Actividad. Valores faltantes y análisis preliminar de los datos

Unidad	3
Entrega	Subir de forma grupal el documento a la actividad en el campus virtual.

1. Enunciado

La actividad está organizada en ejercicios y se evalúa sobre un máximo de 10 puntos. Todos los ejercicios tienen el mismo peso en la evaluación de la actividad. Un ejercicio puede contener varias preguntas o apartados, en cuyo caso la puntuación del ejercicio se repartirá de forma equitativa entre las preguntas o apartados que lo componen.

1. Utiliza el siguiente enlace para descargar el Census Income Dataset, junto con su fichero de nombres. Utiliza un Jupyter Notebook y el paquete Pandas para abrir el archivo y presentarlo en formato DataFrame, donde el nombre de las columnas debe corresponder con el nombre real de las variables.

   <https://archive.ics.uci.edu/ml/datasets/Census+Income>

   a. Como se ha visto en actividades anteriores, los valores de las variables en este DataFrame tienen un espacio vacío al comienzo del string que define la variable. La presencia de este espacio vacío dificulta el trabajo de análisis y puede crear confusiones en el futuro. Busca información sobre el método applymap de Pandas y escribe una pequeña descripción de su funcionamiento. Utiliza el método applymap para limpiar los valores de cada una de las variables que tengan formato string, eliminando el espacio vacío que aparece al comienzo del string. Renombra las variables nuevas para que tengan el mismo nombre que las variables originales.

   b. Crea un DataFrame llamado df_less con los registros del DataFrame original que cobren menos de 50 mil dólares. Crea un DataFrame llamado df_more con los registros del DataFrame original que cobren más de 50 mil dólares.

   c. Utiliza el método value_counts de Pandas sobre df_less para obtener el número de registros que cobran menos de 50 mil dólares agrupados por nivel educativo. Ordena estos resultados de mayor a menor, señalando qué grupos tienen más incidencia y cuáles menos. Repite las mismas operaciones sobre el DataFrame df_more. Explica los resultados y comenta si crees que tienen sentido.

d. Partiendo del DataFrame original y utilizando el método value_counts de Pandas, obtén el número total de registros del dataset por nivel educativo. Utiliza este resultado para normalizar los resultados obtenidos en los apartados b y c.

e. Los resultados normalizados no representan ahora el número de registros por nivel educativo en cada caso (menos de 50 mil dólares y más de 50 mil dólares), sino la probabilidad de que un determinado nivel educativo se encuentre en cada grupo de salario. Ordena las probabilidades de mayor a menor para cada grupo de salario, señalando la probabilidad asociada a cada nivel educativo. Explica los resultados y comenta si crees que tienen sentido.

f. Representa en una gráfica de barras los resultados obtenidos en los apartados c y e, con los valores ordenados de mayor a menor. Recuerda añadir los títulos de los ejes, el título del gráfico, la leyenda y todas las opciones de visualización que consideres relevantes para comunicar mejor los resultados.

g. Representa, en el tipo de gráfica que consideres más conveniente, la manera en la que afecta la edad y el género al hecho de que una persona se encuentre en el grupo de < 50 mil dólares o de > 50 mil dólares. Añade a la gráfica todas las opciones de visualización que consideres relevantes para comunicar los resultados.

2. Utiliza el siguiente enlace para descargar el OkCupid Dataset. Utiliza un Jupyter Notebook y el paquete Pandas para abrir el archivo y presentarlo en formato DataFrame, donde el nombre de las columnas debe corresponder con el nombre real de las variables.

<https://www.kaggle.com/andrewmvd/okcupid-profiles>

a. Crea un DataFrame reducido, llamado df_red, a partir del DataFrame original en el que sólo se consideren las variables "age", "status", "sex", "orientation", "body_type", "diet", "drinks", "height", "income" y "job".

b. Vamos a transformar este dataset en un dataset de clasificación. El objetivo de esta clasificación será predecir si cada una de las personas que aparecen en el dataset trabajan en el ámbito STEM o si no lo hacen. Recuerda que STEM es el acrónimo de *Science, Technology, Engineering and Mathematics.* Para crear el problema de clasificación definiremos una variable nueva en el DataFrame df_red llamada "is_stem". Utiliza el método apply de Python, combinado con una función lambda, para crear la variable "is_stem" de forma que esta variable valga 1 si la persona trabaja en estos ámbitos y valga 0 si no trabaja en estos ámbitos. Considerar como

valor 0 los valores faltantes o los casos en los que no se han aportado respuesta. Eliminar la variable "job" una vez se haya creado la variable "is_stem".

c. Una vez obtenido el DataFrame anterior, calcular el porcentaje de instancias que trabajan en el ámbito STEM. Crear un DataFrame llamado X en el que se consideren todas las variables de df_red excepto "is_stem". A partir de df_red, crear un Pandas Series llamado Y que contenga los valores de la variable "is_stem".

d. Utilizar el método de scikit_learn llamado train_test_split para crear dos DataFrames X_train y X_test y dos Series Y_train e Y_test. El tamaño del conjunto de test deberá ser el 30% del tamaño del dataset original. La semilla de números aleatorios deberá ser igual a 42. La división en conjunto de train y conjunto de test se deberá hacer de manera estratificada, para asegurar que la proporción entre clases del problema de clasificación es similar en ambos subconjuntos.

e. Comprobar que las proporciones de instancias que trabajan en el ámbito STEM en el subconjunto de entrenamiento y en el subconjunto de test son similares.

f. Calcula, para el conjunto de entrenamiento y para el conjunto de test, las distribuciones de las variables "age", "orientation", "body_type", "height" e "income". ¿Son las distribuciones del conjunto de test iguales a las del conjunto de entrenamiento en todos los casos? Representa las distribuciones con histogramas o gráfico de barras, según el tipo de variable. Cada histograma o gráfico de barras debe corresponder a una variable, y se deben representar las distribuciones de test y de entrenamiento. Para hacer más legible la representación, representa las distribuciones con un valor de alpha (transparencia) de 0.7. ¿Hace falta que las dos distribuciones sean iguales en todos los casos? ¿Qué consideraciones deberemos tomar para no obtener métricas distorsionadas respecto de la realidad? Tened en cuenta que, al tener los conjuntos de entrenamiento tamaños distintos, tanto los histogramas como los gráficos de barras deben estar normalizados. ¿Qué ocurre cuando se representan sin normalizar?

3. Utiliza el siguiente enlace para descargar el conjunto de datos del Portal de Datos Abiertos del Ayuntamiento de Madrid sobre accidentes de bicicletas correspondiente al año 2019. Utiliza un Jupyter Notebook y el paquete Pandas para abrir el archivo y presentarlo en formato DataFrame, donde el nombre de las columnas debe corresponder con el nombre real de las variables.

<https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=20f4a87ebb65b510VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default>

a. Utilizando el método apply de Python, junto con una función lambda, crea una nueva variable en el DatafFrame llamada "hora_int", de forma que esa variable represente la hora, sin considerar minutos, en la que cada accidente se ha producido. Por ejemplo, si un accidente se produce a las 19:43:00, la variable "hora_int" debe valer 19.

b. Calcula el número de accidentes que se ha producido en cada hora del día, sin considerar los minutos. Represéntalos en una Series de Pandas de forma que las horas estén ordenadas de menor a mayor. Representa esa Series de Pandas en un gráfico de barras. ¿A qué hora se produce la mayor cantidad de accidentes? ¿Qué hora suele ser la más tranquila?

c. Utiliza el método groupby de Pandas para calcular el número de accidentes por estado meteorológico. ¿A qué estado meteorológico corresponde la mayor cantidad de accidentes?

4. Utiliza el siguiente enlace para descargar el conjunto de datos del Portal de Datos Abiertos del Ayuntamiento de Madrid sobre calidad del aire correspondiente al año 2019. Utiliza un Jupyter Notebook y el paquete Pandas para abrir el archivo y presentarlo en formato DataFrame, donde el nombre de las columnas debe corresponder con el nombre real de las variables.

<https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=aecb88a7e2b73410VgnVCM2000000c205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default>

a. Crea un DataFrame df_no_e4 en el que se tengan en cuenta únicamente las medidas de monóxido de nitrógeno (representado por las letras NO) medidas en la estación 4. Si es necesario, consulta el documento de descripción de los datos, aportado en el mismo enlace que aparece en el enunciado, para saber cómo filtrar las columnas.

b. Utiliza el método stack de Pandas para transformar el DataFrame de forma que contenga el índice y una única columna, llamada 'est_4'. Los valores del índice deben ser, en formato datetime, todos los días del año 2019. Los valores de la única columna deben ser los valores medidos de monóxido de nitrógeno en la estación 4 en cada uno de los días de 2019.

c. Representa los valores medidos de monóxido de nitrógeno en la estación 4 en una gráfica de líneas, de forma que se pueda apreciar la serie temporal.

¿En qué fechas se aprecia mayor contaminación por monóxido de nitrógeno?

d. Repite los pasos de los apartados a y b utilizando los valores de monóxido de nitrógeno medidos en las estaciones 8, 11 y 16, creando los DataFrames df_no_e8, df_no_e11, df_no_e16. Une todas las series temporales que has obtenido en un único DataFrame llamado df_no_total. Este DataFrame debe tener por índice todos los días del año 2019, en formato datetime, y debe tener cuatro columnas, representando los valores de monóxido de nitrógeno medidos en las estaciones 4, 8, 11 y 16.

e. Representa gráficamente el contenido del DataFrame df_no_total. Utiliza todas las opciones de representación (leyenda, títulos de eje, grid...), que consideres necesarias para comunicar adecuadamente el contenido de la gráfica.

f. ¿Están correlacionados los valores de la contaminación por monóxido de nitrógeno medidos en las diferentes estaciones? Basa tu respuesta en las conclusiones que saques de la gráfica que has representado en el apartado anterior. También, calcula la correlación numérica utilizando el método corr de Pandas. ¿Confirman los valores numéricos la conclusión que has sacado a partir de las gráficas?

2. Detalles de la entrega

- Las respuestas de la actividad se deberán entregar en un Jupyter Notebook en el que se haya respondido a cada apartado en una celda independiente, en el orden de las preguntas de este documento. El código de cada celda se debe poder ejecutar para comprobar las respuestas aportadas. Las preguntas teóricas se deben responder en una celda de formato texto.
- Subir de forma grupal el documento a la actividad en el campus virtual.