**Proyecto Analítica de Textos**

**Caso: Elegibilidad de un paciente para ensayos clínicos.**

**Nicolas Orjuela, Camilo Salinas, Felipe Bedoya**

**Sección 1.**

Tabla de contenido

[**1.** **Comprensión del negocio y enfoque analítico.** 1](#_Toc99269400)

[**2.** **Comprensión de los datos y preparación de los datos** 2](#_Toc99269401)

[**3.** **Modelado y evaluación** 2](#_Toc99269402)

[**4.** **Resultados** 2](#_Toc99269403)

[**5.** **Trabajo en equipo** 2](#_Toc99269404)

# **Comprensión del negocio y enfoque analítico.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Oportunidad/problema Negocio** | | Se necesita determinar la elegibilidad de los pacientes para ensayos clínicos de cáncer a partir de textos descriptivos. | |
| **Descripción del requerimiento desde el punto de vista de aprendizaje de máquina** | | Escoger y entrenar un modelo que, a partir de la analítica de textos y un conjunto de textos descriptivos, pueda determinar la etiqueta de nuevos textos, siendo la etiqueta la elegibilidad para ensayos clínicos de cáncer. | |
| **Detalles de la actividad de minería de datos** | | | |
| **Tipo aprendizaje** | **Tarea de aprendizaje** | | **Algoritmo e hiper-parametros (con la justificación respectiva)** |
| Supervisado | Clasificación | | Naive Bayes ('CNB\_\_alpha': 1.1, 'CNB\_\_fit\_prior': True, 'tfidf\_\_ngram\_range': (1, 1), 'tfidf\_\_use\_idf': True) |
| Supervisado | Clasificación | | Support Vector Machines (‘C’: 1, ‘gamma’: 1, ‘kernel’: ‘poly’) |
| Supervisado | Regresión | | Logistic Regression ('clf\_\_C': 1.0, 'clf\_\_penalty': 'l2', 'clf\_\_solver': 'newton-cg', 'tfidf\_\_ngram\_range': (1, 1), 'tfidf\_\_use\_idf': True) |

Todos los hiperparámetros de los modelos fueron escogidos a través de un GridSearch que determinó cuales son los más apropiados para este caso en particular a partir de un conjunto de posibles hiperparámetros.

La escogencia de los hiperparámetros que debían ir en el GridSearch se basó en aquellos que más influenciaran los resultados y el comportamiento del modelo. En LogisticRegression por ejemplo uno de los hiperparámetros es el solver, que puede tener implicaciones de velocidad y efectividad dependiendo del tamaño y tipo de los datos.

# **Comprensión de los datos y preparación de los datos**

Al cargar los datos y analizar sus características, nos dimos cuenta de que estos no tienen nulos, pero están presentados de maneras distintas. Algunos comienzan con comillas, otros son espacios. Era necesario lograr entradas similares. Adicionalmente la información importante del paciente estaba ubicada después del primer punto. Además, vimos que los labels son categóricos, por lo que era necesario volverlos numéricos. Aunque un label encoder haría esta tarea con facilidad, se quería conservar la categoría implícita que ya traen.

Otras transformaciones realizadas a los textos fueron la partición de estos en tokens y lematizándolos. Finalmente se utilizó un modelo de Bag Of Words y tf-idf para identificar las palabras importantes. También se utilizó el corpus de nltk para quitar las palabras conectoras que generaran ruido. Estas transformaciones fueron añadidas a un pipeline para poder ajustar el modelo escogido por cada uno.

# **Modelado y evaluación**

Se escogió Naive Bayes debido a que este es un modelo que propone un acercamiento con un costo computacional bajo, ideal para ser puesto en un escenario de producción. Adicionalmente se comporta bien al momento de utilizarse para casos de análisis de textos. Se realizó un GridSearch para encontrar los mejores hiperparámetros para el modelo. El mejor modelo obtenido tuvo una exactitud de 0.78 frente a los datos de prueba.

Se escogió Support Vector Machines debido a que estos modelos son considerados unos de los más apropiados para la analítica de textos gracias a su manejo de hiperplanos en la clasificación de datos. Se realizó un GridSearch para encontrar los mejores hiperparámetros para el modelo. El mejor modelo obtenido tuvo una exactitud de 0.82 frente a los datos de prueba.

Se escogió Logistic Regression debido a que presentaba un punto de vista distinto utilizando una tarea de aprendizaje contrastante con las anteriores. Aunque finalmente se utilice como un clasificador dada la naturaleza de las etiquetas, sirve para hacer interpretaciones y comprender más a profundidad el comportamiento de los datos y que patrones siguen. Viendo unas métricas buenas, se podría decir que los datos siguen una regresión logística, aunque sea muy difícil poner en contexto numérico al Bag of Words generado y la amplia matriz con palabras con columnas. Para este algoritmo también se realizó un GridSearch con un conjunto de hiperparámetros que incluyen las dos penalidades posibles y un rango de coeficientes de regularización. También se incluye en el grid si se usa o no el IDF para ver esto como varía con el comportamiento frecuencial de los tokens. Esto se probó con dos solvers. El modelo más optimo tuvo como coeficiente de determinación R2 un valor de 0.84, junto con un valor de 0-39 en su Root-Mean-Square Error. La exactitud del modelo con los datos de prueba fue de 0.806.

# **Resultados**

Como se puede observar en el tablero de control,

# **Trabajo en equipo**

1. Roles:
   * Líder de proyecto: Felipe Bedoya
   * Líder de negocio: Camilo Salinas
   * Líder de datos: Felipe Bedoya
   * Líder de analítica: Nicolás Orjuela
2. Algoritmo trabajado:
   * Naive Bayes: Camilo Salinas
   * Support Vector Machines: Nicolás Orjuela
   * Logistic Regression: Felipe Bedoya
3. Retos enfrentados:
   * En el preprocesamiento, cuando se le pasan los datos al pipeline, se le pasa la variable objetivo (Y) y las variables independientes (X), pero el pipeline solo modificaba las variables independientes.
   * La ejecución del GridSearch del Support Vector Machines tomó aproximadamente 3 horas, por lo que era necesario tener un conjunto pequeño pero diverso de hiperparámetros para evitar tener que probar con nuevos conjuntos debido a la limitación de tiempo.
   * Uno de los retos fue poder exportar los datos preprocesados a un csv, puesto que el pipeline no retorna un dataframe común y corriente, sino que retorna una sparse matrix.
4. Soluciones planteadas:
   * Para el preprocesamiento, se partió el dataframe y se pasaron 2 series de datos (no dataframes), además, el encoder de los labels se realizó por fuera de los pipelines antes de entrenar los datos.
   * Se investigo acerca de los rangos y valores óptimos en donde los hiperparámetros funcionaban mejor para el SVM, por lo que solo fue necesario correr el GridSearch 2 veces.
   * La sparse matrix se convirtió en dataframe y se exportó a csv. Adicionalmente, hay que tener en cuenta el tamaño del csv exportado (~260MB), por lo que no se subió este directamente a Bloque Neón, sino que se subió a OneDrive y se envió el enlace.
5. Tareas realizadas:
   * Preprocesamiento de los datos: Felipe Bedoya (3 horas)

El preprocesamiento consistió en un label encoder manual de los datos, la partición de estos en tokens y su lematización. Finalmente se utilizó un modelo de Bag Of Words y tf-idf para identificar las palabras importantes. También se utilizó el corpus de nltk para quitar las palabras conectoras que generaran ruido. Estas transformaciones fueron añadidas a un pipeline para poder ajustar el modelo escogido por cada uno.

* + Naive Bayes: Camilo Salinas (3 horas)

Con el procesamiento de datos ya hechos, se armó un pipeline con el modelo de Naive Bayes, se utilizó un modelo Multinomial ya que daba una precisión mayor. posteriormente se utilizó GridSearch para hallar el modelo con los mejores hiperparámetros para los datos dados.

Support Vector Machines: Nicolás Orjuela (6 horas)

Ya con el procesamiento hecho, se armó el pipeline junto con el modelo SVM y se aplicó un GridSearch para hallar el modelo con los mejores hiperparámetros para los datos dados. Finalmente, y ya con el modelo entrenado se probó su exactitud y se generó una matriz de confusión para verificar que tan bien clasifica los datos el modelo.

* + Logistic Regression: Felipe Bedoya (3 horas)

Ya con el procesamiento hecho, se armó el pipeline junto con el modelo Logistic Regression y se aplicó un GridSearch para hallar el modelo con los mejores hiperparámetros para los datos dados. Finalmente, y ya con el modelo entrenado se probó su exactitud y se generó una matriz de confusión para verificar que tan bien clasifica los datos el modelo.

* + Investigación del mercado, negocio y estado del arte del problema: Camilo Salinas (3 horas)

Se investigo a través de un documental a cerca de “Clinical Trials for Cancer” de CNBC, sumado a Wikipedia para saber el costo de operar una prueba clínica para en este caso el cáncer. Se observaron varios videos de distintas fuentes para entender las etapas de una prueba clínica y bajo que parámetro son mas efectivas.

* + Análisis de resultados y creación del tablero de control: Felipe Bedoya (1 hora)
  + Preparación del video y presentación: Camilo Salinas y Nicolas Orjuela (1 Hora)

1. Repartición de los puntos:

Cada uno cumplió los roles asignados, por lo que, aunque se realizaron partes diferentes del proyecto, todos aportamos una parte igual a la realización de este, por lo que lo mas justo es repartirnos equitativamente los 100 puntos.

* + Felipe Bedoya: 33 1/3 puntos
  + Nicolás Orjuela: 33 1/3 puntos
  + Camilo Salinas: 33 1/3puntos

1. Puntos por mejorar en el siguiente proyecto:
   * Aprovechar aun más el tiempo y no dejar todo para última hora
   * Aunque la comunicación entre el grupo es buena, nos podríamos beneficiar de tener varias reuniones en donde discutamos la estrategia para abarcar la siguiente etapa del proyecto
   * Se puede aumentar la colaboración y ayuda entre nosotros cuando se nos presentan errores o confusiones teóricas o al momento de ejecutar el código.