

Diplomatura en Business Analytics

Trabajo Final Introducción a Python

Facultad de Ciencias Económicas y Jurídicas

**GENERO, Pablo Ariel**

Docentes:

Cdor. Cristian Darío Ortega Yubro

Mgter. Gustavo Machín Urbey

Contenido del Trabajo Final Introducción a Python

[**Introducción**](#_dgpxhwbvi9s6) **2**

[**Desarrollo**](#_pp50yisgxmme) **4**

[**Conclusión**](#_dtc754ra7ztb) **7**

[**Anexo**](#_bwe44a3x3qfc) **8**

# 

# Introducción

En el armado del Trabajo Final del Módulo 7 he hecho uso del conjunto de datos provisto llamado “análisis facturas de ventas”. El mismo contiene información sobre la facturación de una empresa en el transcurso de los años 2018 y 2019.

El dataset original se compone de 9 columnas, las cuales hacen referencia a las facturas emitidas por la entidad. Las columnas “CANTIDAD”, “PRECIO” e “IMPORTEMONRANSACCIÓN”, contienen datos numéricos de tipo flotante (decimales – “float64”). El resto de las columnas contiene datos que en Python son leídos inicialmente como objetos (“object”), pero que representan distinta información: la columna “FECHA” contiene la fecha de la transacción; el campo “DOCUMENTO” tiene datos alfanuméricos que identifican a cada comprobante; el resto de las columnas poseen información textual propia de las operaciones realizadas (“ORGANIZACIÓN”, “TIPOFACTURA”, “PRODUCTO”, “MONEDA”).

A simple vista, se puede distinguir que la empresa en cuestión es un responsable inscripto, debido a que los comprobantes emitidos son del tipo A y B. El dataset contiene la información del cliente al que se le está facturando, así como también el producto, la cantidad vendida y la moneda de la operación, lo que puede ayudar a obtener varias conclusiones del análisis de esos datos.

Haciendo una lectura rápida de la información de la columna “PRODUCTO”, se entiende que la empresa está relacionada a la comercialización de productos y servicios a fines a la tecnología/computación.

# 

# Desarrollo

* + - 1. **Activar módulos a utilizar**

Para el desarrollo del trabajo en Colab, los módulos empleados son los siguientes:

• Numpy

• Pandas

• Matplotlib

• Altair

• Plotly

* + - 1. **Lectura de los datos y creación del dataframe (DF)**

Una vez habilitados lo módulos, procedo con la obtención de los datos que se encuentran almacenados en el enlace de GitHub. Para dar lugar al DF, se crea el objeto “facturas” al cual se le asigna a lectura del CSV cargado. Con la ejecución del objeto “facturas”, se obtiene una primera visualización de los datos con los que se va a trabajar. Podemos observar que la carga y lectura de los datos funcionó. También nos da una idea de la estructura del DF, y ayudados por el index (que se presenta con números enteros correlativos), podemos entender la dimensión y tamaño de este (159 filas x 9 columnas).

* + - 1. **Visualizando el DF**

Al haber ejecuta el DF, la vista resumida del mismo permite apreciar algunos elementos que podrían requerir algún tratamiento en los pasos siguientes del proceso. La columna “TIPO” no parece hacer referencia al tipo de documento emitido ya que repite en cada fila la palabra “Factura”. Por otra parte, la columna “CANTIDAD” esta expresada en formato decimal, lo cual no sería lógico teniendo en cuenta el tipo de rubro en el que se desempeña la empresa. La columna “PRECIO” posee decimales, lo cual parece ser irrelevante, y podría ensuciar la visual de alguna gráfica futura.

A continuación, se ejecuta la función “info()” lo que permite comprender y conocer un poco más los datos. Ahí vemos que la columna “FECHA” es un objeto, a pesar de que visualmente parece estar bien expresada. También sirve para confirmar el formato de los datos de las restantes columnas, y para asegurarnos de que no existen valores nulos.

Se ejecuta también la función “nunique()” para ver si los datos se repiten o no. Con esto confirmo que la columna “TIPOFACTURA” no contiene un dato relevante, ya que evidentemente repite la palabra Factura en todas las celdas. También se puede ver que en “MONEDA” hay otro valor además de “Pesos Argentinos”, ya que en la vista resumida del DF (5 filas primeras y últimas) no se alcanzaba a distinguir. La otra moneda es Dólares, y para conocerla agrupo los datos y armo un conteo en base a esta columna, que también me permite observar que solo 2 transacciones se han hecho en Dólares. Habiendo conocido esto, se podría seguir indagando más sobre estas 2 operaciones y sacar alguna conclusión que permita decidir sobre mantenerlas o no en el DF: *el importe que visualizamos en el DF ¿Son Dólares? ¿Están convertidos a Pesos? ¿Mantengo o elimino estas transacciones? ¿Podría mantenerlas y pesificarlas para que el análisis se haga en una misma moneda?, etc…*

* + - 1. **Trabajando con el DF**

A partir de lo interpretado al visualizar el DF, procedo a realizar algunas transformaciones en el mismo. Para empezar, la columna “TIPOFACTURA” no contiene un dato relevante y podría eliminarse. Pero si vemos en “DOCUMENTO”, además del número del comprobante se encuentra el tipo. Entonces lo que hago es extrae el primer dígito de esta última columna, y llevarlo a la columna “TIPOFACTURA”, reemplazando los valores preexistentes, con lo que estaría ahorrando la tarea de eliminar la columna y luego crear otra para alojar el tipo de comprobante A/B.

A continuación, sobre la columna “PRECIO” aplico un redondeo de los importes para que se vean solamente 2 decimales lo que aporta algo de prolijidad al DF, y ayuda en lo visual en caso de querer llevarlo a un gráfico.

En el momento que se ejecutó la función “info()” no se habían detectado valores nulos que puedan requerir alguna acción. Otra verificación que hago es revisar si las 3 columnas que contienen datos numéricos poseen valores en 0, porque que no sería lógico facturar a precio $0, o vender 0 unidades, o emitir una factura por $ 0.

Luego se ejecutan las funciones “dropna” y “drop\_duplicates” con el propósito de eliminar valores nulos o duplicados. Para el caso de los nulos, dijimos que según lo observado no había de estos en el DF. En cada caso, verifico por medio de “shape” si el DF continúa teniendo la misma estructura o si se vio alterado.

Lo que sigue es cambiar el tipo de dato de algunas columnas: “FECHA” esta como “object” y la llevo a formato fecha (“datetime64”), mientras que “CANTIDAD” esta como dato flotante y lo paso a número entero “int32” ya que es lo más lógico según el rubro en el que opera esta empresa. También se insertan dos columnas con valores que se generan a partir de “FECHA”, y que van a contener el número del año y el nombre del mes (“AÑO” y “MES” respectivamente). En esa misma celda de código el valor del año se deja con el formato “object”, que me va permitir darle uso a la hora de graficar de modo que sea identificado como una categoría (caso contrario la gráfica podría tomarlo como valor numérico y realizar alguna suma, lo que no sería muy útil).

En cuanto a los nombres de las columnas, se cambian “IMPORTEMONTRANSACCION” por “TOTAL” y “TIPOFACTURA” por “TIPO”, para tener un nombre más breve y fácil de interpretar.

Por último, habiendo realizado las transformaciones que consideré necesarias en el DF, reordeno el mismo en base a la fecha (me puede ser útil a la hora de graficar) y luego reinicio 00el valor del index para que se genere de acuerdo al nuevo orden.

* + - 1. **Gráficos en base al DF**

Teniendo el DF adecuado, paso a realizar algunos gráficos, para lo cual voy a estar usando las bibliotecas Matplotlib, Altair y Plotly.

El primero de los gráficos se construye con Matplotlib y lo utilizo para evaluar el peso de cada tipo de comprobante en la facturación, según el total de unidades comercializadas y según el total facturado. Para poder crear el gráfico previamente construí un sub dataframe que agrupa o suma las cantidades y el total facturado según el tipo A/B (también suma el precio, lo que no sería coherente, pero como no lo utilizo no me preocupo en quitarlo). A su vez, con el tipo de comprobante genero también un gráfico de líneas que permite evaluar la evolución de las facturas A y B a lo largo del tiempo que abarca el conjunto de datos (Plotly).

A continuación, en otro gráfico de líneas creado mediante Plotly analizo la evolución en el tiempo de la facturación según la moneda (ver comentarios en apartado *Conclusión*).

Al trabajar con el DF se había creado una columna con el nombre de los meses de las transacciones. En base a estos y mediante la librería Plotly, construyo dos gráficos del tipo Box Plot que permiten comparar el total facturado en cada mes y año, pudiendo detectar algunos valores atípicos.

Otro de los gráficos, y también mediante el uso de Plotly, representa la facturación total de cada mes y la compara entre los años 2018 y 2019. Para poder generarlo, como paso previo se agrupan los datos por mes y año, siendo muy importante que el año quede con formato “object” porque de lo contrario se toma como un valor numérico dando lugar a otro tipo de visualización. Aquí cree dos alternativas: una de barras paralelas, y otra de barras apiladas divididas según el año respectivo (a mi parecer el gráfico de barras paralelas es mejor para este tipo de análisis).

Por último, usando la biblioteca Altair, generé dos gráficos de barras horizontales que analizan los productos comercializados. En el primero se visualiza el importe total facturado, mientras que en el segundo se pueden ver las unidades vendidas, que en conjunto permiten hacer una evaluación sobre la importancia que puede tener cada producto en las operaciones de venta.

* + - 1. **Guardando el DF**

El DF construido con Colab, se almacena en la unidad de Google Drive de la cuenta provista por la facultad, y se puede acceder al mismo mediante el siguiente link:

[***https://drive.google.com/drive/folders/1YcHPfN2ETokhjyyi\_3TUoA5lzNVSSkWu?usp=sharing***](https://drive.google.com/drive/folders/1YcHPfN2ETokhjyyi_3TUoA5lzNVSSkWu?usp=sharing)

El link lleva a la carpeta COLAB\_PYTHON, y podrán acceder a la misma quienes también posean una cuenta con el dominio de la facultad. En esa carpeta también se ha almacenado el presente informe y el cuaderno Colab de Trabajo Final del Módulo 7.

# 

# Conclusión

En cuanto al trabajo realizado, uno de los puntos que considero interesante resaltar es la primera aproximación al DF generado. Si bien es importante trabajar los aspectos “técnicos” de nuestros datos para su correcto procesamiento en Python (como ser la lectura del index, saber si existen valores nulos, cual es tipo/formato de datos con que se trabaja), no hay que perder de vista el conocimiento de la realidad del negocio y su impacto en la calidad del conjunto. Con esto me refiero a lo que pude ver al controlar la existencia de valores únicos:

**facturas.nunique()**

DOCUMENTO 138

ORGANIZACION 20

FECHA 127

TIPOFACTURA 1

PRODUCTO 23

CANTIDAD 23

PRECIO 125

IMPORTEMONTRANSACCION 134

***MONEDA 2***

dtype: int64

Se puede ver que el conjunto de datos contiene 2 tipos de monedas, y tal como comenté más arriba, sólo 2 transacciones estaban en esta moneda, sin tener del todo claro (en base a la información de que dispongo) si los importes que se ven están expresados en Dólares o si están pesificados. Posteriormente al hacer el gráfico de la evolución de la facturación por moneda, se puede ver la incidencia de esas transacciones en el conjunto:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Considero que es muy importante poner atención a estos pequeños detalles, siendo necesario saber como atenderlos, y cuáles serían las consecuencias de esas acciones. De lo contrario, podríamos pasarlo por alto y continuar realizando transformaciones y análisis con un conjunto que no es homogéneo y que nos podría llevar a conclusiones o decisiones erróneas. Aquí nuestro set de datos posee 159 registros, siendo sólo 2 los que se encuentran en esta situación, lo cual no sería difícil de resolver y probablemente no tenga un impacto significativo. Pero si lo trasladamos a la realidad, los datos que puede manejar una empresa suelen contener miles de registros, en donde un pequeño error o imprevisto nos puede llevar a una distorsión y pérdida de calidad.

# Anexo

En este anexo se presenta información relacionada al cuaderno Colab con algunos códigos empleados que considero interesante comentar y que creo que pueden aportar valor a la hora de trabajar con un DF.

Para profundizar la consulta sobre las monedas de las operaciones, consideré oportuno realizar un “groupby”, que muestra cuales son las monedas y en cuantos registros tiene impacto. Luego, mediante un “query” al DF en base a la columna “MONEDA”, individualicé las dos operaciones en Dólares. Si el DF fuera más grande, y con un mayor volumen de operaciones, esta última consulta podría no adecuada:

Captura de pantalla de un celular

Descripción generada automáticamente

Otro código que creo que es muy útil, es la extracción de caracteres de una columna a partir de los cuales se puede crear un nuevo campo. Esta operación es algo que hago muy seguido con los datos de la empresa en donde trabajo:

Captura de pantalla de computadora

Descripción generada automáticamente

Para indagar sobre los valores de una columna, con el siguiente código se puede verificar cuantos registros cumplen con una condición, que para este caso debían ser diferentes a 0:

len(facturas[facturas.IMPORTEMONTRANSACCION!= 0])

Por último, veo mucha utilidad en el gráfico de Box Plot de Plotly, porque da la opción de definir el parámetro “hover\_name” con alguna de las columnas del DF. Esto sirve para tener mas información sobre los valores atípicos. Para el caso graficado, usé la columna “ORGANIZACIÓN”. Al pararme en el valor atípico, la etiqueta con información me muestra el nombre del cliente (ayuda a identificar el valor fácilmente en caso que requiera algún tratamiento particular, o podría ser útil para una mejor comprensión):

Diagrama

Descripción generada automáticamente