Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja

pROYECTO ANALÍTICA DE TEXTOS – ETAPA 2

Santiago Bobadilla Suárez - 201820728

Juan José Beltrán – 201819446

~\*~

Mateo Visbal – Experto

Silvana Ruiz – Experta

Laura Mejía – Experta

Icono

Descripción generada automáticamente con confianza media

Contenido

[INTRODUCCIÓN 1](#_Toc103021264)

[PROCESO DE AUTOMATIZACIÓN DEL PROCESO DE PREPARACIÓN DE DATOS, CONSTRUCCIÓN, PERSISTENCIA DEL MODELO Y ACCESO MEDIANTE UN API 1](#_Toc103021265)

[Preparación de los datos – Carpeta Data 2](#_Toc103021266)

[Componentes WEB de la aplicación – Carpeta HTML 2](#_Toc103021267)

[Construcción del modelo – Carpeta Model 3](#_Toc103021268)

[Evaluación del modelo – Carpeta test 3](#_Toc103021269)

[Implementación del API – Main.py 3](#_Toc103021270)

[Desarrollo de la aplicación y justificación 4](#_Toc103021271)

## INTRODUCCIÓN

Después de una primera etapa de implementación de modelos de análisis de textos mediante inteligencia artificial que sigue los lineamientos del negocio, se pasa a una etapa en la que se les entrega el mejor modelo a los operarios de la organización (que en este caso serán médicos), para automatizar y facilitar el proceso de diagnóstico de los pacientes. Se debe recordar que el objetivo de este proyecto de consultoría es “Ayudar al proceso de diagnóstico médico por medio de algoritmos de clasificación basados en información histórica de resúmenes de diferentes pacientes”. Este objetivo se cumple mediante algoritmos robustos de clasificación, pero también mediante una entrega adecuada de la información de los modelos a los médicos, teniendo en cuenta que esta es una herramienta de apoyo a la decisión y que no sustituirá el trabajo realizado por un especialista.

## PROCESO DE AUTOMATIZACIÓN DEL PROCESO DE PREPARACIÓN DE DATOS, CONSTRUCCIÓN, PERSISTENCIA DEL MODELO Y ACCESO MEDIANTE UN API

A continuación, se presentan las diferentes etapas que se realizaron por cada uno de los integrantes del equipo, en donde cada uno tenía uno o más roles específicos. El primero de ellos es el científico de datos, creando un proceso replicable y de calidad para el procesamiento y entendimiento de los datos, construir el modelo analítico y hacer la evaluación respectiva. Posteriormente, el ingeniero de datos toma el producto entregado por el científico de datos y hace las validaciones necesarias con el fin de revisar si se ha generado un modelo de calidad que cumpla con los respectivos requerimientos no funcionales. Finalmente, el ingeniero de software, quien tiene el trabajo más importante de esta etapa y en donde se concentra el trabajo en esta iteración, está encargado de construir una aplicación web o móvil que permita el uso generalizado de la herramienta.

La creación de un API con el cual los médicos se pueden conectar y hacer solicitudes HTTP está implementada en el repositorio BI\_Classifier de GitHub. En este, se encontrarán varias carpetas en donde cada una cumple una función en específico:

### Preparación de los datos – Carpeta Data

En esta carpeta se va a tener la información más importante para poder implementar el modelo, distribuida en los siguientes archivos:

* : en este archivo se guarda toda la información disponible de los extractos médicos entregados por los especialistas, son de utilidad a la hora de entrenar el modelo
* : en esta parte se definen claramente las columnas que se utilizarán para entrenar el modelo, en donde simplemente se tienen los abstractos médicos como variable independiente y los problemas descritos como la variable dependiente
* : este archivo guarda información relevante acerca de las palabras más repetidas en cada una de las patologías, con el fin de obtener las nubes de palabras, y de esta manera darles a los doctores una representación gráfica de los resultados que está arrojando el modelo
* : finalmente, se tienen en este archivo las funciones específicas de transformación de los datos que ya fueron usadas en la etapa 1 del proyecto, en donde se tiene la homogeneización de las palabras a minúsculas, eliminación de los caracteres no ASCII, reemplazo de los números, eliminación de la puntuación y las , para finalmente lematizar y derivar las palabras.

### Componentes WEB de la aplicación – Carpeta HTML

En esta carpeta se tienen los componentes más importantes de la implementación WEB de la aplicación, la cual consta de los siguientes archivos:

* : carpeta con los recursos multimedia necesarios
* : hoja de estilo de la aplicación
* carpeta con archivo JavaScript que permite mostrar los datos correctamente y conectar la aplicación con el modelo.
* : archivo que contiene el esqueleto gráfico de la aplicación

### Construcción del modelo – Carpeta Model

Esta carpeta cuenta con dos archivos importantes:

* : en este archivo se tiene la parte central desarrollada en las etapas anteriores, es decir, el Pipeline con el modelo de análisis de textos. Si se ingresa al archivo, se puede ver que hay tres componentes principales dentro de este, los cuales son la vectorización de los datos, que permiten ajustar la representación de los datos, una transformación a la representación , que recordemos, nos permite tener una noción del porcentaje del documento que está compuesto por una palabra específica.

Aparte de estos tres componentes, se encuentran las funciones encargadas de cargar y escribir el modelo, con el fin de que quede guardado y de esta manera permitir a los médicos usar los diagnósticos a partir de un modelo correctamente entrenado.

Finalmente, se tiene la función encargada de hacer predicciones individuales a partir de un modelo ya entrenado, que será la funcionalidad principal de la aplicación, y es que un médico suba un diagnóstico y se le entregue la patología correspondiente.

### Evaluación del modelo – Carpeta test

En esta carpeta se encuentran los archivos que permitirán evaluar las funcionalidades de la aplicación. El primero de ellos es el archivo , el cual es una colección que entrena el modelo mediante peticiones HTTP, este se puede ver más a detalle en el video. El segundo archivo se llama , y el modo de usarlo es dentro de la plataforma subir el texto guardado en este archivo, para que el modelo prediga la patología correspondiente.

### Implementación del API – Main.py

Finalmente, para poder integrar toda la ejecución del modelo, se usa el framework Uvicorn. En este API que es bastante sencillo de utilizar se encuentran contenidas todas las funciones de la aplicación. Primero, desde esta clase se inicializará el modelo, y en caso de que no exista, se creará uno automáticamente. Adicionalmente, se crean todas las solicitudes HTTP necesarias para el funcionamiento de la aplicación, en donde están entrenar el modelo con todos los pasos necesarios (hacer el under sampling en caso de tener clases no balanceadas y hacer todo el preprocesamiento de datos), predecir un valor en específico y entregar al usuario una respuesta que pueda entender claramente. Más allá de lo anteriormente explicado, los archivos serán simplemente dependencias necesarias para poder ejecutar el API y los modelos desde Python. Para más información, abrir estos archivos en caso de querer ejecutar la aplicación de cero.

Ya habiendo explicado toda la aplicación, se puede ver que está desplegada en un servidor de Heroku, al cual se puede acceder con la siguiente URL: <https://medical-test-bi.herokuapp.com/>

## Desarrollo de la aplicación y justificación

Para acceder a la aplicación directamente en el código fuente, se puede ir al siguiente enlace del repositorio de GitHub correspondiente: <https://github.com/Otros-Codigos/BI_Classifier>

Ya teniendo esto en cuenta, se mencionan algunos aspectos importantes de la aplicación. Lo primero, es que la aplicación está destinada a ser usada por médicos de la organización, quienes tienen una agenda atareada y no les es posible evaluar de forma tan detallada todos los abstractos médicos. Nuevamente es importante mencionar que esta es una herramienta de apoyo a la decisión, y que no es la verdad absoluta y en ningún momento reemplaza al especialista. Adicionalmente, esta herramienta está creada para apoyar el proceso de negocio de diagnóstico de pacientes, ya que permitirá ahorrar tiempos en la identificación de patologías por parte de los médicos para tener más tiempo de acción, con el fin de tratar oportunamente la enfermedad. Esta aplicación es importante porque aumenta de manera significativa la productividad de los médicos que se encuentran en esta zona del negocio, de tal forma que se puedan procesar más pacientes de forma precisa, lo cual, nuevamente se dice, permite tratar las enfermedades de forma oportuna. Además de esto, permite a los médicos tener una noción inicial de la patología, teniendo en cuenta que una descripción tampoco será una conclusión 100% robusta para decir exactamente qué sucede con el paciente, pero permite tener una noción inicial para comenzar con el análisis correspondiente.

## Resultados

Para ver el video de la aplicación funcional, por favor entrar al siguiente video: <https://www.youtube.com/watch?v=5HHr2Yk7kIo>