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.

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 hiperparámetros (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.

2. 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.

3. 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.

4. Resultados

Los resultados son en general congruentes entre sí. Los dos clasificadores lograron predecir con buena exactitud aquellos pacientes que son o no son elegibles para las pruebas según su descripción médica. Para lograr homogeneidad de los resultados se usó esta métrica y otras más por encima de precisión por su conveniencia en especial considerando el uso de una regresión. Aun así, se recomienda el uso de recall por su consideración de los falsos negativos ya que habiendo pruebas de medicamentos experimentales puede ser mejor tener un mayor número de estos casos y que no haya incertidumbres que puedan afectar la salud de los pacientes.

El mejor modelo fue SVM con una exactitud del 82.6%, una métrica excepcional que indica su calidad. El modelo desde un punto de vista técnico es bueno y en la matriz de confusión podemos ver que los falsos positivos y falsos negativos no representan un porcentaje significante y que están equivalentemente distribuidos los 4 cuadrantes, aun así, es un modelo que requiere mucho tiempo y con más datos para pacientes o descripciones mayores se puede tener un grave problema de sobre dimensionalidad. El uso de múltiples tipos de aprendizaje y técnicas para el mismo nos deja un buen sabor de boca, ya que pudimos corroborar el patrón logístico que tienen estos datos y lograr usar un método y un modelo mucho más eficiente computacional y temporalmente para reproducir los resultados de SVM, aunque con menor eficiencia. Se recomienda usar los dos modelos, uno reentrenado (LogisticRegression) con datos nuevos y uno con un entreno base (SVM) para predecir y reconfirmar los datos.

De todas maneras, no recomendamos la confianza plena en los modelos, aunque su nivel de seguridad sea alto. Cuando la vida de un individuo está en juego es un riesgo económico como negocio y moral como personas confiar el futuro de una enfermedad en una predicción inexacta.

Por su parte, el tablero es sumamente interesante. Debido a la dimensión exagerada de la matriz resultado, se utilizó pandas para sacar aquellas palabras cuya suma en la columna diera los mayores números, significando una alta presencia y un alto índice tfidf, lo que también se traduce en palabras que influencian fuertemente la decisión del clasificador. Es obvio que palabras como cáncer y diagnosis sean necesarias, mientras que precancer es casi imperceptible por su bajo porcentaje de influencia. Es fácil ver que pacientes de linfoma en etapa IV recurrente, o pacientes en condiciones similares con cáncer de seno, carcinoma o pulmón pueden ser pacientes frecuentemente aceptados en los ensayos clínicos. De manera contraria Pacientes que estén consumiendo o hayan consumido Levonorgestrel probablemente sean muy pocos o de muy poca influencia y no terminen siendo escogidos para la prueba debido a posibles efectos secundarios.

5. Trabajo en equipo

a. Roles:

- Líder de proyecto: Felipe Bedoya
- Líder de negocio: Camilo Salinas
- Líder de datos: Felipe Bedoya
- Líder de analítica: Nicolás Orjuela

b. Algoritmo trabajado:

- Naive Bayes: Camilo Salinas
- Support Vector Machines: Nicolás Orjuela
- Logistic Regression: Felipe Bedoya

c. 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.

d. 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.
e. 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.

Los datos preprocesados se encuentran en este enlace: Datos preprocesados

- 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 más efectivas.

- Análisis de resultados y creación del tablero de control: Felipe Bedoya (1 hora)

Debido a la dimensión exagerada de la matriz resultado, se utilizó pandas para armar el tablero y sacar aquellas palabras cuya suma en la columna diera los mayores números, significando una alta presencia y un alto índice tfidf, lo que también se traduce en palabras que influencian fuertemente la decisión del clasificador.

- Preparación del video y presentación: Camilo Salinas y Nicolas Orjuela (1 Hora)

Ya con los resultados y tablero de control se procedió a hacer un análisis desde la perspectiva del negocio y analizar lo obtenido a lo largo del proyecto.

f. 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 más justo es repartirnos equitativamente los 100 puntos.

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

g. Puntos por mejorar en el siguiente proyecto:
    - Aprovechar aún 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.