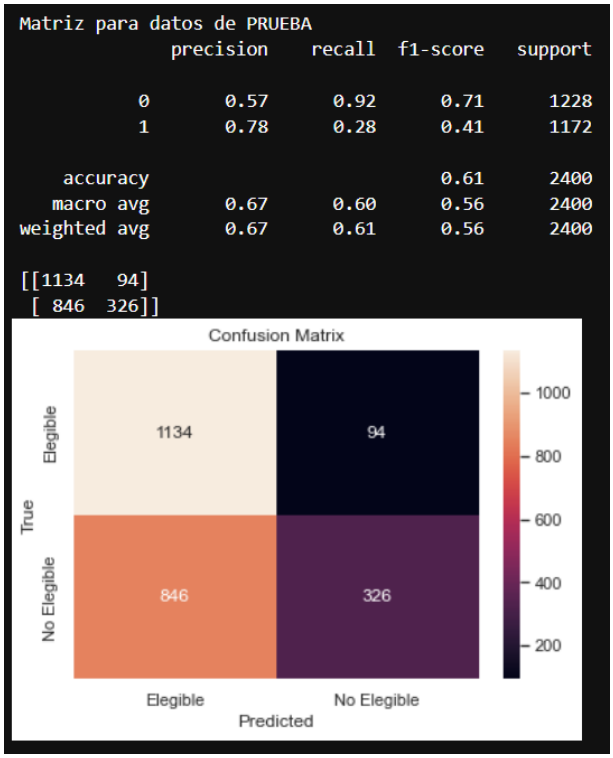

Figura 2: Matriz de confusión del modelo de Arboles de Decisiónl

Como se puede observar, aquí la clasificación también es buena, pues tiene una exactitud 0.61. En cuanto a la precisión de la clasificación de cada una de las clases se puede decir que el modelo lo hace correctamente para los datos de la clase 1, sin embargo no clasifica tan bien los datos de la clase 0. Esto quiere decir que, los datos que clasifica como verdaderos positivos respecto a el total de datos que clasifica como positivos es bastante bueno para la clase 1 (pacientes no elegidos). Adicionalmente la sensibilidad de aquellos datos que clasifica como verdaderos con respecto al total que son verdaderos, es muy buena también para la clase 0, pues tiene un recall de 0.92, pero para la clase 1 lo hace no muy bien (0.28).

Adicionalmente, se intento construir un árbol para poder identificar las palabras que posiblemente podían influir más al momento de construir el modelo, y se obtuvo lo siguiente:

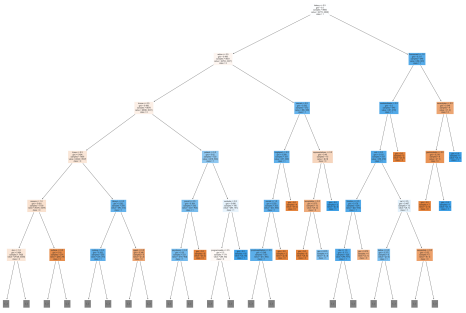

Figura 3: Arbol de Decisiónl

Ahora bien, en cuanto a esto ultimo, se puede observar que no hay palabras fundamentalmente importantes, pues las que aparecen como palabras mas determinantes no son fundamentales para la caracterización del cáncer.

2. Support Vector Machine

2.1. Justificación

Se puede utilizar una SVM siempre y cuando se tengan exactamente dos clases. En esta situación se observan estas dos clases en la forma de 'incluido' y 'excluido', por lo que es ideal. Una SVM funciona de tal manera que encuentra el mejor "hiper-plano"que separa todos los datos que pertenecen a una clase de todos aquellos que no. Funciona por medio de vectores, por lo que es necesario transformar los datos en texto a valores de este tipo. En especifico para problemas con texto, que tienen una gran cantidad de rasgos diferentes, es recomendable utilizar una SVM de kernel lineal, lo que quiere decir que el hiper-plano que se utilizará para separar los datos es una línea recta.

Las ventajas de utilizar una SVM lineal incluyen la velocidad de entrenamiento y el hecho de que el único parámetro que se debe optimizar es la C-regularización. Además, cuando se comparan con otros algoritmos de clasificación, como redes neuronales, tienden a ser más rápidos y a tener mejores resultados cuando la cantidad de datos es relativamente pequeña (No pasa de los 20,000). Teniendo en cuenta que se está trabajando con 12,000 datos, una SVM lineal podría ser la forma ideal de clasificación para este problema.

2.2. Modelo

De los hiperparámetros que se pueden cambiar para una SVM en sklearn.svm.LinearSVC, el único que tienen un efecto en nuestro problema es la C-regularización. Teniendo este hecho en cuenta, se llevó a cabo un experimento para encontrar el mejor valor de C posible para los datos que se tienen. Para entender cuál es el mejor valor para la C-regularización, es necesario entender lo que este hiperparámetro es en general, por lo que se dará una corta explicación a continuación.

2.2.1. C-regularización

En una SVM se intenta encontrar dos cosas: un hiper-plano que separe instancias de la forma más correcta posible, y un hiper-plano con el más alto margen mínimo posible. Estos dos factores, sin embargo, no siempre van de la mano, y por esta razón existe la C-regularización. Esta le indica al algoritmo que tan importante es obtener las instancias de la forma más correcta posible, en comparación con la importancia del margen. Entre mayor sea el valor de C, más importancia se le dará a clasificar las instancias de la forma más correcta posible.

Hay casos en los que clasificar las instancias de la forma más correcta posible puede ser contraproducente, reduciendo la posibilidad de que nuevos datos sean clasificados correctamente si no son extremadamente parecidos a los datos que se han obtenido anteriormente. Esto quiere decir que un C alto es ideal si se espera que los valores nuevos sean muy similares a los que ya se han obtenido, mientras que un C bajo es ideal si se espera obtener nuevos datos que no sean extremadamente similares a los ya obtenidos pero que requieran ser clasificados junto con ellos.

Para este caso especifico, la métrica principal a utilizar será el recall que se consiga con cada valor de C, ya que este nos ayuda a asegurar que pacientes que clasifican para el tratamiento lo obtengan, hasta si esto quiere decir que algunos que no clasifican pasan también. Esta forma de medición indica que se puede esperar que un valor bajo de C-regularización sea el indicado. De forma secundaria, también se busca un valor lo más alto posible de precisión, ya que también es importante asegurar que los pacientes incluidos no hagan parte del grupo que debía ser excluido.

Teniendo en cuenta los factores anteriormente mencionados, se llevó a cabo un procedimiento de optimización del valor de C, buscando maximizar el valor de f-β, donde beta tiene un valor de 2, para conseguir así el modelo que mejor cumpla con las expectativas del negocio. También se llevó a cabo el procedimiento tres veces distintas, una con los datos del cáncer, una con los datos del estudio, y una con todos los datos, para así escoger la opción adecuada.

2.3. Evaluación de Resultados

Los resultados de la búsqueda del mayor f-β se encuentran tabulados a continuación.

Datos	Valor de f-β
Cáncer	0.806
Estudio	0.568
Ambos	0.803

Cuadro 1: Máximos valores obtenidos de f-β para un valor de C entre 1 y 15 para cada grupo de datos

Estos valores nos demuestran que la mejor opción para SVM es utilizar los datos exclusivamente del cáncer, aunque utilizar ambos datos tiene un valor casi igual. Al buscar cuál valor de C obtuvo el mayor resultado de f-β, se descubre que este pasa cuando C=1, lo que concuerda con la hipótesis inicial de que el valor sería bajo para reducir la posibilidad de que personas que cumplan con los prerrequisitos para obtener tratamiento no lo obtengan, inclusive si unas pocas más de las que no cumplen terminen obteniéndolo.

Para la SVM ideal encontrada, se muestra la matriz de confusión obtenida a continuación.

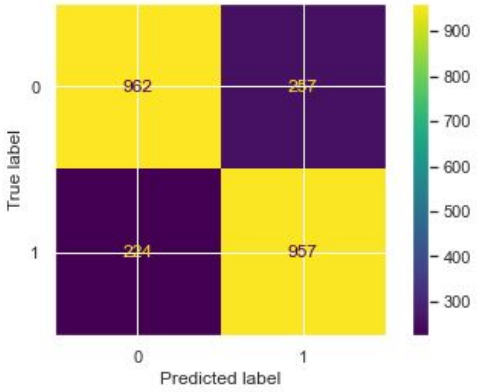

Figura 4: Matriz de confusión de SVM ideal

Los mejores valores calculados para SVM lineal se muestran a continuación.

Resultados de predicciones con datos de prueba

	precision	recall	f1-score	support
0	0.77	0.78	0.78	1219
1	0.77	0.76	0.77	1181
accuracy			0.77	2400
macro avg	0.77	0.77	0.77	2400
weighted avg	0.77	0.77	0.77	2400

Figura 5: Valores de medida obtenidos parala SVM optimizada

3. Knn

3.1. Justificación

El algoritmo de KNN es uno de clasificación supervisada, el cual se usa para para estimar la probabilidad que tiene un elemento de pertenecer a cierta clase según sus características dadas. En el reconocimiento de patrones, este algoritmo es usado como un método de clasificación basado en entrenamiento mediante ejemplos cercanos al espacio de elementos. Se decidió poner a prueba este algoritmo bajo el contexto especifico de este problema, con el fin de identificar si es posible determinar acertadamente si un determinado paciente es elegible o no.

3.2. Modelo

Con el fin de determinar la probabilidad que tiene un nuevo paciente de ser elegible o no, se decidió trabajar sobre los datos en conjunto, de este modo se puede llegar a una mejor estimación para el objetivo del modelo. Una vez se definió el conjunto de datos, se realiza una partición de los datos en un conjunto de entrenamiento y en un conjunto de prueba, con estos conjuntos definidos se procede a buscar los mejores hiperparámetros para el problema especifico, para esto se define un espacio de búsqueda para k entre 1 y 11, y posteriormente, con ayuda de GridSearchCV, se define cual es el mejor modelo. Finalmente, tras las búsqueda de los hiperparámetros, se encontró que el valor de k óptimo es 1, por lo que se construye el modelo tomando esto en cuenta.

3.3. Evaluación de Resultados

Al entrenar el modelo con el conjunto de datos correspondiente y hacer las respectivas pruebas sobre los datos, obtenemos los siguientes resultados:

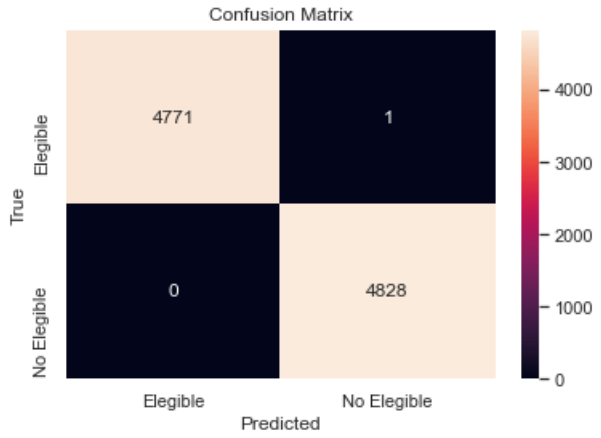

Figura 6: Matriz de confusión sobre los datos de entrenamiento

Matriz para datos de PRUEBA

	precision	recall	f1-score	support
0	0.69	0.81	0.74	1228
1	0.75	0.61	0.67	1172
accuracy			0.71	2400
macro avg	0.72	0.71	0.71	2400
weighted avg	0.72	0.71	0.71	2400

Figura 7: Resultados sobre los datos de prueba

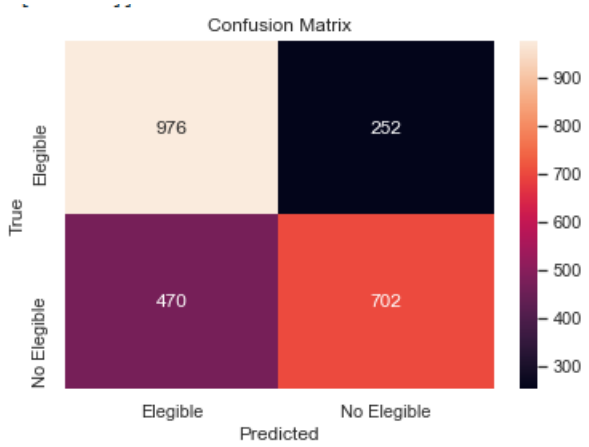

Figura 8: Matriz de confusión sobre los datos de prueba

Después de entrenar el modelo y establecer los mejores hiperparámetros del mismo, vemos que este tiene una exactitud sobre el conjunto de datos de entrenamiento de prácticamente el 100 %, lo que indica que el modelo aprendió casi a la perfección el conjunto de datos sobre el que fue entrenado. Una vez se tiene entrenado el modelo, se procede a probarlo con la partición de los datos correspondiente, en general, se tiene una exactitud aceptable del 70 %. Sin embargo, como se ve en la matriz mostrada anteriormente, la mayor parte de los datos predecidos erróneamente son lo que se consideran falsos-positivos, teniendo que la mayor parte de estos errores dan como elegibles a pacientes que en realidad no lo son.

4. Regresión Logística

4.1. Justificación

La regresión logística es uno de los algoritmos de aprendizaje automático más simples y comúnmente utilizados para la clasificación de dos clases. Es fácil de implementar y se puede utilizar como base para cualquier problema de clasificación binaria.

La regresión logística es un método estadístico para predecir clases binarias. La variable de resultado u objetivo es de naturaleza dicotómica. Dicotómico significa que solo hay dos clases posibles. Es un caso especial de regresión lineal donde la variable objetivo es de naturaleza categórica. Utiliza un registro de probabilidades como variable dependiente. La regresión logística predice la probabilidad de ocurrencia de un evento binario utilizando una función sigmoide.

Para esto, sabemos que la Regresión logística es útil pues, comparado con la regresión lineal, que brinda una salida continua, la regresión logística proporciona una salida constante, y para este caso, necesitamos hacer un modelo predictivo que determine si un pacientes es o no elegible para el tratamiento de cáncer especifico.

4.2. Modelo

Ahora bien, para lograr el objetivo anteriormente mencionado, haciendo uso de una validación cruzada, que divide en k particiones disjuntas el conjunto de datos de entrenamiento. Luego, se fija un valor de hiper parámetro y se toma como conjunto de validación la primera partición y el resto para entrenar el modelo. Este proceso se repite para cada partición. Al finalizar, se determina el valor promedio de las métricas seleccionadas. Este ciclo se repite sobre diferentes valores del hiper parámetro y se seleccionan aquel que ofrezcan el mejor rendimiento. Esto dio como resultado que el mejor modelo es:Comprensio

$modelo\_final = LogisticRegression('C' : 0{,}1,' penalty' :' l2',' solver' :' liblinear')$

4.3. Evaluación de Resultados

Ahora bien, como resultados se encontró la siguiente matriz de confusión para los datos de prueba:

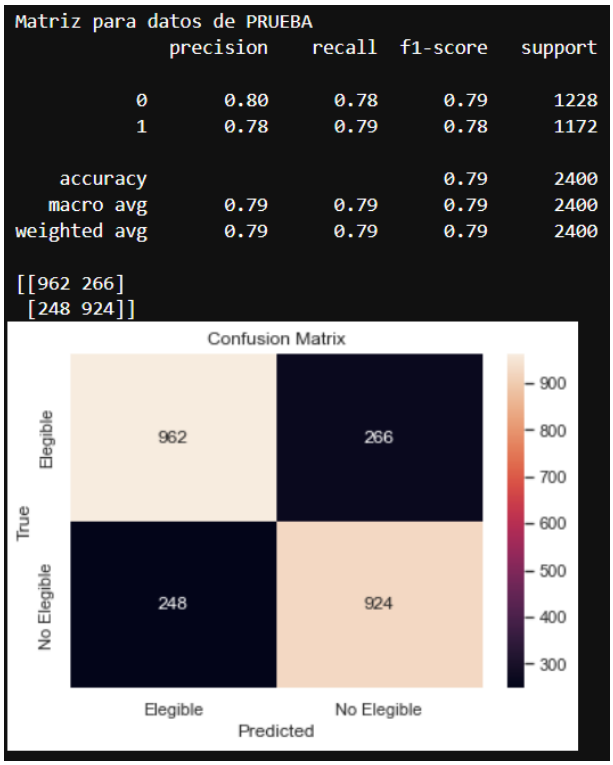

Figura 9: Matriz de confusión del modelo de Regresión Logística

Como se puede observar, la regresión es buena, pues tiene una exactitud de 0.79, y es de hecho la mejor entre todos los modelos realizados. En cuanto a la precisión de la regresión de cada una de las clases se puede decir que el modelo lo hace correctamente para los datos de entrenamiento para las dos clases, pues la precisión es de 0.80 y 0.78, ligeramente mejor para la clase 0 que para la 1. Es decir que los datos que clasifica como verdaderos positivos respecto a el total de datos que clasifica como positivos es bastante bueno para los pacientes que no son elegibles, y para aquellos que lo son. Adicionalmente la sensibilidad de aquellos datos que clasifica como verdaderos con respecto al total que son verdaderos es buena para ambas clases, pues tiene un recall de 0.78 y 0.79, respectivamente.

Conclusiones

- Finalmente se puede concluir que el modelo que hace la mejor predicción de la elegibilidad de un paciente dado el diagnostico es la Regresión Logística.
- Se observa que la precisión y el recall podrían incrementar si se llevara a cabo una estandarización en la metodología de escribir los informes médicos recibidos. Las nubes de palabras dan información acerca de las palabras cruciales a la hora de llevar a cabo la separación entre cada categoría, por lo que el cliente puede estandarizarlo hasta el punto que le sea conveniente.
- Sin importar la precisión que se encuentre en modelos como estos, estos siempre deberán ser utilizados como apoyo a la toma de decisiones médicas, no como un reemplazo, ya que decisiones tan cruciales como la de quién recibe tratamientos que podrían salvar su vida y quién no siempre deben tener un filtro humano en algún momento. Estas herramientas ayudarán a tomar la decisión más velozmente, pero no la harán por el médico.
- Se podría llevar a cabo investigaciones más a fondo con otros modelos de clasificación para encontrar alguno que sirva mejor para el caso especifico del cliente, buscando primero un recall y luego una precisión mayores a los obtenidos en esta investigación. Sin embargo, teniendo en cuenta el costo computacional que algunos modelos tienen, esto podría terminar siendo contraproducente, ya que siempre será necesaria la revisión humana y los valores obtenidos ya son suficientes para cumplir con las necesidades del cliente.

5. Referencias

[1]G. Bedi, "Simple guide to Text Classification(NLP) using SVM and Naive Bayes with Python", Medium, 2022. [Online]. Available: https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34. [Accessed: 25- Mar- 2022].

[2]S. Learn, "sklearn.svm.LinearSVC", scikit-learn, 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html. [Accessed: 27- Mar- 2022].

[3]B. Stecanella, "Support Vector Machines Algorithm Explained", MonkeyLearn, 2017. [Online]. Available: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/. [Accessed: 29- Mar- 2022].