**INFORME DE RESULTADOS: PROYECTO ANALITICA DE TEXTOS**

**SEGUNDA ETAPA**

**DIEGO ALEJANDRO GONZÁLEZ VARGAS**

**LEONIDAS VILLAMIL PACHÓN**

**DAVID BURGOS MENDEZ**

**Logotipo

Descripción generada automáticamente**

**UNIVERSIDAD DE LOS ANDES**

**2023-10**

Tabla de contenido

[Introducción: 2](#_Toc133760730)

[Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de la API: 2](#_Toc133760731)

[Proceso de automatización de procesos de preparación de datos: 2](#_Toc133760732)

[Construcción del modelo: 3](#_Toc133760733)

[Persistencia del modelo: 4](#_Toc133760734)

[Acceso por medio de la API: 4](#_Toc133760735)

[Desarrollo de la aplicación & justificación 5](#_Toc133760736)

[Desarrollo de la aplicación & sus funcionalidades 5](#_Toc133760737)

[Justificación de utilidad de la plataforma 5](#_Toc133760738)

[Trabajo con el área de estadística: 5](#_Toc133760739)

[Trabajo en equipo 6](#_Toc133760740)

[Identificación de roles: 6](#_Toc133760741)

[Actas de reunión de los espacios planteados: 6](#_Toc133760742)

[Reflexión & actividades realizadas por los miembros: 6](#_Toc133760743)

[Puntos a mejorar en la siguiente entrega: 6](#_Toc133760744)

# Introducción:

El presente documento tiene como objetivo la evidencia de la automatización de un proceso replicable para aplicar la metodología de analítica de textos en la construcción de modelos analíticos. En este sentido, se enseñará el desarrollo de una aplicación web que utilice el modelo analítico de aprendizaje automático desarrollado para la entrega anterior, y se explicará la utilidad de la misma para un rol específico dentro de una organización, empresa o institución. Adicionalmente, se explicará como fueron las tareas de interacción con el grupo interdisciplinario de estadística para validar y mejorar la calidad de la solución analítica planteada.

# Proceso de automatización del proceso de preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de la API:

Para empezar, se hace necesario un breve contexto que permita entender que fue lo que se desarrolló y en este sentido que las respuestas a cada una de las preguntas formuladas por el enunciado cobren sentido.

Así pues, lo realizado por el equipo de trabajo fue una aplicación WEB utilizando el framework Django que permitiera la utilización del modelo planteado para la entrega anterior: un modelo de regresión logística. En este sentido, se tuvo en consideración que montar un API en Django ofrecía ventajas de

Para poder dar respuesta a todas las actividades planteadas en este enunciado, se considera pertinente la división de cada una de las mismas, como procede:

## Proceso de automatización de procesos de preparación de datos:

Para poder automatizar el preprocesamiento de los datos en la aplicación se utilizó un modelo de programación orientada a objetos, en el que se creó una clase de Python que tuviera todos los métodos necesarios para la limpieza y preparación de los datos que contemplaba el modelo original. Para poder anexarlo a la aplicación web, se hizo de vital importancia las recomendaciones dadas en el horario de atención por parte de los monitores del curso, ya que estos dieron el aval para anexar esta “clase” dentro de cada uno de los componentes de la aplicación de django. Como resultado, se obtuvo una clase denominada multiply\_columns\_transformer.py que se convertiría en la encargada del preprocesamiento de la información en todos los casos que fuese necesario, como lo muestra la figura 1.

Captura de pantalla de computadora

Descripción generada automáticamente

*Figura 1: Clase para preprocesamiento*

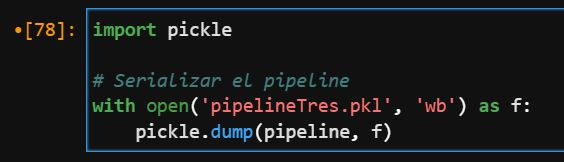
De esta forma, además de tener los métodos del preprocesamiento empaquetados, se puede hacer una simple importación de esta clase en los métodos en encargados de recibir y organizar la información del modelo como es el caso de las views de la aplicación de los comentarios,como se muestra en la figura 2:

Captura de pantalla de computadora

Descripción generada automáticamente

*Figura 2: Clase para preprocesamiento*

## Construcción del modelo:

A pesar de que el preprocesamiento de los datos es independiente para cada uno de los lotes de datos que los clientes le den a la aplicación, el modelo de regresión logística debe ser el mismo siempre, ya que se busca que mantenga y corresponda al modelo con las estadísticas presentadas en la primera entrega. Para alcanzar este objetivo, se siguió con el modelo de programación orientada a objetos y se buscó la estandarización del modelo en un objeto tipo pipeline ejecutable las veces que fuera necesario. En consecuencia, se vio la necesidad de utilizar una librería que permitiera exportar un pipeline con el modelo entrenado. Para ello, se utilizó Pickle, y se creó una versión funcional del pipeline como sigue:

*Figura 3: Construcción de un modelo utilizable en la API*

## Persistencia del modelo:

Para poder alcanzar la persistencia del modelo como tal, se siguió la estrategia anteriormente mencionada de la construcción de un pipeline que, apoyado en el preprocesamiento dado por la clase correspondiente (la expuesta en la primera subsección), mantuvieran funcionando dentro la aplicación el modelo entrenado que se entregó en la primera etapa y que finalmente es quien conserva las métricas de precisión y f-score optimizadas. Sin embargo, el enunciado nos da pie a mencionar el manejo de la persistencia en la aplicación web Django. En este sentido, vale la pena mencionar que, dado que se busca que cada vez que un usuario ingrese a la aplicación pueda insertar sus propios datos y que las estadísticas y demás resultados enseñados correspondan exclusivamente a sus comentarios, se tuvo que hacer una pequeña modificación a la persistencia de Django.

En este sentido, se debe primero que todo mencionar que Django es un framework integral que ofrece persistencia de la información de la aplicación a partir de un ORM propio que permite anclar su información de manera automática con bases de datos como relacionales, como la sqlite3 utilizada en el proyecto. En este sentido, se debe de mencionar que este ORM guardaba la información de manera permanente. Es así que se hizo necesario la incorporación, dentro de las funciones en las que ingresaba la información, de métodos que borraran todos los objetos “comentario” que se tuvieran en la base de datos en el momento que se ingresaran nuevos datos, asegurando resultados transparentes para el usuario final.

## Acceso por medio de la API:

El API se realizó a partir del framework Django. A partir de ahí, el api se organizó en 3 secciones cada una de ellas con la misma importancia:

* Csvs: Esta aplicación de Django dentro de la aplicación web es la que finalmente se encarga de recibir los datos de los clientes y organizarlos en objetos comentario dentro de la base de datos sqlite3
* Comentario: Esta aplicación de Django dentro de la aplicación web es la encargada de hacer los tratamientos de la información. Dentro de ella se encuentra el modelo de comentario, así como todos los objetos de este tipo, por lo que es la que permite la creación de las estadísticas de palabras mas negativas y de gráficas de palabras más o menos negativas.
* Singles: En este modulo se plantea el análisis diferenciado de cada comentario, como herramienta de apoyo a la funcionalidad principal.

Así pues, la primera y la tercera funcionalidad son las que tienen el uso y acceso a los modelos de regresión logística comprimidos en los pipelines. En este sentido, se hace necesario mencionar que la página principal, así como la app principal dentro de Django hotelapp, no hacen uso de las herramientas estadísticas desarrolladas.

# Desarrollo de la aplicación & justificación

## Desarrollo de la aplicación & sus funcionalidades

Como se mencionó anteriormente, se realizó una aplicación web que tiene las siguientes características:

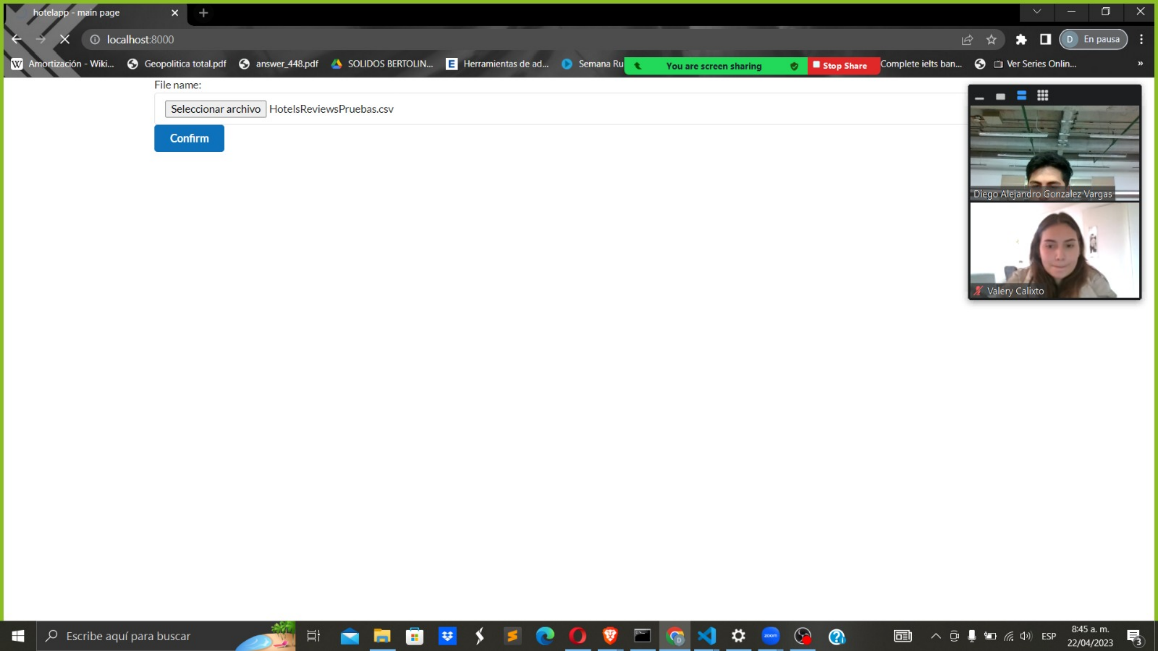
* Ingreso de datos: tomando como organización cliente a la plataforma web a partir de la cual se creó el modelo, se plantea la subida de información de los comentarios a partir del formato dado en que los tiene la plataforma. En este sentido sentido, la aplicación admite los archivos que se plantearon como archivos de pruebas. En este sentido, la información puede ser posteriormente subida en la pestaña de la aplicación web destinada para este fin.
* Revisión de las palabras: A partir del mejor modelo encontrado en la etapa 1(el modelo de la regresión logística), se encontró un diccionario de palabras que expresaban el mayor inconformismo de una persona de un hotel. Estas palabras, además de los adjetivos que de forma automática entraban en el modelo, se encuentran compuestas por elementos propios de los hoteles y que pueden ser de gran importancia a la hora de la reserva de uno de ellos.
* Grafica de pastel: A partir de los resultados dados por el modelo en el análisis de los comentarios ingresados por el usuario, se puede mencionar que la aplicación web enseña una gráfica de pastel que le permite a cada uno de los hoteles la visualización gráfica de su percepción dentro de la plataforma
* Resumen de comentarios negativos: Para facilitar las tareas de mejora de cada uno de los hoteles que pasan sus comentarios de la plataforma dentro de nuestra aplicación web, se plantea la impresión de cada uno de estos comentarios en una sección diferenciada de la plataforma. Esto le permite a cada uno de los hoteles el trabajo especifico en los comentarios negativos, y no en el dispendioso análisis de todos los demás comentarios

Justificación de utilidad de la plataforma:

El usuario final de la aplicación y el modelo de analítica es la industria hotelera que quiere conocer si una reseña es positiva o negativa. Con un modelo de análisis de texto, se puede extraer información valiosa de las reseñas de los huéspedes, como las opiniones sobre la limpieza, el servicio, la ubicación, la comodidad y otros aspectos importantes de su experiencia. Esta información puede ayudar a los hoteles a identificar áreas de mejora y a implementar cambios para satisfacer mejor las necesidades de los clientes. Todo esto se resume en una comprensión mas profunda de la satisfacción del cliente. Adicionalmente un modelo de análisis de texto también puede ser útil para identificar tendencias y patrones en las reseñas de los huéspedes, lo que puede ayudar a los hoteles a comprender mejor las necesidades y expectativas de sus clientes. El desarrollo de una aplicación web permite que el usuario acceda al modelo desde cualquier dispositivo y su uso sea sencillo e intuitivo. Además, la aplicación web permite el acceso de muchos usuarios simultáneamente, así como un alto volumen de datos. Esto le permite ser flexible y tener la capacidad de escalar si la demanda aumenta. Otra de las razones por la que el modelo esta incorporado en una pagina web es para permitir su integración con otras herramientas que utilice el usuario, permitiendo automatizar procesos y en general mejorar la eficiencia operativa. En conclusión el producto le permitirá a su usuario final acceder a información sobre sus clientes y su experiencia en el hotel que le permitirá tomar decisiones informadas para mejorar sus servicios.

## Trabajo con el área de estadística:

A partir del trabajo con el área de estadística, se mostró un primer modelo en el cual la aplicación solo ingresaba los datos y enseñaba tanto los comentarios negativos como los gráficos estadísticos pertinentes. En este sentido, se debe mencionar que la persona del área de estadística se reunió con nosotros y tomo partido en la construcción de la aplicación a partir del planteamiento de nuevas funcionalidades, como filtros que permitieran un mejor aprovechamiento del modelo por parte del usuario final. En este sentido, se debe de mencionar que la aplicación se le añadió la funcionalidad del filtro por palabras negativas para que los clientes finales pudiesen hacer una mirada mas rápida y concisa a los conceptos de especial interés para los clientes. A continuación, se consideró pertinente la exposición de material visual que permitiera la comprobación de los espacios de reunión de la persona de estadística con el grupo:



*Figura 4: Evidencia de la reunión*

# Trabajo en equipo

A continuación, se describe los roles y tareas realizadas por cada integrante del grupo con relación al modelo de analítica de textos y el desarrollo de la aplicación web

## Identificación de roles:

* Líder del proyecto: Diego González. Se encargo de la gestión general del proyecto y verificación de las asignaciones de tareas. Así mismo coordino el encuentro con el área de estadística.
* Ingeniero de datos: Leonidas Villamil. Se encargo del proceso de automatización del modelo de analítica.
* Ingeniero de software responsable del diseño de la aplicación y resultados: David Burgos. Lidero la implementación de la aplicación web.
* Ingeniero de software responsable de desarrollar la aplicación final: Diego González: Se encargo del desarrollo de la aplicación web

## Actas de reunión de los espacios planteados:

1. Lanzamiento y planeación: Se definieron los roles mencionados anteriormente. Se propusieron soluciones para la página web, se repartieron las tareas y se acordó la fecha para la próxima reunión
2. Reunión de seguimiento: Se mostraron los avances a la fecha, se terminaron de asignar tareas.
3. Reunión con área de estadística: Se tuvo el encuentro con la persona de estadística, se le mostro los realizado hasta la fecha y se recibió retroalimentación
4. Reunión de finalización: Se realizo una última reunión para revisar el producto final, se realizaron las ultimas correcciones y se acordó los detalles de la entrega.

## Reflexión & actividades realizadas por los miembros:

Los lenguajes y en particular el español puede llegar a ser ambiguo y contextual, por lo que las palabras pueden tener diferentes significados según el contexto en el que se usan. Lo que hace que sea difícil para una máquina entender la intención detrás de un texto.

Otro desafío en la creación de un modelo de analítica de textos es el preprocesamiento de los datos. Los datos de texto, en el caso del proyecto las reseñas, requirieren una limpieza significativa, incluyendo la eliminación de caracteres no alfabéticos, el manejo de la puntuación y la eliminación de palabras vacías. Además, el procesamiento del lenguaje natural puede ser muy intensivo en recursos, ya que se necesitan grandes cantidades de datos y potencia de procesamiento para entrenar un modelo de calidad, esto se tradujo en tiempos significativos entre mejoras del modelo y retrasos no esperados.

## Puntos a mejorar en la siguiente entrega:

1. Planeación más dinámica: Frecuentemente la realización de ciertas tareas tomo mas tiempo del estimado, por lo que el proceso completo de construcción del modelo y la aplicación web se iba alargando constantemente. Se recomendaría asignar mas tiempo del estimado a las tareas.
2. Investigación más enfocada: El análisis de textos tiene la particularidad de ser dependiente del lenguaje, las herramientas más comunes y utilizadas en el análisis de texto es para el idioma inglés, y por consecuente muchas de estas herramientas no se traducían correctamente al español.
3. Mejor comunicación con el área de estadística: Fue un reto explicarle a una persona ajena al desarrollo del proyecto su funcionamiento y la lógica de su implementación, por lo que fue complico que el estudiante de estadística diera su retroalimentación del modelo, sin embargo las sugerencias que dio acerca de las métricas del modelo fueron utilizadas para mejorar el producto final.