Para abordar este proyecto, necesitaremos realizar varias tareas:

1. **Extracción de datos de la variable respuesta en EUROSTAT:**

• Se descargan de la pagina [Statistics | Eurostat (europa.eu)](https://ec.europa.eu/eurostat/databrowser/view/lfsq_pganws/default/table?lang=en), o a través de la librería Eurostat de Python o directamente el csv

- **DUDA**: Dado que se nos dice que nos basaremos para el tiempo en la serie temporal de menor duración (desde 2014, GDELT), ¿los datos de todas las variables predictoras serán desde 2014 o tenemos que llevar a cabo algunas transformaciones, imputar variables o modelar por separado?

-**DUDA**, ¿descargamos directamente el csv y filtramos las condiciones, que son muchos más registros, o usamos la biblioteca de Eurostat de Python para descargar los datos? Los datos se descargan de otra forma (la columna tiempo del otro Dataframe contiene todos los tiempos de 1998 a 2023 y en el otro hay una columna por tiempo)

-**DUDA**, variable geo, eu20-27 y ea20, ¿los añadimos? Luego están los países individuales, no todos de la unión europea, ¿los usamos todos en el filtro? O solo Europa

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

-**DUDA**, variable citizen, cogemos SOLO FOR O TAMBIEN EU27-20 FOR Y NEU27 2020 FOR

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

-**DUDA**, de la variable respuesta, ¿NECESITAMOS UNA SERIE QUE INCLUIR EN EL MODELO X CON LOS DATOS FEMENINOS Y OTRO CON LOS MASCULINOS?

(SEXO, EDAD, WSTATUS, GEO, MAS LA SERIE, OBS VALUE EN MILES DATOS)

1. **Diseño y Desarrollo de la API para Google Trends:**

• Define los endpoints de la API para la extracción de información de Google Trends.

• Utiliza bibliotecas FastAPI para implementar la API en Python.

• Integra la API con la interfaz de Google Trends para realizar consultas y obtener datos agregados sobre volúmenes de consultas.

* **DUDA**, ¿ES NECESARIO CREAR UNA APP CON FASTAPI O FLASK PARA LA LLAMADA, O PODEMOS HACERLO EN PYTHON O EN GOOGLE COLAB DIRECTAMENTE?

1. **Extracción de datos de Google Trends:**

• Filtramos de Google Trends (o bien con la API o bien en Python) para extraer datos de SVI para los temas en los períodos que necesitamos (preguntar si desde 2014)

• Almacenaremos los datos extraídos en un formato adecuado, como un DataFrame de Pandas en Python si es posible, para poder usarlo de una forma más fácil.

* **DUDA**: los topics para Pytrends (¿serán estos?)

[Kohns and Bhattacharjee 2023](about:blank):

* Labour Market: topic - Unemployment benefits, topic - jobs, topic - Unemployment, Welfare & unemployment
* Credit, Loans & Personal Finance: topic - Investment, topic - Mortgage, topic - Interest rate, Credit & lending, Investing
* Jobs & Education: Education, Jobs
* Business & Industrial Activity: Construction, consulting & contracting, Business services, Transportation & logistics, manufacturing
* Health: que puede refinarse utilizando la categoría, Medical Facilities & Services

¿Los usamos todos? ¿usamos otros?

* **DUDA**, error 429, demasiadas request en pruebas. Vamos a probar con Google colab, para ver la solución.
* **DUDA:** las seis muestras que debemos tomar de Pytrends y luego promediar SON con la misma fecha y filtros pero POR CADA TOPICS QUE ANALICEMOS, ¿NO?

1. **Diseño y Desarrollo de la API para GDELT:**

• Definiremos los endpoints de la API para la extracción de información de GDELT.

• Utilizaremos la biblioteca FastAPI para implementar la API en Python (si es necesario)

• Integraremos la API con la base de datos de GDELT para extraer datos sobre el tono de los artículos y la popularidad de los temas.

* **DUDA**, LA MISMA QUE EN GOOGLE TRENDS, ¿LA LLAMADA TIENE QUE SER IMPLEMENTADA DESDE UNA API OBLIGATORIAMENTE?

**5. Extracción de datos de GDELT GKG:**

• Para acceder a los datos de GDELT, elegiremos entre BigQuery (desventaja, es de cloud, y tiene una capa gratuita y otra que no lo es, NO SABEMOS SI PODREMOS CON LA CAPA GRATUITO OBTENER LOS DATOS QUE NECESITAMOS), o herramientas específicas como GDELTDoc (la mas útil, porque analiza texto, Problema, cantidad de datos y limitaciones de capacidad de nuestro ordenador)

• Filtraremos los datos de GDELT GKG para seleccionar solo nuestros topics, las 3 keywords, la fecha, las palabras del artículo (más de 500…)

• Calcularemos la puntuación del tono de los artículos y la tasa de popularidad del tema promediando las medidas obtenidas de GDELT para los artículos

gd.timeline\_search("timelinetone", f) # para el tono

gd.timeline\_search("timelinevol", f) # para la popularidad

* **DUDA**: ¿se nos puede facilitar mejores servidores para analizar con BIG QUERY o con GDELTDOC?
* **DUDA**: TOPICS DE GDELT, ¿USAMOS ESTOS? De World Bank
* 112 Public Expenditure policy
* 131 Inclusive growth
* 22 Jobs
* 24 Enterprise Development
* 323 MSME Finance
* 324 Financial Inclusion
* 51 Social inclusión
* 651 Access to Education
* 632 Health service delivery
* **DUDA**: ¿los topics hay que usarlos por separados o juntos? Cuando nos dicen que nuestro filtro tenga 3 de nuestras palabras clave, ¿a cuáles se refiere?

**6. Preprocesamiento de Datos:**

• Preprocesaremos los datos obtenidos de Google Trends, GDELT y la variable respuesta de Eurostat para que estén en un formato adecuado para su uso como predictores

• Nos aseguraremos de manejar adecuadamente los diferentes periodos de tiempo y frecuencias temporales de los datos. En un primer momento, GDELT PASARLO DE DIARIO A MENSUAL COMO GOOGLE TRENDS Y LUEGO A ANUAL COMO VARIABLE RESPUESTA) y Google Trends a anual para coincidir con la variable objetivo.

• También tendremos en cuenta que, al estar trabajando con series temporales, puede ser importante DESESTACIONALIZAR para asegurarnos de que todas las series se encuentren en las mismas condiciones.

MODELO NOWCASTING, CREACION

**7. Integración de datos y preparación para el modelo de machine learning:**

• Combinaremos los datos extraídos de Google Trends y GDELT en un único conjunto de datos, utilizando métodos de unión.

Para combinar las series temporales extraídas de Google Trends y GDELT deberemos tener en cuenta:

1. Congruencia en fechas:

Nos aseguraremos de que las series temporales de Google Trends y GDELT estén en la misma frecuencia temporal (por ejemplo, mensual). Verificaremos que las fechas coincidan en ambas series temporales para que puedas combinarlas adecuadamente. Gdelt habrá que pasarla de diario a mensual y de mensual a anual. Google Trends habrá que cambiarlo de mensual a anual (NECESARIO PARA INTRODUCIRLO AL MODELO Y PODER PREDECIR LA VARIABLE OBJETIVO)

1. Normalización de datos:

Es posible que desees normalizar las series temporales para que estén en la misma escala. Esto puede ayudar a mejorar la estabilidad y la convergencia del modelo.

1. Selección de características (opcional, DUDA):

Podemos utilizar técnicas de selección de características para identificar las variables más relevantes de Google Trends y GDELT. Así podremos reducir la dimensionalidad y mejorar la eficiencia del modelo (en apartado DUDA)

• Realizaremos el preprocesamiento adicional que sea necesario para preparar los datos para su entrada en el modelo de Machine Learning. La codificación de variables categóricas sexo y nacionalidad como dummies desde la variable respuesta, unir la variable objetivo con el resto para montar la X, la imputación de NaN…

* **DUDA**: Para aclararnos, ¿cada variable predictora será la serie temporal extraída de cada uno de los topics de Google trends (6, como los tópicos de arriba); los de GDELT serán series temporales del tono y la popularidad de los 9 tópicos definidos arriba (¿9 o 18?, uno por tono y popularidad o uno por cada tópico y por tono y popularidad) o los que nos de randbee; serie temporal de la variable respuesta(1 con sexo total, o dos diferenciando entre femenino y masculino) y las variables dummies de sexo y geo?
* **DUDA**: Se nos ha comentado en el anexo que habría que lidiar con la dimensionalidad de los datos de Google Trends y de GDELT, pero más tarde se determina que si las muestras son insuficientes es cuando tendremos que reducir los inputs según Randbee lo determine, ¿tenemos que realizar feature selection, feature extraction o usar modelos de regresión regularizados: lasso o ridge DIRECTAMENTE o esperar a las directrices de Randbee si es necesario?

**8. Desarrollo y entrenamiento del modelo de machine learning:**

**•** Seleccionaremos el tipo de modelo de machine learning que mejor se adapta a nuestros objetivos de análisis. Será un modelo LSTM de red neuronal vanilla, que alimentaremos con series temporales procedentes de Google Trends, GDELT GKG y las dummies sexo y ciudad de la variable respuesta, así como con la variable socioeconómica como objetivo (se nos solicita que la función pérdida sea MSE, y el optimizador ADAM)

model.add(LSTM(50, activation='relu', input\_shape=(n\_steps, n\_features)))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mse')

• Dividiremos los datos en conjuntos de entrenamiento y prueba y entrenaremos el modelo utilizando el conjunto de entrenamiento.

• Evaluaremos el rendimiento del modelo utilizando el conjunto de prueba y ajustaremos los parámetros del modelo según sea necesario para mejorar su rendimiento

* **DUDA**: la variable objetivo, ¿serán dos series en la X o incluso en la Y de nuestro modelo?, ¿MASCULINO Y FEMENINO, o solo una con todas las personas en activo? ¿Cuál pretendemos que sea nuestro target exactamente, o tendremos varios, por sexo, nacionalidad…?

**9. Validación del modelo y análisis de resultados:**

• Una vez entrenado el modelo, validaremos su rendimiento utilizando métricas de evaluación, en este caso, RMSE, MSE, MAPE… (recordemos que es un modelo de regresión)

• Realizaremos la validación del modelo con técnicas como la validación cruzada.

• Analizaremos los resultados obtenidos y realizaremos los ajustes (en los datos o en el modelo) para mejorarlo en la aplicación final.

**10. Implementación de la API para el Modelo:**

• Definiremos un endpoint en la API para realizar predicciones utilizando el modelo entrenado.

• Integraremos el modelo en la API para que pueda recibir solicitudes y devolver predicciones sobre la variable a predecir.

**11. Pruebas y Validación de las APIS:**

• Realizaremos pruebas de todas las API y del modelo de "nowcasting" (tanto de las extracciones si las hemos usado, como del modelo)

**11. Documentación:**

• Documentaremos adecuadamente todas las API y el modelo de "nowcasting".

**12. POSIBLES MEJORAS PARA EL MODELO EN EL FUTURO**

• Problemas con los que nos hemos encontrado durante nuestro desarrollo, cómo abordarlos y posibles mejoras para el futuro.

**13. Conclusiones, acciones y decisiones**

• BUSCAR POR NUESTRA CUENTA QUE PODRIAMOS HACER CON ESTE MODELO, PARA QUE NOS PODRÍA SERVIR, FUTURAS APLICACIONES…