ES-PEC4-enun

May 17, 2021

1 Programación para la ciencia de datos - PEC4

En este Notebook encontraréis un ejercicio que supone la cuarta actividad de evaluación continuada (PEC) de la asignatura. Esta PEC consta de un único ejercicio a resolver, que engloba muchos de los conceptos cubiertos durante la asignatura.

El objetivo de este ejercicio es desarrollar un **paquete de Python**, fuera del entorno de Notebooks, que nos permita resolver el problema dado. Este tendrá que incluir el correspondiente código organizado lógicamente (separado en módulos, organizados por funcionalidad), la documentación del código (docstrings) y tests. Además, se deben incluir los correspondientes archivos de documentación de alto nivel (README), así como los archivos de licencia y dependencias (`requirements.txt`).

Se nos pide que implementemos un paquete (módulo) de Python que sea capaz de realizar un anàlisis de datos con información sobre diferentes entrevistas en los Estados Unidos durante el último año en relación con el coronavirus y cómo el expresidente Trump ha gestionado la situación. Por un lado tendremos los datos sobre las entrevistas (quién las ha hecho, cuándo, cuál era la pregunta, a qué grupo de población estaba dirigida, ...) y además tendremos información sobre la credibilidad de estas entrevistans en función de quién las ha realizado.

2 Los datos

Los datos a analizar nos son proporcionados en dos colecciones de datos separadas: `covid_approval_polls.csv` y `covid_concern_polls.csv`. Por otro lado tenemos `pollster_ratings.xlsx` con la información sobre la credibilidad de las entrevistas por entrevistador (*pollster*). Estos datos provienen del repositorio Five Thirty Eight. Os aconsejamos usar los archivos que os hemos proporcionado nosotros ya que los datos tienen alguna modificación respecto los datos originales.

`covid_approval_polls.csv` contiene entrevistas sobre la aprobación o no de la actuación de Donald Trump durante el inicio de la pandemia. Mientras que `covid_concern_polls.csv` contiene entrevistas sobre la concienciación de la población sobre el impacte económico o bien el impacto en su salud o la de su familia del coronavirus. Y como hemos mencionado anteriormente `pollster_ratings.xlsx` contiene información sobre la credibilidad del agente entrevistador.

Echando un vistazo a los archivos proporcionados, podréis ver que los diferentes archivos contienen bastante información. Para resolver el ejercicio propuesto, seguramente no usaréis toda la información que contienen estos archivos.

2.1 covid_approval_polls.csv

El archivo `covid_approval_polls.csv` contien 2227 líneas con información sobre 2226 entrevistas:

```
start_date,end_date,pollster,sponsor,sample_size,population,party,subject,tracking,text,
    approve,disapprove,url
2020-02-02,2020-02-04,YouGov,Economist,1500,a,all,Trump,FALSE,Do you approve or
    disapprove of Donald Trumps handling of the coronavirus outbreak?,42,29,https://
    d25d2506sfb94s.cloudfront.net/cumulus_uploads/document/73jqd6u5mv/econTabReport.pdf
2020-02-02,2020-02-04,YouGov,Economist,376,a,R,Trump,FALSE,Do you approve or disapprove
    of Donald Trumps handling of the coronavirus outbreak?,75,6,https://d25d2506sfb94s.
    cloudfront.net/cumulus_uploads/document/73jqd6u5mv/econTabReport.pdf
...
```

La primera línea contiene la cabecera y el nombre de las columnas para cada entrada (línea) del documento.

Comentar que las columnas `approve` y `disapprove` están en tanto por ciento respecto la respuesta sobre la población (`sample_size`) de cada entrevista.

2.2 covid_concern_polls.csv

El archivo `covid_concern_polls.csv` contien 639 líneas con información sobre 638 entrevistas:

```
start_date,end_date,pollster,sponsor,sample_size,population,party,subject,tracking,text,
    very,somewhat,not_very,not_at_all,url
2020-01-27,2020-01-29,Morning Consult,"",2202,a,all,concern-economy,FALSE,How concerned
    are you that the coronavirus will impact the following? U.S. economy,19,33,23,11,
    https://morningconsult.com/wp-content/uploads/2020/02/200167
    _crosstabs_CORONAVIRUS_Adults_v2_JB-1.pdf
2020-01-31,2020-02-02,Morning Consult,"",2202,a,all,concern-economy,FALSE,How concerned
    are you that the coronavirus will impact the following? U.S. economy,26,32,25,7,https
    ://morningconsult.com/wp-content/uploads/2020/02/200191
    _crosstabs_CORONAVIRUS_Adults_v2_JB-1.pdf
...
```

La primera línea contiene la cabecera y el nombre de las columnas para cada entrada (línea) del documento.

Comentar que las columnas `very`, `somewhat`, `not_very` y `not_at_all` están en tanto por ciento respecto la respuesta en la población (`sample_size`) de cada entrevista.

Notar que los porcentajes a veces suman algo más de 100 (o menos). Esto es debido a errores de redondeo de los datos, pero a vosotros no os afecta este hecho en el análisis que tenéis que hacer en esta práctica.

2.3 pollster_ratings.xlsx.

El archivo `pollster_ratings.xlsx` contiene 454 líneas con información sobre 453 agentes entrevistadores:

```
Pollster,Pollster Rating ID,# of Polls,NCPP / AAPOR / Roper,Live Caller With Cellphones,
    Methodology,Banned by 538,Predictive Plus-Minus,538 Grade,Mean-Reverted Bias,Races
    Called Correctly,Misses Outside MOE,Simple Average Error,Simple Expected Error,Simple
     Plus-Minus,Advanced Plus-Minus,Mean-Reverted Advanced Plus Minus,# of Polls for Bias
```

```
    Analysis,Bias,House Effect,Average Distance from Polling Average (ADPA),Herding
    Penalty,latest_poll
Monmouth University,215,108,yes,yes,Live,no,-1.6,A+,D
    +1.3,81%,21%,5.4,6.7,-1.2,-2,-1.6,71,D +1.8,R +0.4,5.2,0.2,3/13/20
Selzer & Co.,304,48,yes,yes,Live,no,-1.3,A+,D +0.1,79%,25%,4.6,6.1,-1.3,-1.8,-1.1,31,D
    +0.2,D +0.2,5.2,0,11/1/18
ABC News/The Washington Post,3,73,yes,yes,Live,no,-1.3,A+,D
    +0.5,72%,7%,2.8,4.8,-1.7,-1.7,-1.2,68,D +0.8,D +1.3,3.8,0.12,10/31/18
...
```

La primera línea contiene la cabecera y el nombre de las columnas para cada entrada (línea) del documento.

3 Ejercicio

Será necesario que generéis funciones que os permitan hacer los siguientes cálculos:

1. Del archivo `covid_approval_polls.csv`:

   1.1 Implementad una función que cuente **de forma eficiente** y muestre por pantalla el número de veces que aparecen los patrones descritos (es decir, en cuántas líneas aparece) a continuación en el archivo, incluyendo un mensaje explicativo de los valores que mostráis por pantalla. Los patrones a considerar son:

   - El término *Huffington Post*
   - Una url (sea http o https) con formato pdf. Por ejemplo: *https://d25d2506sfb94s.cloudfront.net/cumulus_uploads/document/73jqd6u5mv/econTabReport.pdf*
     Un ejemplo de *output* sería:

   The pattern Huffington_Post appears X times.

   The pattern url_pdf appears Y times.

   1.2 ¿Si tuviéramos un archivo de 1Gb lo harías igual? Si no es así, implementar la solución para este caso.

   1.3 ¿Si tuviéramos 100 archivos de 1Gb cómo lo harías? No hace falta implementar la solución, sólo una pequeña descripción de cómo resolverías el problema.

2. Leer los archivos facilitados de la forma más eficiente teniendo en cuenta las tareas pedidas a continuación y en el ejercicio 3, 4 y 5. Justificar vuestra decisión. Preparad los datos para cada .csv, obteniendo dos tablas que llamaremos `approval_polls`(proveniente de `covid_approval_polls.csv`) y `concern_polls` (proveniente de `covid_concern_polls.csv`) de forma que se cumplan todos los siguientes requisitos:
   - Sólo estaremos interesados en las entrevistas en las cuales su agente entrevistador (*pollster*) esté en la tabla `pollster_ratings.xlsx`
   - Sólo estaremos interesados en las entrevistas sin *tracking*.
   - Sólo estaremos interesados en las entrevistas en las cuales su agente entrevistador no ha estado vetato (*banned*).

Nota: Para leer archivos en formato *xlsx* podéis instalar librerías addicionales en la máquina virtual.

3. Sobre los datos extraídos en el ejercicio 2 de la tabla `approval_polls`, calculad y representad gráficamente:

   3.1 El número de personas que aprueban (*approve*) y el número de personas que desaprueban (*disapprove*), para las preguntas que contienen las palabras *Trump* y *coronavirus* en el texto. Representaremos estos datos por cada partido (*party*) (*D* (demócratas), *R* (republicanos), *I* (independientes), *all* (personas sin clasificar por partido)).

4. Sobre los datos extraídos en el ejercicio 2 de la tabla `concern_polls`, teniendo en cuenta las siguientes transformaciones sobre el grado en la clasificación (*grade*) $*$, calculad y representad gráficamente (excepto el 4.1):

   4.1 Cuánta gente ha participado en las entrevistas. Representar el resultado por pantalla debidamente formatado.

   4.2 Cuánta gente en la materia (*subject*) de la entrevista relacionada con la economia (*economy*) está *very* (*concern*, preocupación) y cuánta está *not_at_all* (*concern*, preocupación).

   4.3 Cuál es el porcentaje de gente en la matèria (*subject*) de la entrevista relacionada con la infección (*infected*) está *very* (*concern*, preocupación) y cuánta está *not_at_all* (*concern*, preocupación).

   4.4 Cuántas entrevistas hay por cada nota clasificatoria (*grade*).

$*$ La nota clasificatoria será reducida a sólo los valores *A, B, C, D, F* teniendo en cuenta que en caso de tener una valoración entre dos categorías nos quedaremos con la inferior. *Ejemplo B/C* $\rightarrow$ *C, B-* $\rightarrow$ *B, B+* $\rightarrow$ *B*.

5. A partir de los datos del ejercicio 4, crearemos una nueva variable que será la puntuación (credibilidad) que le daremos a ese agente entrevistador. Esta puntuación vendrá dada por:

   puntuación = nota evaluada + *Predictive Plus-Minus*

   Donde la nota evaluada se refiere a que la clasificación por *A, B, C, D, F* se evaluará de la siguiente forma: A $\rightarrow$ 1, B $\rightarrow$ 0.5, C $\rightarrow$ 0, D $\rightarrow$ -0.5, F $\rightarrow$ -1

   5.1 Calcular (y representar gráficamente) para aquellas entrevistas que su puntuación sea superior o igual a 1.5:

   a) El número de personas según el nivel de preocupación (*concern very, somewhat,...*) en función si la entrevista había finalizado estricatament antes del 2020-09-01 (1 de septiembre de 2020), o después.

   b) El porcentaje de personas según el nivel de preocupación (*concern very, somewhat,...*) en función si la entrevista había finalizado estricatament antes del 2020-09-01, o después. (**Nota:** porcentaje respecto el número de personas por cada grupo antes del 2020-09-01 y después).

   5.2 ¿Qué podéis decir de las dos gráficas obtenidas en el ejercicio anterior (5.1)? ¿Qué conclusiones podéis extraer?

Nota: en el caso que alguna de las variables sobre las entrevistas (filas) que necesitemos usar no sea un campo informado se descartará aquella entrevista. Si hay campos no informados pero que no los usamos a lo largo del ejercicio no será necesario eliminar esa entrada.

Además, tendréis que gener código que permita **representar los resultados de los ejercicios 3, 4 (excepto 4.1) y 5 gráficamente**, podéis igualmente representar el resultado por pantalla de forma que podamos comprobar vuestros resultados de manera exacta. Para cada función será necesarioa que penséis qué tipo de gráfica es la más adecuada para representar el resultado correspondiente.

El código tendré que estar correctamente comentado, incluyendo la documentación de funciones, y correctamente testeado usando la librería `unittest`. Los tests proporcionados tendrán que dar una cobertura mínima del 50% de la funcionalidad propuesta.

3.1 Cobertura de los tests

La medida de la cobertura de los test se usa para evaluar la eficacia de los test propuestos. En particular, sirve para determinar la calidad de los tests desarrollados y para determinar las partes críticas del código que no han estado testeadas. A modo de medida para este valor os proponemos el uso de la herramienta `Coverage.py`. En la documentación, podréis encontrar cómo instalarla y cómo usarla.

Para evaluar la calidad de los test desarrollados en la PEC4, pedimos un mínimo del 50% de cobertura.

4 Uso de Git

Por tal de poner en práctica lo que habéis aprendido en la Unidad 6 sobre `Git`, proponemos el uso de `GitHub Classroom` para desarrollar vuestro paquete Python. GitHub Classroom es una herramienta gratuita de código abierto que ayuda a simplificar el uso educativo de GitHub. Hemos usado GitHub Classroom para crear una aula como esta y donde hemos creado una tarea para la PEC4. Para poder usar este espacio que hemos creado, os aconsejamos seguir los pasos indicados en aquesta guía que explican cómo crear un repositorio para trabajar la tarea que hemos preparado, y que encontraréis en este enlace.

El uso de esta herramienta no es obligatorio para la evaluación de la PEC4, pero creemos que es una muy buena oportunidad para poner en práctica vuestros conocimientos en un entorno vital para todo aquel que trabaje o quiera trabajar en el ámbido de la ciencia de los datos.

4.1 Criterios de corrección

Esta PEC se valorará siguiendo los criterios siguientes:

- **Funcionalidad** (5.75 puntos): Se valorará que el código implemente correctamente lo que pide el enunciado.
  - Ejercicio 1 (0.75 puntos)
  - Ejercicio 2 (0.3 puntos)
  - Ejercicio 3 (0.7 puntos)
  - Ejercicio 4 (1.5 puntos)
  - Ejercicio 5 (1.75 puntos)
  - Visualitzaciones (0.75 puntos)
- **Documentación** (0.5 puntos): Todas las funciones de los ejercicios de esta PEC tendrán que estar correctamente documentadas utilizando docstrings (en el formato que prefiráis).

- **Modularidad** (1 punto): Se valorará la modularidad del código (tanto la organización del código en ficheros como la creación de funciones).
- **Estilo** (0.5 puntos): El código tiene que seguir la guía de estilo de Python (PEP8), exceptuando los casos donde hacerlo complique la legibilidad del código.
- **Tests** (1.25 puntos): El código tiene que contener una o varias *suites* de tests que permitan comprobar el buen funcionamiento de las funciones implementadas, obteniendo un mínimo del 50% de cobertura.
- **Requerimientos** (0.5 puntos): Es necesario crear un fichero de requerimientos que liste (sólo) las librerías necesarias para ejecutar el código.
- **README** y **licencia** (0.5 puntos): Se valorará la creación de un fichero README, que presente el proyecto y explique cómo ejecutarlo, así como la inclusión de la licencia bajo la cual se distribuye el código (podéis elegir la que queráis).

4.1.1 Importante

Nota 1: Del mismo modo que en las PECs anteriores, los criterios transversales se valorarán de manera proporcional a la parte de la funcionalidad implementada.

Por ejemplo, si el código sólo implementa la mitad de la funcionalidad requerida, y la documentación de esta parte es perfecta, entonces la puntuación correspondiente a la parte de documentación sería de 0.25.

Nota 2: Es imprescindible que el paquete que entreguéis se ejecute correctamente en la máquina virtual, y que el fichero de README que incluyáis explique claramente cómo se tiene que ejecutar vuestro código para generar las gráficas resultantes del análisis. Además de cómo ejecutar los test y comprobar la cobertura.

Nota 3: Entregad el paquete como un único archivo .zip en el Registro de Evaluación Continua. **El código de Python tendrá que estar escrito en ficheros planos de Python.**