# BUFFERS BASICOS

## get\_data.py

Realiza la primera extracción de datos del SQL Server. Lo primero que detecta es la semana actual (la semana en la que se está ejecutando el programa) y a partir de ahí hace la extracción histórica, el limite de la extracción es una variable de esta función.

Todas las consultas son filtradas con un JOIN con la vista PadresClones\_Basicos, toma en cuenta los padres con etiqueta básico y todos sus clones, aunque estos estén dados de alta como van o res.

Comienza con la consulta de los inventarios [Flash\_BC].[dbo].[InvHistBC], acotada entre las fechas acordadas, se realiza un poco de limpieza para que las semanas coincidan con la tabla de ventas ya que invhist se encuentra desfazada.

De la misma manera se consultan las ventas históricas [Flash\_BC].[dbo].[VentaVinTBC], algunas operaciones se hacen en Python para restar trabajo al server de sql.

Después de hacer un cruce de esta información, se realiza la siguiente consulta [Allocations].[dbo].[Item\_BC], llamada clones ya que su propósito es detectar los padres de todos los productos y agruparlos, aprovechando la extracción se consulta también las características de los productos, como lo son su división code, descripción, producto group code item category code y weather.

Posteriormente comienza la preparación de la data,

* Se transforman los tipos de datos para que coincidan
* Se coloca domingo como día corte para el análisis
* Creación de un archivo Excel con datos de ventas positivas e inventarios negativos para su posterior análisis ("inv\_negativos\_ventas\_positivas\_"+BU+".xlsx").
* Limpieza, si se tiene venta 0 y el inventario es negativo, cero o nulo se elimina de la tabla base que se estará utilizando
* Eliminando tienda T000 y T085
* Determina si se cuentan con semanas suficientes de historia para poder realizar una proyección, el mínimo de semanas es un input de la función, si e así se elimina de nuestra tabla base
* Cruce con las características de las tiendas:
  + Se elimina de la tabla base todas las tiendas que están cerradas
  + ShMan: hay productos que tienen historia de ventas en tiendas que no tienen shman la principal razón es por Coutumer Order, con excepción de la categoría BOTTLES M
  + Algo similar con Candy. Si existen estos casos se crea un Excel con los datos de estos productos en sus respectivas tiendas, para tenerlo mapeado.

Lo que finalmente retorna es una tabla con los inventarios y ventas históricas.

# CreateProyeccionActual.py

El propósito de este script es determinar la semana en la que se esta ejecutando el programa, con ello detecta cual son las 4 semanas anteriores y las 8 semanas siguientes, todo lo coloca en un rango, se evalúan cada uno de los elementos de este para decidir que tipo de proyección se realizará:

* Year\_Anterior: el cálculo se hará con datos del año anterior ya que algún valor dentro del rango esta dentro de la temporada alta

### TemporadaAlta.py \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Se obtienen los datos limpios con ayuda del auxiliar get\_data.py a partir de 2021-01-10 hasta el día en el que se cree la proyección.

Realiza un cruce con uno de los inputs que es el archivo Excel de LeadTime (CEDIS - Tiendas) para realizar la cobertura, convertido a semanas

Ahora comienza a analizar las ventas del año anterior para realizar la proyección al día de hoy.

\*\*\* Si no se cuenta con datos del año anterior queda vacío, esto posteriormente se corregirá un algoritmo de clusters en el que ya se esta trabajando. Por el momento se envía a un archivo Excel llamado no\_data +BU.xls los productos qu no tienen datos en su determinada tienda, esto puede pasar por aperturas.

Finalmente, retorna una tabla por cada producto, en cada tienda y su determinada proyección.

### TemporadaNormal.py \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Igual que el script anterior, utiliza get\_data pero esta vez a partir de 2022-01-09, se decidió esta fecha ya que al momento de hacer la limpieza de nulo inventario pueden que queden muy pocos datos si el rango es corto.

Se seleccionan las últimas 4 semanas de venta que si contaron con inventario y cumplieron con las características de este tipo de temporada y se realiza la proyección actual (ventas promedio)

Todo se resume en una tabla con cada producto, en cada tienda, con su determinada proyección.

El cálculo de la proyección actual puede tomar mucho tiempo, y ya que se realiza con datos históricos se sugiere hacer una vez a la semana.

# BuffersBasicos.py (main)

A partir de la obtención de la fecha actual se realizan los cálculos para el cálculo del bufer actual.

Lo primero que realiza es una consulta de la proyección actual, hace una consulta las bases de datos para saber el inventario actual de los productos.

Consulta el input de Mínimos\_Maximos para obtener estos, además de sus múltiplos, después se agrupa a nivel padre. Se agregan datos de las tiendas y se limpia en caso de ser necesario.

La categoría candy tiene una cobertura especial de 6 semanas en tiendas normales y de 5 en tiendas hot

A partir de aquí comienza el calculo del buffer:

* La proyección actual se redondea (número mas cercano) a los múltiplos de distribución de cada producto
* Si la proyección actual es menor al mínimo de tienda, la tienda tendrá el mínimo
* Si la proyección es mayor al máximo se coloca el máximo
* En cualquier otro caso se conserva la proyección actual
* Si no existe alguna de esta información el programa se detiene y manda un mensaje de alerta con los productos a los que les hace falta información
* Posteriormente se te añadé la proyección creada con la RNN y se calula su buffer de la misma manera (se crea una flag por si la proyección llega a ser muy elevada y si no hay datos)

Se agregan las características de los productos para un mejor análisis

Consultan los on order de las bases de datos en el rango de fecha a partir de el día en el que se ejucute el programa hasta la cobertura. Se agrega a nuestra tabla base desglosado por fecha en la que se espera que llegue recibo.

Se adjunta igualmente la matriz de cada producto para así poder consultar del framework el RES LEAN

Igualmente, del framework se extrae el buffer anterior/actual, se calcula la rotación proyectada (Nuevo\_Buffer / Ventas promedio) y WOS (InventarioActual / Ventas Promedio)

Se calcula la diferencia del buffer y el buffer anterior y se divide entre el múltiplo, esto para ver si hay mucha diferencia por ‘caja’

Finalmente, todo se agrupa a nivel producto y se crea el archivo resumen, ambas tablas se les aplica el archivo de buffers dinámicos y se envía a un Excel en la carpeta output.