UNIVERSIDAD

EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMÁTICA

Proyecto de Computación 1

Grupo 6

Actividad 4: Clasificador en Python

Resumen

Para este trabajo de la asignatura Proyecto de Computación (I) hemos establecido como objetivo el uso de diferentes procesos y técnicas para poder predecir y catalogar si unas noticias, extraídas de varias fuentes, tratan sobre la temática de 'Odio'.

Para ello hemos realizado un análisis de una amplia cantidad de datos sobre noticias y poder diseñar y entrenar un modelo que tome la decisión mencionada previamente. Para desarrollar todo esto, hemos diseñado una aplicación de escritorio en la cual se nos permitirá tanto "Traininear" como "Testear" sobre un dataset dado, y mostrar el resultado obtenido.

Índice

Abstract

For this work of the Computing Project (I) subject, we have established as an objective the use of different processes and techniques to be able to predict and catalog if some news, extracted from various sources, deals with the theme of Hate.

For this we have carried out an analysis of a large amount of data on news and be able to design and train a model that makes the aforementioned decision. To develop all this, we have designed a desktop application in which we will be allowed to both "Train" and "Test" on a given dataset, and show the result obtained.

Introducción

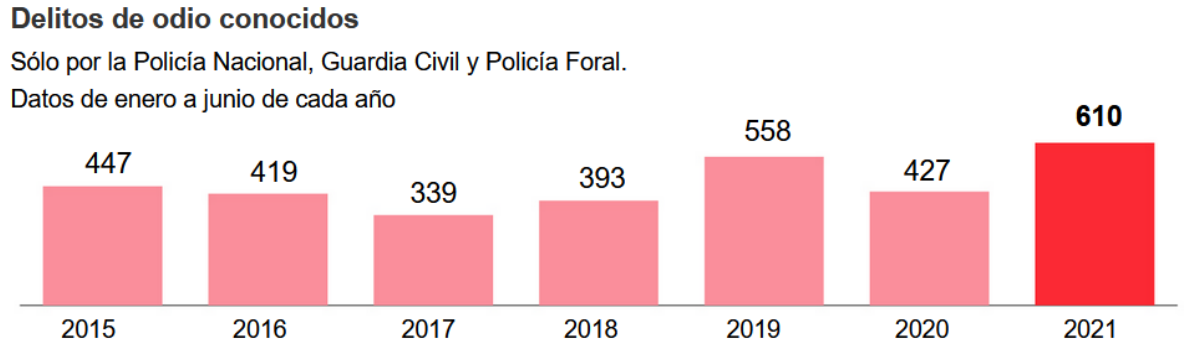
Contextualización

El objetivo de la aplicación consiste en realizar un clasificador de noticias relacionadas con el creciente problema de los delitos de odio en España.

A raíz de la pandemia, pese a que supuso una bajada de los mismos, y siguiendo una tendencia alcista desde hace años; los delitos de odio por temas políticos, demográficos o étnicos se han convertido en un gran problema en nuestra sociedad.

Descripción del Problema

Como podemos apreciar en la siguiente imagen, sacada de El País haciendo uso de datos del ministerio del interior, el número de delitos de odio en España es cada día algo más común en nuestra sociedad.



Fuente: Ministerio del Interior.
EL PAÍS

El aumento de noticias sobre supuestos casos de delitos de odio aparecidas recientemente en los medios de comunicación no es casual. Las estadísticas del Ministerio del Interior reflejan que, en los seis primeros meses de este año, Policía y Guardia Civil ha recibido 610 denuncias por infracciones de este tipo, un 9,3% más que en el mismo periodo en 2019, cuando se contabilizaron 558.
A mayores, una reciente encuesta de Interior señala que solo uno de cada 10 personas que sufren un delito de odio lo denuncia ante la Policía, por lo que queda mucho por concienciar a la sociedad sobre este problema creciente.

Organización del documento

En este documento encontraremos, repartido en apartados principales y sus subapartados correspondientes, todos los temas a tratar sobre el proyecto. Empezaremos con un breve Estado del Arte de la cuestión, en donde hablaremos de herramientas similares a la que hemos desarrollado y hablaremos en mayor profundidad sobre la temática de delitos de odio y su pasado.

De forma seguida empezaremos a tratar la solución que hemos llevado a cabo, metodologías empleadas, recursos, presupuestos y las conclusiones finales.
También será detallado el manual de instalación necesario para correr el programa sin problemas y en su totalidad.

Estado de la cuestión

Estado del arte

A lo largo de la historia, en España siempre se ha llegado a ver delitos de odio por diversas razones sociales, sin embargo, desde el inicio de la pandemia los delitos de odio han repuntado hasta alcanzar cifras superiores a las de antes de la pandemia.

En el primer semestre de 2021 la policía recibió 610 denuncias (un 9,3% más que en el mismo periodo de 2019).

Varios informes presentados por el Ministerio del Interior de España reflejan la evolución de los delitos de odio en España correspondiente 2020, ya que el pasado año se contabilizaron 1.401 denuncias por delitos de odio, un 17,9% menos que en 2019, cuando se conocieron 1.706 casos. El estudio señala que en este descenso ha tenido una clara influencia el confinamiento sufrido por la población durante los meses de marzo a junio debido a la pandemia de la covid-19.

El balance también refleja que la mayoría de las víctimas que sufren este tipo de delincuencia son hombres (un 64%), y con una edad comprendida entre los 26 a 40 años (30,1%). Los menores de edad constituyen el 6,7% del conjunto de las víctimas de "delitos de odio" en 2019, una cifra similar a la del año 2018.

La distribución de las víctimas según su nacionalidad pone de manifiesto que en primer lugar se encuentran las de nacionalidad española, con el 72,3% del total de victimizaciones registradas, siendo la cifra de víctimas extranjeras un 27,7%.

En cuanto a quién comete estos delitos, el perfil del responsable detenido/investigado por "delitos de odio", indica que son principalmente hombres (83%). La mayoría de los autores de estos hechos se encuadran dentro del rango de "18 a 40 años", en concreto, el 54,7%. Y en lo relativo a su procedencia, la mayoría de los detenidos/investigados por incidentes de "delitos de odio" son de nacionalidad española (84,7%).

Competencia

En el mercado existen sistemas de catalogación de noticias que hacen uso de metodologías iguales o similares a las nuestras, como por ejemplo el comercializado por el Institute of Electrical and Electronics Engineers; haciendo uso de Deep Learning.

Sin embargo, consideramos que nuestro sistema está enfocado únicamente en las noticias de odio (lo que conlleva también un uso de datasets mucho más concreto), ciñéndonos exclusivamente a esta problemática. Así, consideramos que somos una buena opción en el mercado.

Antecedentes

Dado a que este tema de delitos de odio tiene una repercusión directa en la sociedad a nivel social, económico y sanitario, para nuestro desarrollo del producto hemos hecho una pequeña investigación sobre diferentes propuestas que se encuentren ya en el mercado que ayuden a mitigar o a gestionar este tema.

Actualmente existen algunas aplicaciones encargadas de la clasificación de noticias como son Flipboard y El Skimm, las cuales son encargadas de recoger noticias y clasificarlas en función del tema sobre el que tratan, estando entre ellos el que más nos interesa a nosotros: ‘delitos de odio’.

Solución

Objetivos generales y específicos

Concluido el desarrollo e implementación de todo el código de nuestro proyecto, nuestro principal objetivo es ser lo más precisos a la hora de clasificar si una noticia trata sobre la temática de delitos de 'Odio' o 'No Odio', teniendo en cuenta todos los factores de los apartados previos.

Para ello, podremos distinguir principalmente 4 apartados en nuestro producto final:

- Proceso de extracción y almacenamiento de los datos necesarios, siendo esto el pilar fundamental de nuestra aplicación. Este proceso es llevado a cabo gracias a una herramienta llamada Pentaho Data Integration, que nos permite acceder a diferentes fuentes de información, extraer de forma precisa la información que queremos y almacenarla en ficheros .txt en nuestros archivos locales.
- Proceso de limpieza y formateado de todo nuestro dataset. Llevado a cabo el proceso anterior, y ya haciendo uso de nuestra aplicación (pese a que sea un proceso automatizado), todo este contenido pasa a través de un proceso de tokenizado, limpieza y estemizado que garantiza que el contenido es relevante y útil.
- Aplicación de los diferentes algoritmos con los que trabajaremos y que serán nuestro motor de trabajo. Usaremos, detallados más adelante, K-NN, Naive Bayes y Árbol de Decisión. Estos harán uso de las librerías pertinentes para, dados los datasets que hemos generado, entrenarse y ser capaces de tomar las decisiones más precisas sobre si una noticia es o no sobre delitos de 'Odio'.
- Visualización de los resultados obtenidos después de todo el proceso. Para ello, hemos diseñado una interfaz gráfica con QtDesigner (explicado más adelante) que hace uso de unas tablas para representar las matrices de resultados de nuestros modelos; tanto para los entrenamientos como para las pruebas.

Cabe destacar que consideramos que la aplicación desarrollada es una herramienta innovadora ya que en el mercado actual se da una escasa existencia de herramientas similares que sean capaces de realizar predicciones sobre temáticas concretas.
A diferencia de otras aplicaciones nuestra interfaz presenta una forma sencilla y precisa de manejar para aquellas personas de este sector en concreto puedan aprender a usarla; además de poder guardar los modelos entrenados en una base de datos local.

Descripción de la solución propuesta

La aplicación discierne, principalmente, de dos funcionalidades que trataremos por separado:

1. **Entrenar diferentes versiones de un clasificador**. Dicho entrenamiento será sobre un algoritmo seleccionado de entre los predeterminados (K-NN, Naive Bayes o Árbol de Decisión).

Para el entrenamiento tendremos que seleccionar una carpeta donde se encuentren las noticias de odio y no odio diferenciadas por el nombre. Dependiendo del algoritmo elegido se creará un clasificador, que podremos guardar en la ruta que el usuario haya decidido.

Después se cargarán noticias que no están marcadas, así podremos comprobar la precisión que tiene nuestro clasificador, y los resultados de este entrenamiento también se guardarán en un fichero para poder consultar las estadísticas en otros momentos, una vez haya acabado nos mostrará estas estadísticas.

Entre los datos mostrados y guardados tras el entrenamiento nos encontraremos la precisión, el recall, número de noticias usadas para el entrenamiento, el porcentaje de acierto, etc.

2. **Ejecutar el clasificador** haciendo uso del algoritmo entrenado.

Para ejecutarlo primero tendremos que cargarlo en la aplicación y después elegiremos la carpeta donde se encuentran nuestras noticias sin clasificar ni marcar, y el clasificador predecirá si las noticias son o no de la temática de Odio.

Utilizaremos la biblioteca pyqt5 junto con el editor QtDesigner para poder desarrollar las interfaces gráficas, mientras que para el desarrollo de los algoritmos emplearemos el entorno de desarrollo integrado Visual Studio Code.

Para la lectura de datos que vamos a utilizar para el modelo usaremos la biblioteca pandas y por otro lado, en la creación de modelos y predicciones, utilizaremos la biblioteca Scikit-learn.

Tecnologías y Librerías usadas

Para el desarrollo de nuestra aplicación hemos usado una serie de herramientas, las cuales utilizan como móvil principal el lenguaje de programación Python.

Utilizaremos la biblioteca pyqt5 junto con el editor QtDesigner para poder desarrollar las interfaces gráficas, mientras que para el desarrollo de los algoritmos emplearemos el entorno de desarrollo integrado Visual Studio Code.

Para la lectura de datos que vamos a utilizar para el modelo usaremos la biblioteca Pandas, que es una extensión de Numpy y ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.

Por otro lado, en la creación de modelos y predicciones, utilizaremos la biblioteca scikit-learn; la cual nos facilitará el uso de los algoritmos con una sencilla llamada a las funciones con los datos de entrada pertinentes.

Datos de Entrada y de Salida

- Con respecto a los datos que entrarán en nuestro sistema, destacamos principalmente dos variantes: Los de entrenamiento y los de pruebas.
  Así, si nuestra intención es entrenar el modelo, los Datos de entrada que le tendremos que facilitar serán aquellos que ya estén catalogados desde un principio. Se pretende así que con ese input pueda empezar a entrenarse e ir aprendiendo.
  Por otro lado, si queremos usar la aplicación con la finalidad de predicción/catalogación, el Input que recibirá serán noticias de carácter genérico que no hayan sido catalogadas aún, consiguiendo así hacer uso de nuestros algoritmos y el entrenamiento previo. La fiabilidad dependerá del nivel de precisión del sistema.

- Con respecto a los datos de salida serán generados únicamente cuando usemos la Predicción.
  La salida esperada serán todas aquellas noticias que estén relacionadas con la temática del proyecto, y serán catalogadas entre “Odio” y “No Odio” en sus respectivos directorios.
  Así mismo, en el apartado de Training, se podrá generar un fichero con el modelo utilizado.

Algoritmos de aprendizaje utilizados

- ***Gaussian Naive Bayes:***

Es un algoritmo de aprendizaje fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales, el cual nos permite construir un modelo de altas prestaciones con un pequeño conjunto de datos.

- ***K-Nearest-Neighbor:***

Es un algoritmo el cual está basado en instancia de tipo supervisado de Machine Learning, el cual puede usarse para clasificar nuevas muestras (valores discretos) o para predecir (regresión, valores continuos).
Es un método sencillo de clasificación de valores buscando los puntos de datos” más similares”, es decir, por cercanía aprendidos en la etapa de entrenamiento.

- ***Árbol de decisión o Decision Tree Classification:***

Es un algoritmo de aprendizaje supervisado, que se utiliza principalmente en problemas de clasificación y funciona para variables de entrada y salida categóricas como continuas. En clasificaciones sencillas suele tener un rendimiento bastante elevado.

Diseño

Fuentes de datos y Dataset generado

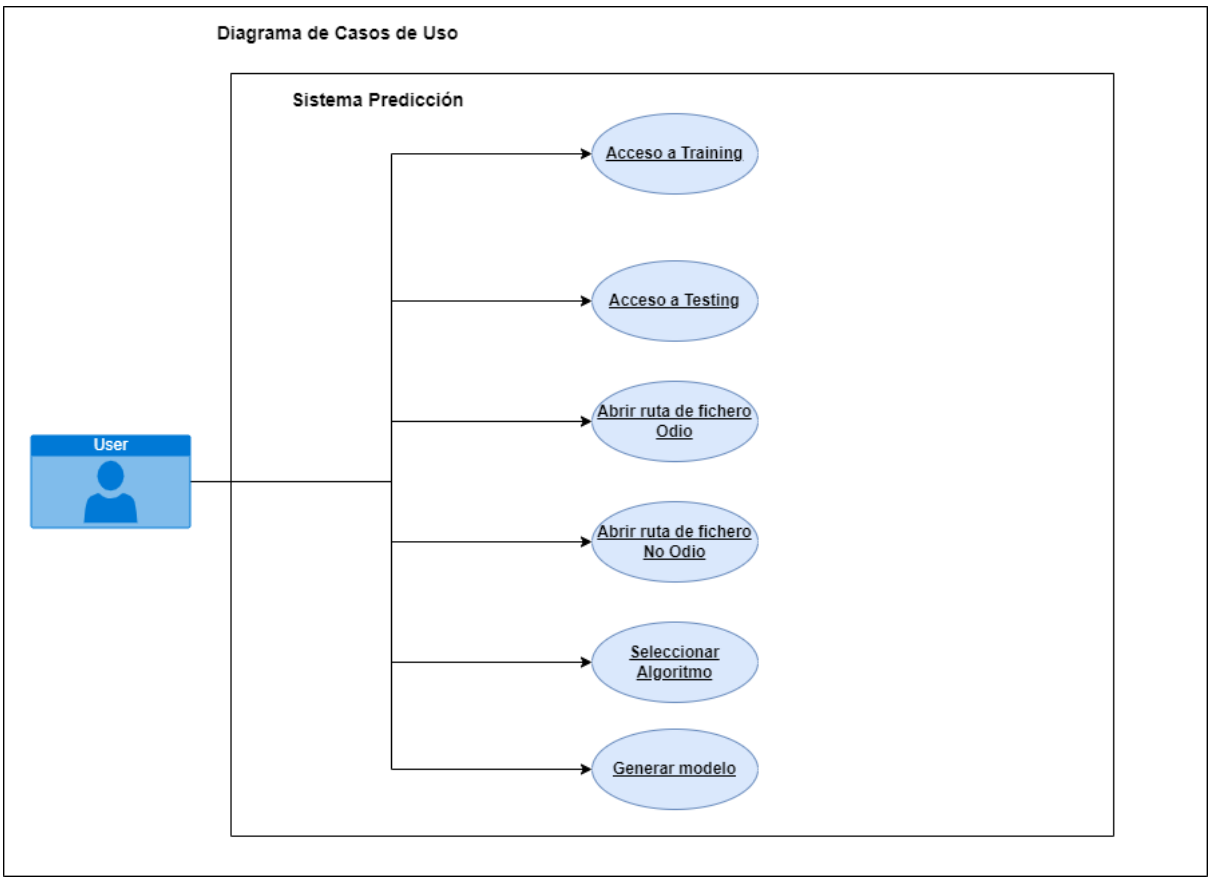
Haciendo referencia a las fuentes de datos, hemos utilizado las siguientes para la extracción de nuestros datasets: “elDiario”, “Vozpópuli” y “20minutos”

Ahora hablaremos de cómo se ha procesado todo el texto obtenido de cada fichero de texto, ya que este contiene muchas palabras sin importancia y elementos que no nos interesan para los siguientes apartados: Testing y Training.
Así, los pasos que hemos seguido han sido los siguientes:
➔ Limpiar toda la información de los textos, eliminando caracteres indeseados.
➔ Tokenizar la información, dividiendo así todo el contenido en palabras sueltas.
➔ Aplicar las listas de parada (stop words), para eliminar sobre los elementos tokenizados aquellos que no aporten información.
➔ Stemmizar el resultado, reduciendo cada elemento a su raíz o palabra originaria.

Todo esto lo conseguiremos haciendo uso de la librería “NLTK”, que será descrita en el apartado “recursos”.

Diagrama UML de clases

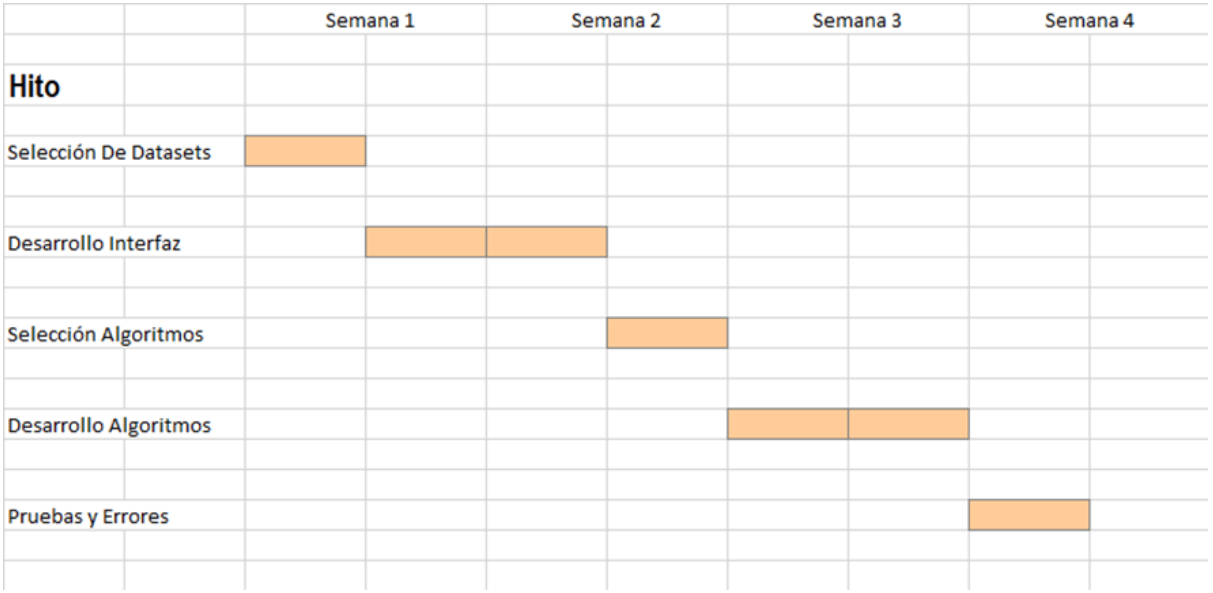
Dadas las características de nuestro producto, hemos diseñado los posibles casos de uso (interacciones con el sistema) entre el actor Usuario y el Sistema de Predicción (la aplicación).



Metodología de trabajo

Plan de trabajo

El diagrama de Gantt nos sirve para planificar las tareas programadas para así saber qué tareas tiene que completarse y en qué fecha. Gracias a este diagrama podemos ver de forma clara en qué fase del proyecto se encuentra nuestro equipo.



Recursos

Visual Studio Code: Editor de código fuente, compatible con varios lenguajes de programación, así como plugins para poder trabajar con el cómputo en la nube.

Terminal de Anaconda (Anaconda Prompt): Usamos Anaconda como suite multiplataforma, que además nos permite instalar y administrar paquetes, dependencias y entornos para los datos con Python.

NLTK: Es un conjunto de programas y bibliotecas usadas para construir programas para análisis de texto en el entorno de programación de Python.

Qt Designer: Una de las herramientas que usamos para diseñar y construir interfaces gráficas de usuario (GUI) con QtWidgets, dichos widgets y formularios creados con Qt Designer se integran perfectamente con el código programado utilizando las señales y el mecanismo de ranuras de Qt para poder asignar fácilmente el comportamiento de los elementos gráficos.

pyqt5: Es uno de los módulos más utilizados en la creación de aplicaciones con interfaces gráficas en Python y esto es debido a su simplicidad; ya que una de sus grandes características es facilita el desarrollo de aplicaciones gráficas complejas en poco tiempo, es decir, arrastra widgets para crear formularios.

pandas: Es un paquete de Python el cual nos proporciona unas estructuras de datos similares a los dataframes de R, también pandas dependen de Numpy, es decir, la librería que añade un potente tipo matricial a Python.

scikit-learn: es una biblioteca dedicada al aprendizaje de máquinas de software libre para el lenguaje de programación Python. Nos incluye algoritmos de clasificación, regresión y análisis de grupos, Cuenta con un diseño para interoperar con las bibliotecas numéricas y científicas de NumPy y SciPy.

Presupuesto

Propuesta económica

Para que el desarrollo del proyecto se lleve a cabo con éxito hemos tenido en cuenta una estimación del valor monetario aproximado que necesitamos.

DESCRIPCIÓN		HORAS	TOTAL (€)
Fase de análisis del proyecto		15	1245
Arquitectura de la información		18	1000
Fase de prototipo		10	830
Desarrollo de las UI		19	1577
Implementación de algoritmos		17	1411
Prueba de errores		13	1079
Despliegue de la app		18	1493
SUBTOTAL			8635

IVA 21%			1813,35
TOTAL:			10.448,35 €

Servicios adicionales

DESCRIPCIÓN		HORAS	TOTAL (€)
Mantenimiento de la herramienta (coste anual)		75	6.059
Servicios extra de soporte de servidores		720	20
TOTAL:			6.079

Manual de instalación

Para la instalación necesitaremos descargar Python, Visual Studio Code y librerías de python.

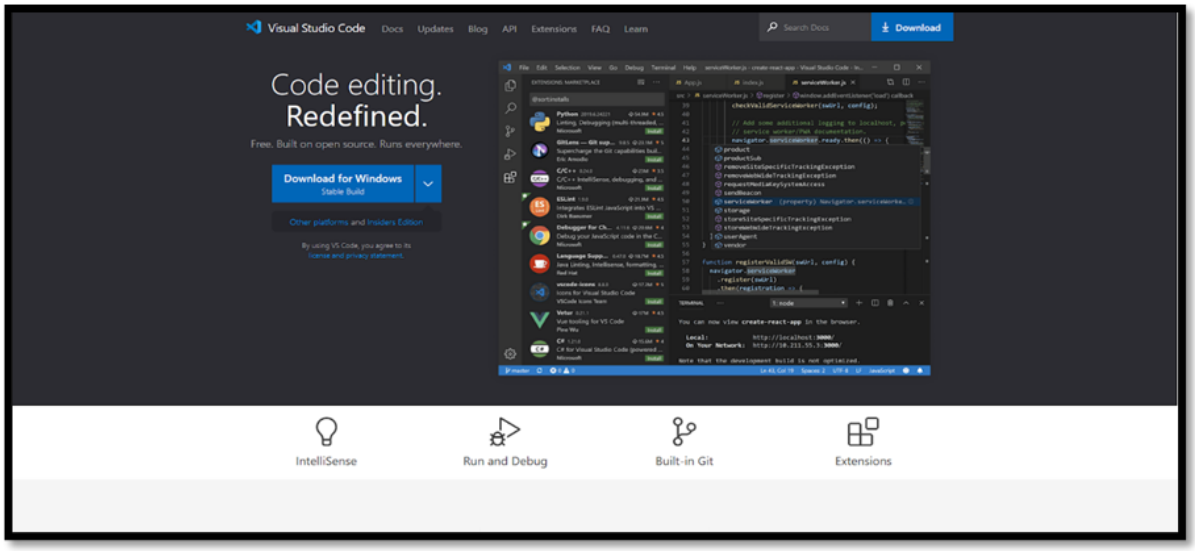
- Python: El instalador puede descargarse desde:

https://www.python.org/downloads/

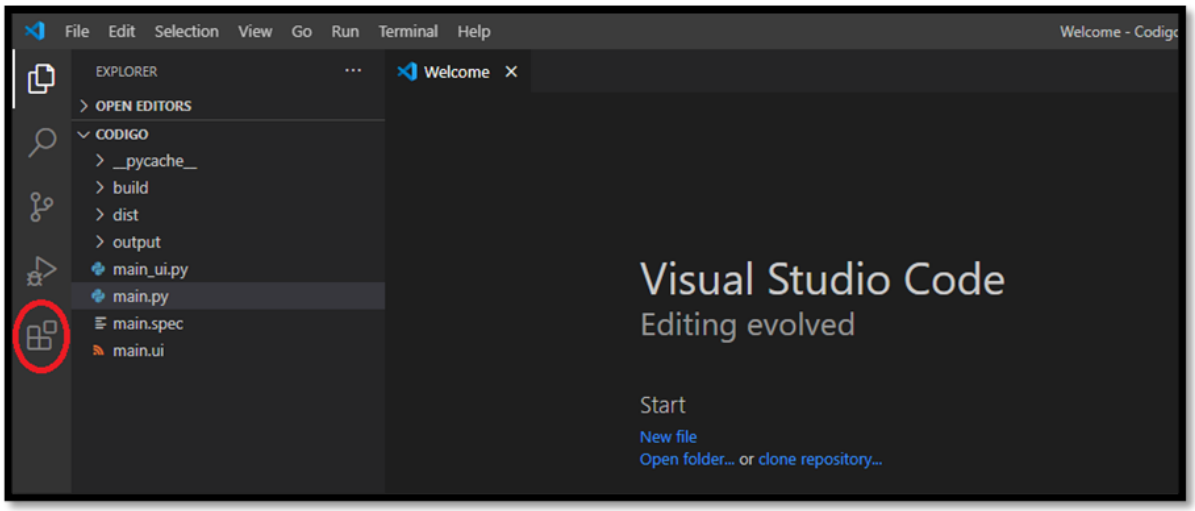
Una vez instalado pasamos a instalar Visual Studio Code

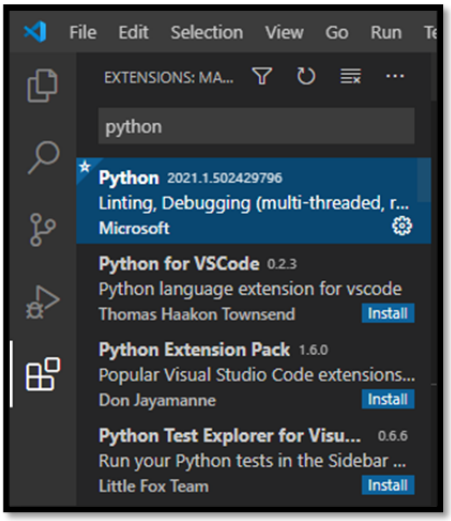
- Visual Studio Code: El instalador se puede descargar desde la página oficial:

https://code.visualstudio.com/



Una vez instalado, abriremos el gestor de extensiones situado en la barra lateral en la izquierda e instalamos la extensión de python.



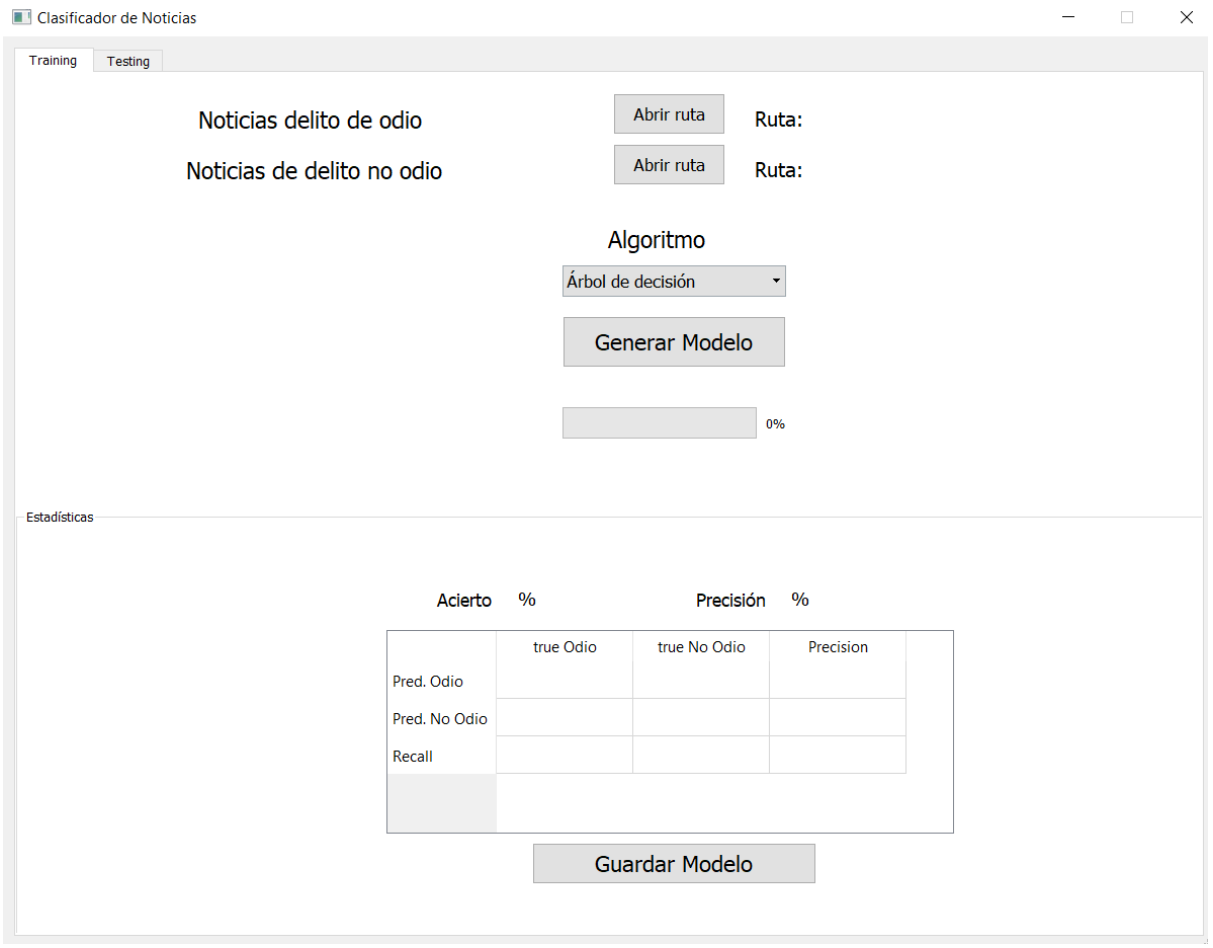


• Librerías: Para la instalación de las librerías usaremos el terminal del sistema operativo y el comando de python “pip”. Por lo que los comandos a introducir en el terminal son:

1. pip install sklearn
2. pip install nltk
3. pip install pandas
4. pip install pyqt5

Manual de usuario

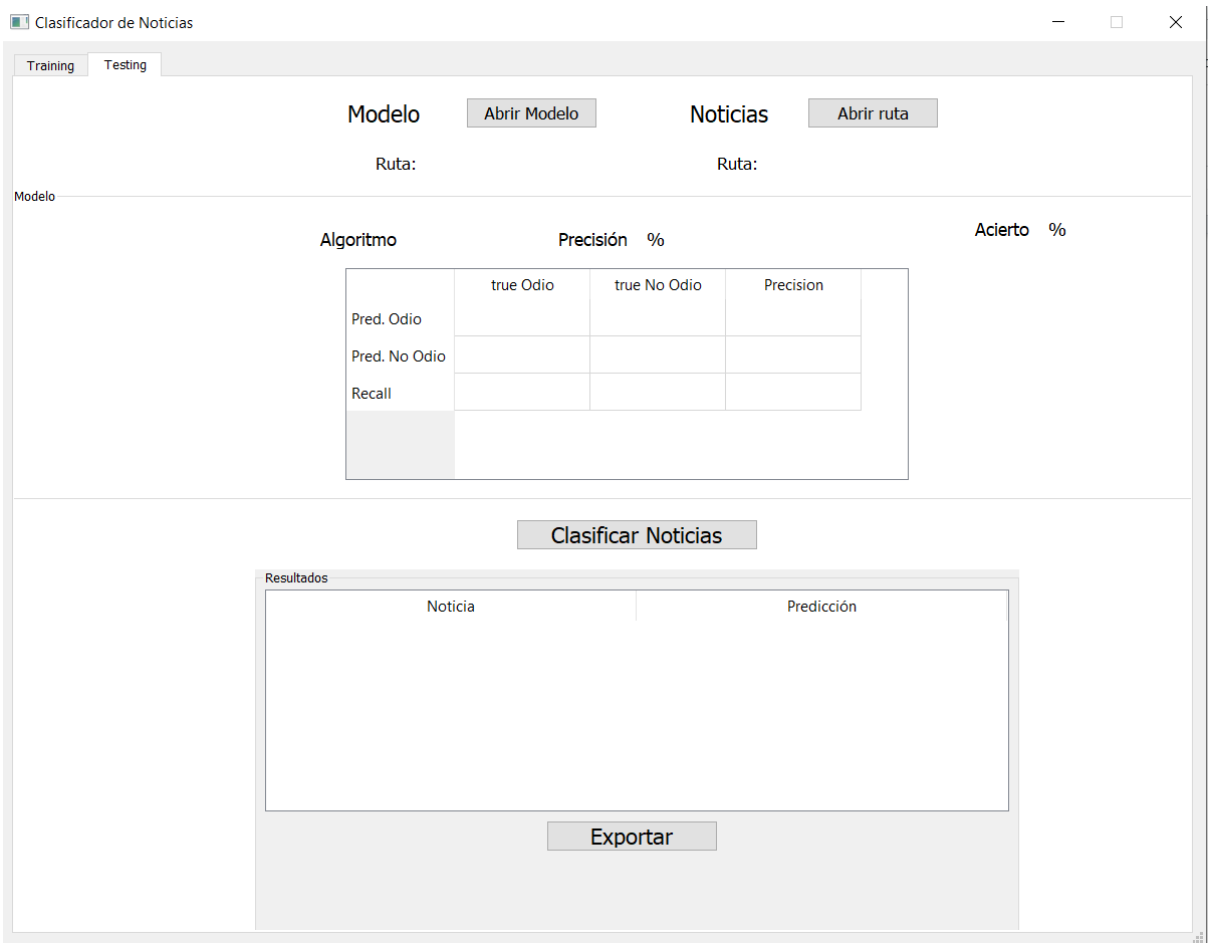
Para ejecutar la aplicación tendremos que abrir la consola de nuestro sistema operativo, dirigirnos al directorio donde se encuentra “main.py” (en la carpeta codigo del proyecto), e introducir el siguiente comando: python main.py



Botón 'Abrir ruta': con este botón se te abrirá una nueva ventana para que selecciones el directorio donde se encuentran todas las noticias en formato de texto.

Botón 'Generar Modelo': al generar el modelo se te mostrará las estadísticas obtenidas de dicho modelo junto con una tabla en la que se muestra que ha considerado que era cada noticia y qué era realmente.

Botón 'Guardar modelo': se abrirá una nueva ventana donde seleccionamos en qué ruta deseas guardar el modelo.



Botón 'Abrir Modelo': se abrirá una ventana en la que seleccionas el modelo que quieres usar para clasificar las noticias. Una vez cargado el modelo se mostrarán sus estadísticas en el grupo justo debajo.

Botón 'Abrir Ruta':Se abrirá una ventana en la que seleccionamos el directorio donde se encuentran las noticias sin clasificar.

Botón 'Clasificar Noticias': ejecuta el modelo sobre las noticias no marcadas y muestra cada noticia y su predicción en la lista justo debajo.

Botón 'Exportar': se exporta el resultado a un excel y se dividen los ficheros utilizados en dos carpetas: despoblación y no despoblación, y en cada una de ellas se insertará la noticia correspondiente a la predicción del modelo

Conclusiones

Queremos destacar en primer lugar, con respecto a la utilización de los modelos de aprendizaje, que el resultado de nuestro sistemas y futuras variaciones depende directamente del volumen y calidad de los datasets utilizados. Con esto queremos hacer referencia a que muchas veces, dados unos resultados en principio dentro de los estándares de calidad que queremos, al probar dicho modelo en testing los resultados

finales obtenidos distan bastante de los deseados. Esto se puede deber al sobre-entrenamiento de modelos (comúnmente llamado Overfitting) o al escaso volumen de noticias utilizadas durante el entrenamiento (under-fitting).

Otro factor que consideramos clave, y que sería una modificación a tener en cuenta para el futuro, es el desarrollo de la interfaz gráfica. QtDesigner nos brinda la posibilidad de desarrollar una interfaz gráfica completa y funcional en un breve período de tiempo y con poco esfuerzo. Sin embargo, consideramos que para mayor flexibilidad, diseño más moderno y nuevas funcionalidades; el desarrollar una interfaz completamente desde cero y Ad-Hoc hacia el cliente tendría un valor añadido de gran calidad.

Por último, queremos hacer referencia a la posición de nuestro producto en el mercado. Dado el trabajo realizado y el resultado del mismo, creemos que nuestro sistema es un fiel competidor con respecto a las diferentes ofertas existentes a día de hoy. Por ello, nuestra intención es ir creciendo poco a poco como empresa e ir adentrándonos en nuevos sectores de los Sistemas de Catalogación y Recomendación.

Trabajos futuros

Una de nuestras principales intenciones con esta aplicación de predicción sobre noticias de 'Odio' sería intentar presentarnos a un concurso público que nos permita acceder a mayor parte de la sociedad gracias a algún instrumento de la sociedad.

Consideramos que dado el background social que contiene nuestro sistema, una posible línea de negocio a futuro sería trabajando directamente con organismos del Estado que pongan en manos de instituciones públicas (colegios, institutos, universidades, etc.) nuestro programa. Así, conseguimos una nueva forma de catalogación de noticias evadiendo la desinformación, sobreinformación o titulares engañosos; basándonos íntegramente en el contenido de las noticias.

Bibliografía

Delitos de Odio, disponible en:
http://www.interior.gob.es/web/servicios-al-ciudadano/delitos-de-odio

"El gobierno eleva la desinformación a la categoría de amenaza", disponible en:
https://www.elmundo.es/espana/2021/12/28/61cb41befc6c83a5638b45cc.html

Pandas Documentation, disponible en: https://pandas.pydata.org/

Diseño de interfaces gráficas, disponible en: https://www.qt.io/design

Scikit-Learn documentation, disponible en: https://scikit-learn.org/stable/