**Prediciendo ingresos tributarios en México**

**Organización de carpeta**

Camilo Arias

Verano 2019

Organización de la carpeta del proyecto de predicción de ingresos tributarios. De forma general, la carpeta se organiza de la siguiente manera:

**Descripción de contenido**

El contenido específico de cada carpeta se describe a continuación:

**/curso python:** Material que hemos cubierto en el curso de Python los martes y jueves.

**/inputs:** Carpeta con insumos para predicción. Todo lo que se encuentra en esta carpeta, en el primer nivel, fue descargado a mano. Lo que fue descargado y es descargado de forma automática se encuentra en la subcarpeta downloads.

* variables.xlsx Descripción de las variables que se han descargado y el método de descarga.
* tasas\_impuestos.xlsx: Excel con tasas históricas de IVA, ISRP e ISRE
* semana\_santa.xlsx Histórico anual con los meses en los que ha caído semana santa
* semana\_santa\_fin\_marzo.xlsx Histórico con los años en los que Semana Santa ha caído en los últimos días de marzo. Relevante porque en esos años, parte de la declaración del ISRE se captura entonces en abril.
* ingresos\_tributarios\_desglosados.xlsx: Excel con ingresos tributarios brutos. Fue construido por UPIT y lo actualicé con los últimos datos brutos.
* ingresos\_tributarios\_desglosados\_updated.xlsx: Excel con ingresos tributarios brutos. Tiene dato actualizado hasta junio. Fue construido por UPIT y lo actualicé con los últimos datos brutos.
* calendario\_lif.xlsx: Excel con el calendario de la Ley de Ingresos de la Federación desde 201t hasta 2019.
* ajustes.xlsx Ajustes que Paola y Mariana realizan a las series de ingresos para predecir (Ajustan eliminando montos que no corresponden. Por ejemplo, para unos años, eliminan del ISR lo que resultó de un programa de repatriación.)
* calendario\_lif.pdf: PDF del calendario de la LIF de 2019
* /bases\_iniciales: Material inicial compartido por la UPIT. Se encuentran do files, archivos de ingresos brutos y cálculos de elasticidades.
* /cuadros\_preliminares: Carpeta con los cuadros preliminares de la recaudación que son compartidos por el SAT en los primeros días del mes.
* */downloads* Carpeta con los CSV descargados de forma automática del INEGI, BANXICO, DATOS ABIERTOS y FED. El folder se llena al correr el Jupyter Notebook *download.ipynb* de la carpeta ../scripts/

***/Scripts*:** Carpeta con scripts de Python y con *Jupyter Notebooks* ejecutores.

SCRIPTS (Archivos con terminación .py):

* *config.py* Script con los datos necesarios para las descargas de las API's. Incluye datos como token de descarga, identificadores de las variables a descargar, y nombres de las variables. Es usado por el script download.py.
* *download.py* Script con funciones para descargar los datos y para cargar los datos a Pandas DataFrame. El script depende de config.py principalmente. En la parte de descarga se concentra en 4 funciones: *get\_files\_inegi(), get\_files\_datos\_abiertos(), get\_files\_banxico(),* y *get\_files\_fed().* Cada una de ellas, a su vez, llama a otra función que es la que se conecta con la API. Las funciones de carga son varias, y todas comparten varios puntos en común: Leen un archivo excel o csv, y retornan un Pandas DataFrame. Algunas retornan el DataFrame con valores nominales y reales (*load\_ingresos\_fiscales..., load\_balanza\_comercial*...) otras pueden convertir datos trimestrales a mensuales si se especifica (*load\_pib*), otras solo cargan indicadores de INEGI, de BANXICO o de la FED. Finalmente, hay dos funciones que sirven para extraer datos de los cuadros de SHCP: *extract\_from\_cuadro\_preliminar* y *extract\_from\_cuadro\_isr\_iva\_ieps*.
* *descriptive.py* Script que contiene funciones esenciales para describir los datos y transformar los datos. Las funciones principales son:
  + *Plot* series. Ampliamente usada para graficar en varios de los Jupyter Notebooks.
  + *transformation:* Para hacer transformaciones a las variables.
  + *revert\_transformation:* Para revertir transformaciones
  + *cross\_tab*: Para hacer tablas comparativas por año de variables. Sirve para ver comportamiento de compensaciones y devoluciones, de ISR bruto, etc.
  + *cross\_tab\_lif*: Similar a la anterior, pero especifica para hacer comparaciones con la Ley de Ingresos.
* *models\_multivariate.py*: Script con las funciones para hacer estimaciones. La función que arropa todo el script es *run\_predictions*. Esta función toma varios argumentos, como el nombre del modelo a estimar, la lista de parámetros, los inicios y finales de las predicciones, el DF, una lista con las columnas endógenas, etc. Esta función, recibe los datos, crea un DataFrame con las variables exógenas dependiendo de los parámetros que el usuario haya incluido, y después hace un *loop* para cada una de las especificaciones en la lista de parámetros, correr un modelo predictivo. Dependiendo del nombre del modelo, usa a las funciones *predict\_with\_econometric\_model* o *predict\_with\_ml\_model*. Las funciones *predict\_with\_econometric\_model* y *predict\_with\_ml\_model* son similares en que recibe los datos, crean un DataFrame con los datos endógenos y exógenos, dividen los *DataFrames* en train y en test usando la función *split\_train\_test*, dividen los parámetros del modelo entre parámetros que se deben pasar al llamar *.fit()* o al construir el modelo y obtienen las predicciones llamando la función *run\_model*. Se diferencian en un aspecto clave: Mientras los métodos econométricos ya contaban con un método que hacia las predicciones paso a paso y de manera recursiva, los métodos de machine learning no, y hubo que implementar estos pasos.
* *grid.py* Script que sirve para construir el *grid* completo de posibles parámetros de cada modelo. Es llamado por el *Jupyter Notebook* de predicciones. Tiene tres tamaños de grid: *small, medium y big*.
* *dashboard.py*: Código para crear *dashboard* usando *dash* y *plotly*. El código que tiene los botones interactivos y las gráficas interactivas del *dashboard* que hemos visto.
* *dashboard\_aux.py*: Script auxiliar del *dashboard*. Tiene las listas de variables necesarias para crear el *dashboard*.

JUPYTER NOTEBOOKS (Terminación .ipynb):

* *download.ipynb: Jupyter notebook* para hacer descarga de datos de INEGI, BANXICO, FED y Datos abiertos (Estadísticas oportunas)
* *Estimacion\_marco\_macro.ipynb: Jupyter notebook* con estimaciones del marco macro mexicano y de Estados Unidos.
* *Estimaciones\_cierre.ipybn: Jupyter notebook* con estimaciones finales de ingresos tributarios. Usa modelos seleccionados por Pipeline\_analyze.ipynb
* *Machote\_eficiencia\_recaudatoria.ipynb: Jupyter notebook* Machote para hacer análisis de eficiencia recaudatoria, así como cambio anual en ingresos tributarios, cambio en compensaciones y predecir el PIB.
* *Pipeline\_descriptive.ipynb: Jupyter notebook* con descripción de variables recaudatorias, análisis de estacionariedad y descomposición en ciclo y tendencia.
* *Pipeline\_multivariate\_descriptive.ipynb: Jupyter notebook* con análisis descriptivo de recaudación y otras variables. Se analiza comportamiento de variables Macro de USA y de México.
* *Pipeline\_univariate\_predict.ipynb Jupyter notebook* para hacer predicciones de ingresos tributarios sin covariables. Fueron las primeras estimaciones hechas. No son las estimaciones usadas en el reporte final.
* *Pipeline\_multivariate\_predict.ipynb: Jupyter notebook* donde se corren todas las estimaciones. Este es el notebook que utiliza el script models\_multivariate.py de manera importante, y estima todos los modelos para cada uno de los cortes transversales. En este notebook se producen los resultados.
* *Pipeline\_analyze.ipynb Jupyter notebook* donde se leen los csv de resultados y se obtienen los mejores modelos. Se hacen gráficas de mejores modelos y se seleccionan los modelos que después se utilizan en estimaciones de cierre.

SUBCARPETAS

* **/reportes\_eficiencia: Carpeta con reporte de eficiencia recaudatoria (Versiones estáticas del Machote de Eficiencia)**
* **/assets: Carpeta con imágenes del dashboard**
* **/estimaciones: Estimaciones de cierre de los ingresos tributarios**

**/figures** Gráficas generadas en el análisis de recaudación. Tiene gráficas que resultaron de *Machote\_multivariado.ipybn*

**/Presentaciones eficiencia** Presentaciones de eficiencia recaudatoria

**/Presentaciones\_prediccion** Presentaciones del proyecto de predicción. Incluye la presentación de avance con el BID, presentación inicial con Víctor Higo y presentación final.

**/reporte** Carpeta con reporte final. Incluye archivo LaTeX (.tex), archivos intermedios (.aux, .log y .out) y archivo pdf, así como carpeta con imágenes.

**/results** Carpeta con resultados de las estimaciones realizadas. Son archivos .csv que después son importados en el *Jupyter* *Notebook* *Pipeline\_analyze.ipynb*

* Calendario\_lif.pdf Calendario de la Ley de Ingresos de 2019
* README.md archivo de texto similar a este para si presentación en *GitHub*.
* Cronograma.xlsx Aproximación del cronograma seguido en el proyecto.
* Requirementes.txt: Liberias que se requieren para correr todas las funciones