Ciencia de datos: guía de entrega individual

Agosto 2020 - El Puente

INSTRUCTOR: Gabriel Vázquez Torres
gabriel@thebridgeschool.es

PROFESOR: Clara Piniella Martinez
clara.piniella@thebridgeschool.es

Explicación de entrega

Esta entrega individual tiene como objetivo practicar diferentes conceptos sobre EDA y API. Además, se presentará el proyecto.

El alumno debe elegir una asignatura que prefiera. Idicamente, esta entrega se extenderá con los Unity 2 y 3 del tema. Esto es, además de esta entrega de EDA, habrá más entregas enfocadas en Machine Learning (Unity 2) y Data Science como producto (Unity 3).

Requisitos

Los siguientes requisitos son obligatorios:

1. El proyecto debe dar respuesta a una **hipótesis** explicado a continuación).
2. El estudiante hará una presentación y deberá documentar todos los pasos que realice.
3. El alumno debe dividir las tareas que tiene que hacer.
4. Es obligatorio que el alumno utilice *trello* (u otro relacionado) para gestionar las tareas en diferentes estados: TODO, HACIENDO y HECHO. BACKLOG y REVIEW son opcionales.
5. El envío debe realizarse antes del 31/08/2020 a las 23:59.

---

6. La entrega debe enviarse en a. *cremallera* archivo por correo electrónico con esta estructura:

a. Una carpeta **s rc /** que contiene todo el código fuente.

si. Una carpeta **documentación /** que contiene todos los documentos relacionados con documentación.

C. Una carpeta **r recursos /** que contiene otro contenido útil (imágenes, ...)

re. Una carpeta **src / utils /** que contiene todos los módulos utilizados por el *principal* expediente.

mi. Un archivo **src / main.ipynb** que contiene toda la funcionalidad. Este archivo solo debe contienen importaciones, pandas, matplotlib, solicitudes, ... y llamadas a su **src / utils / *** módulos.

F. Hay, al menos, cuatro módulos dentro **src / utils /:**

yo. “Folder_tb.py” que contiene la funcionalidad genérica relacionada con abrir, crear, leer y escribir archivos.

ii. “Visualization_tb.py” que contiene la funcionalidad genérica relacionada con pandas, matplotlib, seaborn y otras bibliotecas que se enfocan en visualizaciones.

iii. "Mining_data_tb.py" que contiene la funcionalidad genérica relacionada con la recopilación de datos, datos limpios y otros (métodos de disputa, como trabajar con múltiples jsons)

iv. “Apis_tb.py” que contiene la funcionalidad genérica relacionada con el trabajo con API.

v. Otros que necesita el alumno.

gramo. Un archivo **src / api / server.py** que contiene la funcionalidad que inicia el matraz API. Hay, al menos, una función GET:

yo. Uno que debe permitirle recibir un ***t oken_id*** v alor y, si ***t oken_id*** es igual a S , devuelve los jsons que contienen la lógica explicada a continuación. De lo contrario, devuelve una cadena con un mensaje de error.

1. S es el DNI del alumno que comienza con la letra: Ejemplo: "B80070012"

ii. Otros que son relevantes para el proyecto.

h. Los json que se devuelven son estos al menos:

yo. Al menos, el alumno deberá devolver un json que represente los datos tratados y depurados.

ii. Dependiendo de los datos y el problema, el estudiante debe devolver datos interesantes con el objetivo de que su programa sea útil.

Hipótesis

Normalmente, el objetivo en un proyecto EDA es responder una pregunta o demostrar un axioma. Esto es, dando todas las razones necesarias para explicar por qué la respuesta a la pregunta es *o específicamente* y refutar o reafirmar un axioma. *dondeste*

Un ejemplo de una hipótesis en el proyecto de covid-19 podría ser:

Creemos que el estado de alarma de cada país tiene un impacto en la progresión de la infección diaria.

Presentación

Todos los estudiantes deben hacer una presentación sobre su proyecto. El presentador del grupo utilizará un archivo de presentación para explicar todos los pasos del flujo de trabajo con gráficos.

La duración de la presentación no será superior a los 7 minutos por lo que es realmente importante y necesario explicar los puntos esenciales del trabajo.

Los pasos del proyecto

La idea del proyecto consta de diferentes pasos:

1. Encontrar el tema: el alumno debe encontrar el proyecto en sí. Esto es algo que quiere hacer.

2. Encuentra los datos relacionados con el proyecto: investiga dónde puede estar y si es accesible para el público.

3. Defina una hipótesis: encuentre algo que pueda concluir con sus datos.
4. Define los pasos necesarios para demostrar o no tu hipótesis.
5. Con la estructura del código definida y usando Python:
    a. Obtenga sus datos. Tal vez necesite usar una API, tal vez un archivo. Negociación de datos.
    si. Limpia tus datos. Detecte valores atípicos, valores raros y reemplace los valores de NaN si es necesario.
    C. Dibuje todos los gráficos que necesita tanto para comprender sus datos como para mostrar los resultados necesarios.
    re. Cree una API que devuelva lo explicado en el **R** **requisitos** sección.
        Quizás le resulte útil hacer más de un punto final.
    mi. Explique por qué a partir de sus gráficos y otros resultados se puede argumentar la conclusión.

6. Documente todos los pasos, comprima los archivos necesarios y envíelos al correo de los maestros.

NOTA: Realice todos los pasos para completar los requisitos de los criterios.

Los recursos

Con el objetivo de encontrar todos los recursos necesarios, el alumno puede realizar búsquedas en Internet.

Hay páginas en las que puede encontrar buenos ejemplos de proyectos y conjuntos de datos de EDA:

- Kaggle : aquí puede encontrar millones de ejemplos con millones de conjuntos de datos. Hay diferentes partes en las que puede aprender de novatos o expertos.
- Googledatasetsearch : aquí puede encontrar millones de conjuntos de datos. Es una buena página si desea encontrar los datos que necesita.
- GoogleApis : aquí tienes muchas apis de diferentes temas para obtener datos. Páginas del ayuntamiento,
- páginas de estadísticas y miles de API que puedes encontrar en Internet.

Si el alumno no tiene ninguna inspiración, podemos recomendar las siguientes asignaturas de EDA:

- Analizar cómo la situación pandémica ha cambiado la vida de algunos sujetos.
- Analizar tweets para determinar si algún evento cambia las tendencias.
- Analizar conjuntos de datos de películas para concluir si hay más actrices o películas románticas.

- Analizar conjuntos de datos deportivos para concluir si Messi, Lebron James o Fernando Alonso son los mejores en sus deportes.
- Analizar conjuntos de datos de enfermedades para concluir si existen relaciones entre los síntomas y el número de muertes (u otra relación).
- Analizar conjuntos de datos climáticos para concluir si el cambio climático es real.
- Analizar conjuntos de datos de video para concluir si los videos divertidos tienen la mayor cantidad de vistas.

Criterios de evaluación

Para esta entrega, existen diferentes opciones de entrega. Cada alumno debe elegir qué entrega quiere hacer. **C** es el requisito mínimo para esta entrega. Hay una jerarquía en las opciones: **C** → **si** → **UNA** → **A +** *

No está permitido hacer:

- B sin C
- A sin B y C
- A + sin A, B y C

Opcion C

Aparte de todos los requisitos que están escritos en el **Requisitos** sección, están los siguientes ejercicios obligatorios:

1. Documente todos los pasos. Estructura tu código para mantenerlo limpio usando buenas prácticas.
2. Recopile los datos. Intente hacer cada llamada, recopila los últimos datos actualizados.
3. Determine y explique si se limpian los datos. Si no es así, límpielo.
4. Cree una API que devuelva un Json con la lógica explicada. El servidor de matraces debe ejecutarse ejecutando el **s rc / api / server.py** expediente.
5. Muestre diferentes tendencias para cada columna en su conjunto de datos.
6. Represente, en un gráfico circular, el tiempo que necesitó para cada punto del **El proyecto pasos** s ección.
7. Responda las preguntas:
    a. ¿Fue posible demostrar la hipótesis? ¿Por qué?
    si. ¿Qué puede concluir sobre su estudio de datos?
    C. ¿Qué cambiaría si necesitara hacer otro proyecto de EDA?
    re. ¿Qué aprendes haciendo este proyecto?

Opción B

1. Muestre el histograma de cada columna de su conjunto de datos con *si* *ins* = 5. ¿Cómo son los rangos? ¿pintado?
2. ¿Cuáles son las columnas con mayor correlación? Dibuja la matriz de correlación.
3. Utilice las funciones de Matplotlib para mostrar todas las gráficas. No con pandas directamente.

Opcion A

1. Investigue para guardar cada parcela en archivos locales.
2. Utilice distribuir módulos para cada funcionalidad. Los cuadernos jupyter no deben tener ningún bucle ni funciones. Solo debe tener las iniciales import y la llamada a las funciones necesarias.
3. Aparte dematplotlib, use seaborn para mostrar las gráficas.
4. Responda las preguntas:
    a. ¿Hay valores atípicos o datos raros?
    si. ¿Cuáles son las columnas que tienen más valores repetidos?

Opción A +

Hay diferentes A +. Puedes hacer las que quieras:

1. Cree una solicitud de extracción para todo el proyecto.
2. ¿Cómo puedes poner tu servidor de matraces con una IP pública sin Heroku? darse cuenta de que flask inicia el servidor en una red privada por defecto (localhost)
3. ¿Cómo puede poner su servidor de matraces con una URL pública sin Heroku?
4. ¿Cómo puedes poner tu servidor de matraces con una IP pública con H eroku ?
5. ¿Cómo puedes poner tu servidor de matraces con una URL pública con Heroku ?
6. ¿Existen más URL de donde recopilar sus datos ?. Explicar por qué. Si es así, recójalo y combínelo con sus datos.
7. Para practicar POO y conceptos de ingeniería / arquitectura en computación, defina todas las funciones dentro de las clases y haga que el programa sea funcional usándolas. Después de eso, use un programa para crear el diagrama de clases.
8. Usando su propia URL de API, use web scraping para obtener el json y mostrar los datos.