TP Final
Digital House – Data Science

Grupo 1:

- Gonzalo Barbot - Project Manager
- Agustin Stigliano - Code Development Manager
- Fernando Dupont - Presentation Development Manager

Pronóstico sobre Series Temporales

Temas

- **Introducción**
- **DATA SCIENCE: Aplicación de Data Science en el DataFrame**
  - **Definición del intervalo temporal de estudio**
  - **Categoria prioritaria: Requisito 1**
  - **Categoría prioritaria: Requisito 2**
  - **Exportación de los targets para proyectar**

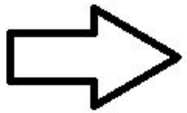
- **MACHINE LEARNING: Implementación de procesos de Machine Learning**
  - **APP1: Función de evaluación de los modelos sobre set de entrenamiento**
  - **APP2: Implementación y exportación de modelos predictivos sobre datos modelizados**
  - **APP3: Función para visualizar las proyecciones de los modelos de forecast**
- **Conclusiones**

Introducción

La propuesta apunta a la implementación de *Proyecciones y Pronósticos de Series Temporales con Machine Learning* (**PPSTML**), y tiene el objetivo de ofrecer información de valor para analizar por el sector comercial de la empresa. Obtuvimos la **base de datos** sobre sus **ventas** de productos pertenecientes a 19 diferentes categorías. El período de comienzo a fin es desde Enero 2012 hasta Septiembre 2013. A continuación mostramos los datos originales y el tratamiento que se les dio para diseñar los modelos predictivos.

date	customer_id	item_id	quantity	selling_price	other_discount	coupon_discount	category	Ventas
2012-04-10	436	26830	1	45.95	0.00	0.0	Natural Products	45.95
2012-04-10	841	26830	1	45.95	0.00	0.0	Natural Products	45.95
2012-04-12	68	26830	1	45.95	0.00	0.0	Natural Products	45.95
2012-04-12	111	26830	1	45.95	0.00	0.0	Natural Products	45.95
2012-04-12	999	26830	1	45.95	0.00	0.0	Natural Products	45.95
...	...	...	...	...	...	...	...	...
2013-06-30	1129	1736	1	95.82	0.00	0.0	Grocery	95.82
2013-06-30	1129	2423	1	81.57	-7.12	0.0	Grocery	74.45
2013-06-30	1129	2777	1	284.60	-71.24	0.0	Grocery	213.36
2013-06-30	1129	2953	4	42.74	-28.50	0.0	Grocery	142.46
2013-06-30	1129	2971	6	64.12	-42.74	0.0	Grocery	341.98

1216481 rows × 8 columns



date	category	Alcohol	Bakery	Dairy, Juices & Snacks	Flowers & Plants	Fuel	Garden	Grocery
2012-04-10		303.92750	161.637857	196.454737	320.220000	9.632366e+06	142.12	125.835760
2012-04-11		564.39500	123.934848	107.352059	498.263333	8.645962e+06	142.12	146.691058
2012-04-12		778.18000	158.606944	116.793704	438.720000	8.897399e+06	213.36	164.758837
2012-04-13		303.88125	182.599318	133.110698	124.310000	3.894734e+07	NaN	152.669070
2012-04-14		213.36000	125.618571	206.520000	355.960000	9.133516e+06	NaN	126.721540
...		...	...	...	...	...	...	...
2013-06-26		266.79000	149.655882	181.002391	186.825000	1.323805e+07	NaN	112.263581
2013-06-27		213.36000	151.791714	576.267143	462.700000	1.222858e+07	9617.40	120.195221
2013-06-28		247.20000	116.581364	170.377679	142.120000	9.141931e+06	1484.64	130.841667
2013-06-29		351.57000	107.375172	192.414375	177.740000	9.903527e+06	605.54	121.465289
2013-06-30		284.60000	155.017111	129.345897	NaN	1.412554e+07	1039.21	110.923770

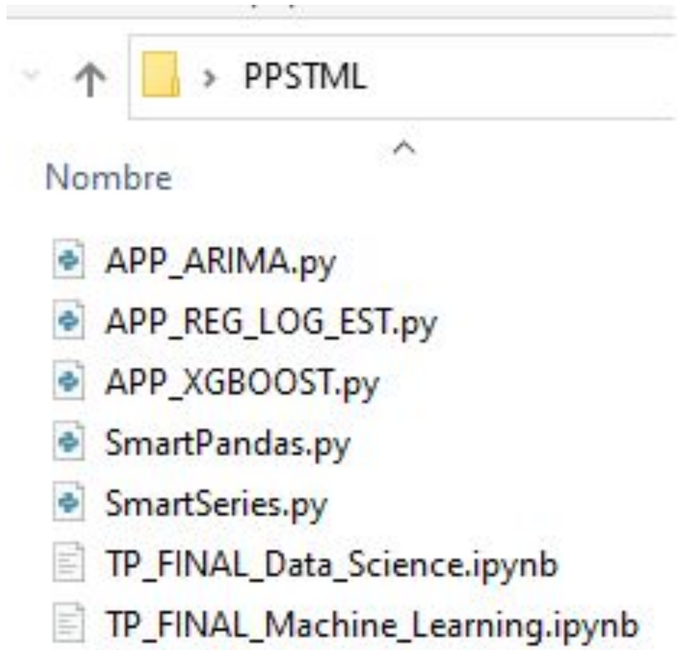
447 rows × 19 columns

Introducción

A continuación veremos el diagrama de flujos de datos, las librerías y documentos que fue necesario desarrollar para completar con el proyecto en cuestión.

Son:

- 2 Jupyter notebooks
- 5 documentos Python
- 2 datasets de un ejercicio (de Kaggle)



datasets: www.kaggle.com/datasets/vasudeva009/coupon-redemption-smote-feature-selection

Diagrama de flujos de datos de PPSTML

Librerias:

SmartPandas.py
SmartSeries.py

Modelos:

Regresiones lineales transformadas con estacionalidad (statsmodels)
test de Dickey-Fuller y proceso de diferenciación optimizado (statsmodels)
Adaptación de series temporales a un problema predictivo (xgboost)

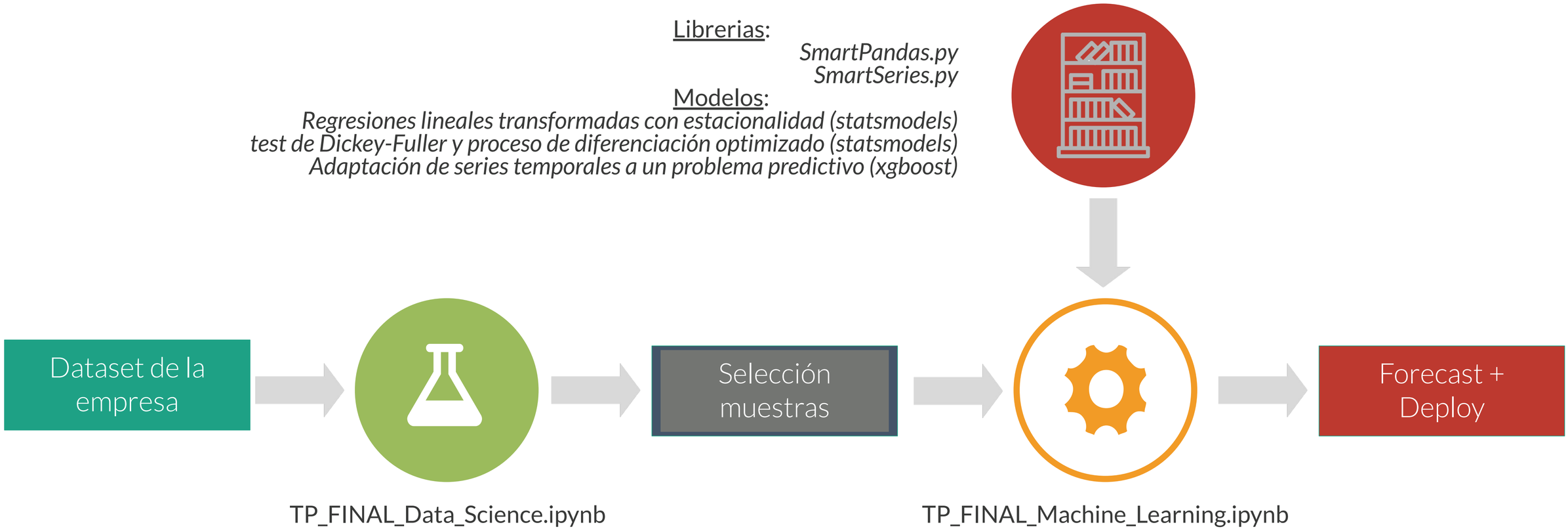


Diagrama de flujos de datos de PPSTML

Librerias:

SmartPandas.py
SmartSeries.py

Modelos:

Regresiones lineales transformadas con estacionalidad (statsmodels)
test de Dickey-Fuller y proceso de diferenciación optimizado (statsmodels)
Adaptación de series temporales a un problema predictivo (xgboost)

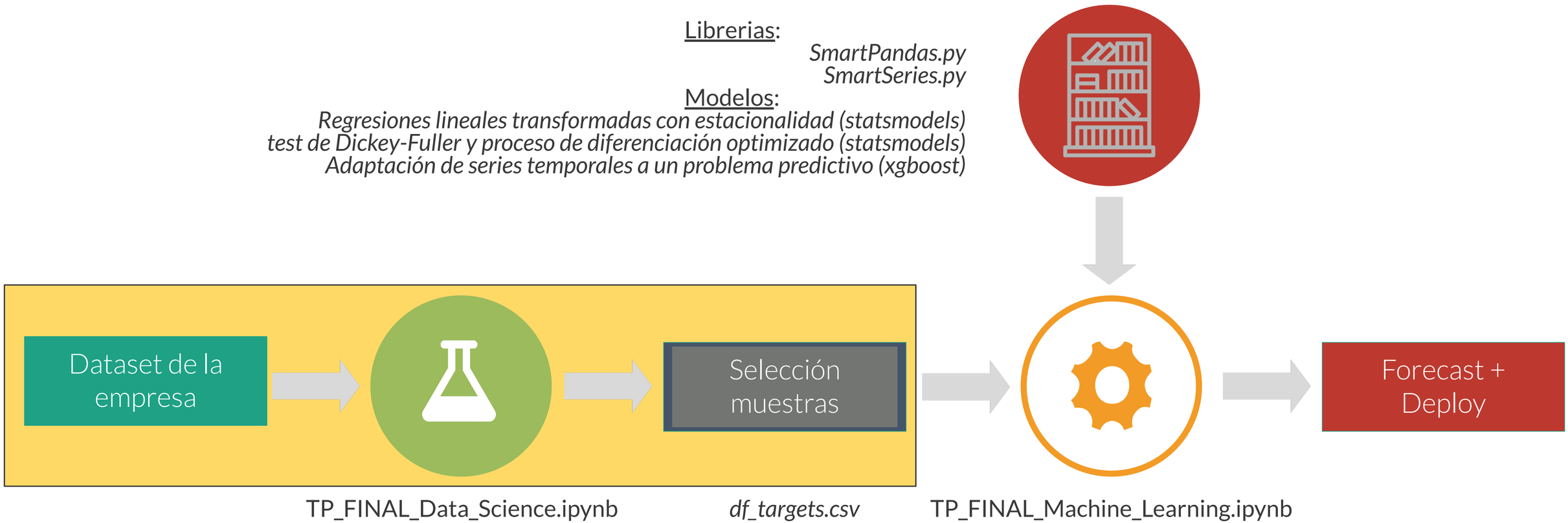


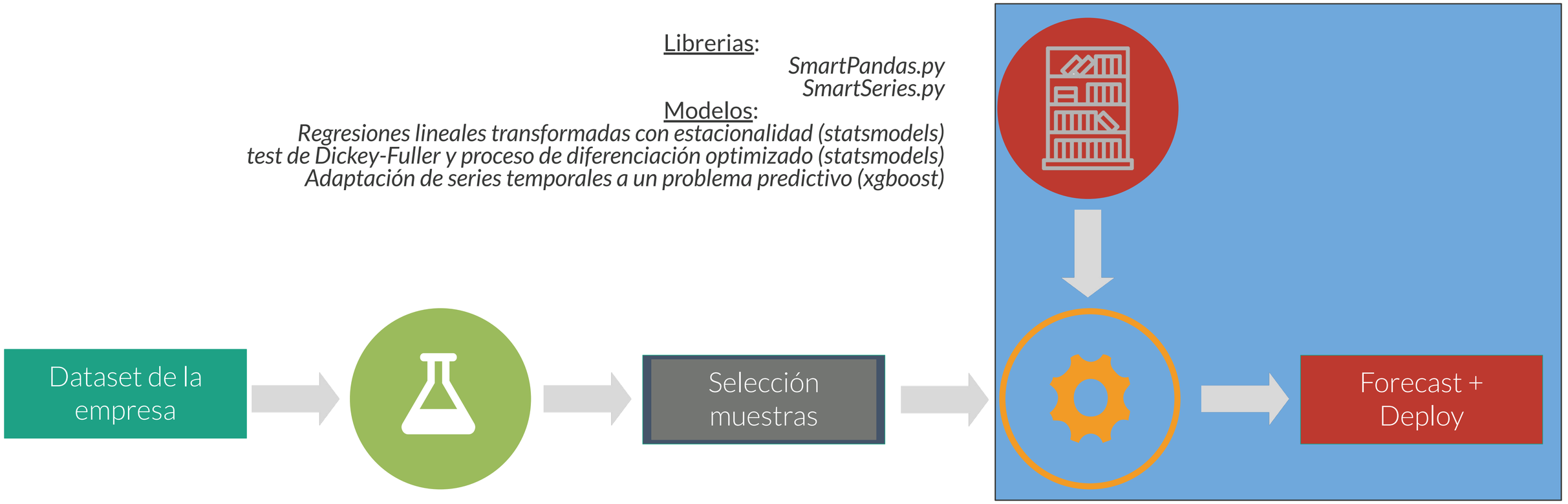
Diagrama de flujos de datos de PPSTML

Librerias:

SmartPandas.py
SmartSeries.py

Modelos:

Regresiones lineales transformadas con estacionalidad (statsmodels)
test de Dickey-Fuller y proceso de diferenciación optimizado (statsmodels)
Adaptación de series temporales a un problema predictivo (xgboost)



TP_FINAL_Data_Science.ipynb

TP_FINAL_Machine_Learning.ipynb

Data Science

TEMAS:

- **Definición del intervalo temporal de estudio**
- **Categoria prioritaria: Requisito 1**
- **Categoría prioritaria: Requisito 2**
- **Exportación de los targets para proyectar**

Introducción

El objetivo del trabajo fue desarrollar modelos de proyección que realicen pronósticos acertados de manera bimensual sobre las ventas. Para cumplir con el cometido fue necesario diseñar los modelos, testearlos y optimizarlos sobre un primer set “forecast on test”, con el que contamos con los datos originales de reserva para poder evaluar la performance.

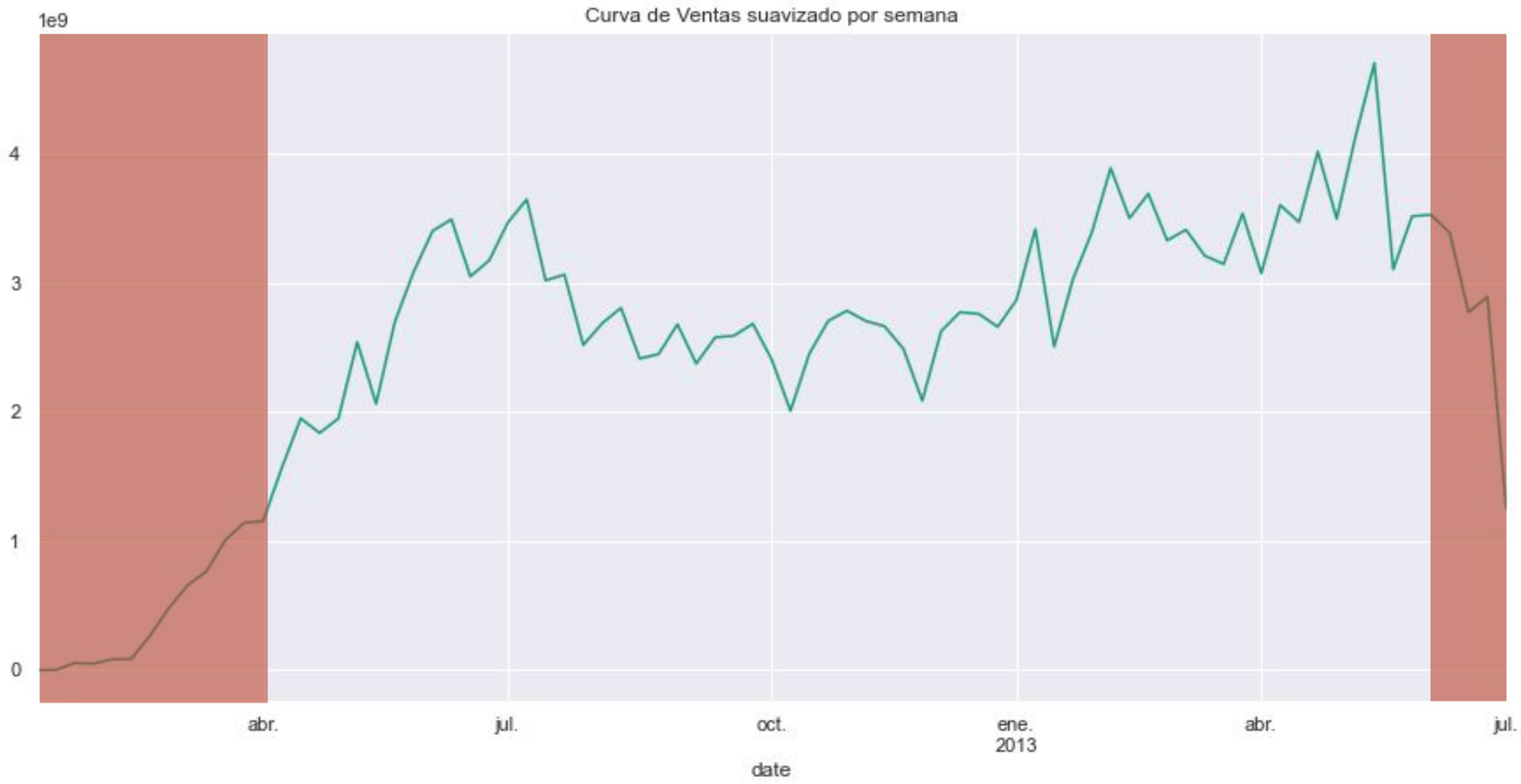
Por eso, para arrancar, es importante hacer hincapié en los **datos**, antes de pasar a los **modelos**, ya que se alimentan de ellos.

En una primera instancia se realizó un estudio para la obtención de información y se definieron **criterios de prioridad** que deben cumplir las categorías para pertenecer a la **selección de las muestras**. Existen dos tipos de muestras “alta prioridad” e “interés particular”.

Cuando las muestras son correctas y los modelos funcionan bien en test, el algoritmo PPSTML ofrece la posibilidad de modelizar datos futuros en base a registros históricos e implementar modelos predictivos sobre esos datos modelizados. Generando pronósticos y perspectivas de comportamiento que de cumplirse en el futuro implicaría la identificación de patrones de comportamiento por parte de estos modelos.

Compensación por datos faltantes

En el gráfico a continuación existen dos intervalos de tiempo que no se recopilaron datos sobre algunas categorías(rangos sombreados). Por lo tanto, hay ciclos transitorios dentro de la curva que no son representativos del valor de las ventas. Para igualdad de condiciones se excluyeron dichos rangos.



Rango a considerar
Abril 2012 a Julio 2013

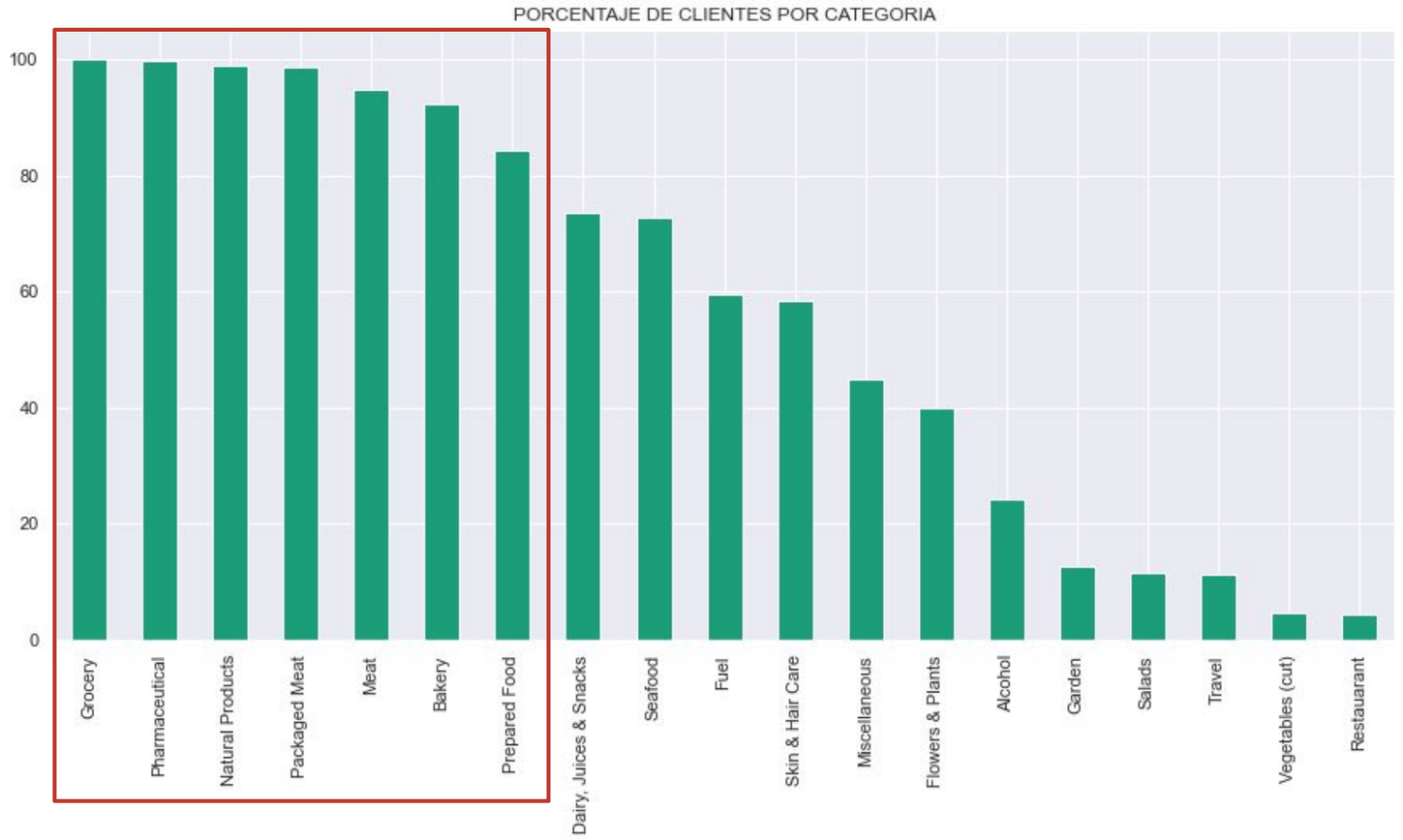
Exploracion dataset

Categorías prioritarias: Requisito 1

En base al análisis definimos una primera condición para definir una categoría como prioritaria. En este caso las categorías de alta prioridad son aquellas que superen un 85% del total de clientes con el que comercializa la empresa.

Categorías prioritarias 1

- Grocery
- Pharmaceutical
- Natural Products
- Packaged Meat
- Meat
- Bakery
- Prepared Food



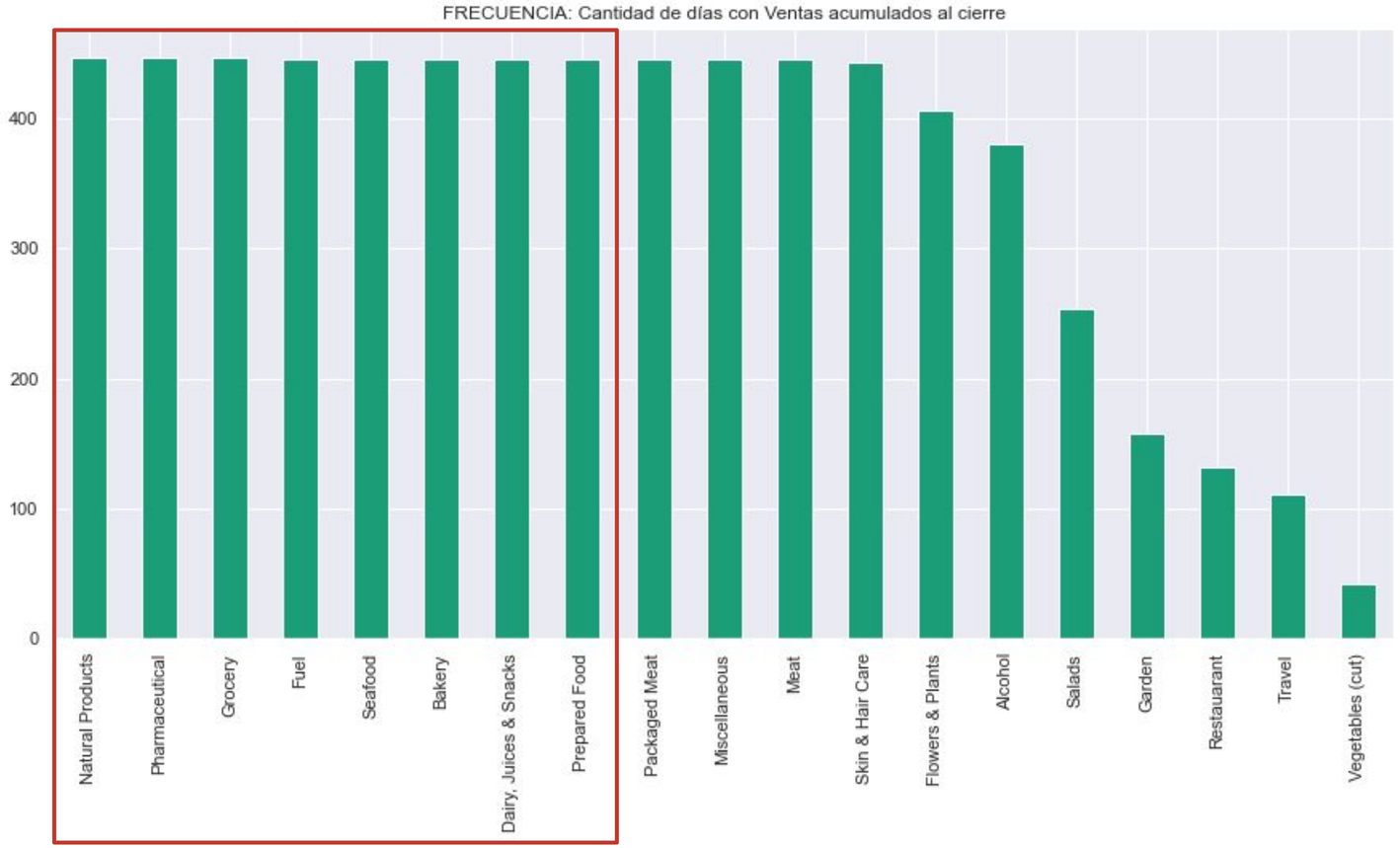
Exploracion dataset

Categorías prioritarias: Requisito 2

Como veremos a continuación hubo dos intervalos de tiempo (al comienzo y al final del período de estudio) que no se recopilaron datos sobre algunas categorías. Luego de un recorte para uniformizar los datos por categoría se tomó como criterio de prioridad 2 que la categoría tenga una frecuencia de ventas del 99% del período en cuestión.

Categorías prioritarias 2

- Natural Products
- Pharmaceutical
- Grocery
- Fuel
- Seafood
- Bakery
- Dairy, Juices & Snacks
- Prepared Food



Exportación de la muestra de datos

La muestra seleccionada se exporta mediante formato .csv para poder ser procesada luego por la notebook de **Machine Learning**.

A continuación veremos la curvas de venta en temporalidad semanal de las categorías en cuestión, estas son:

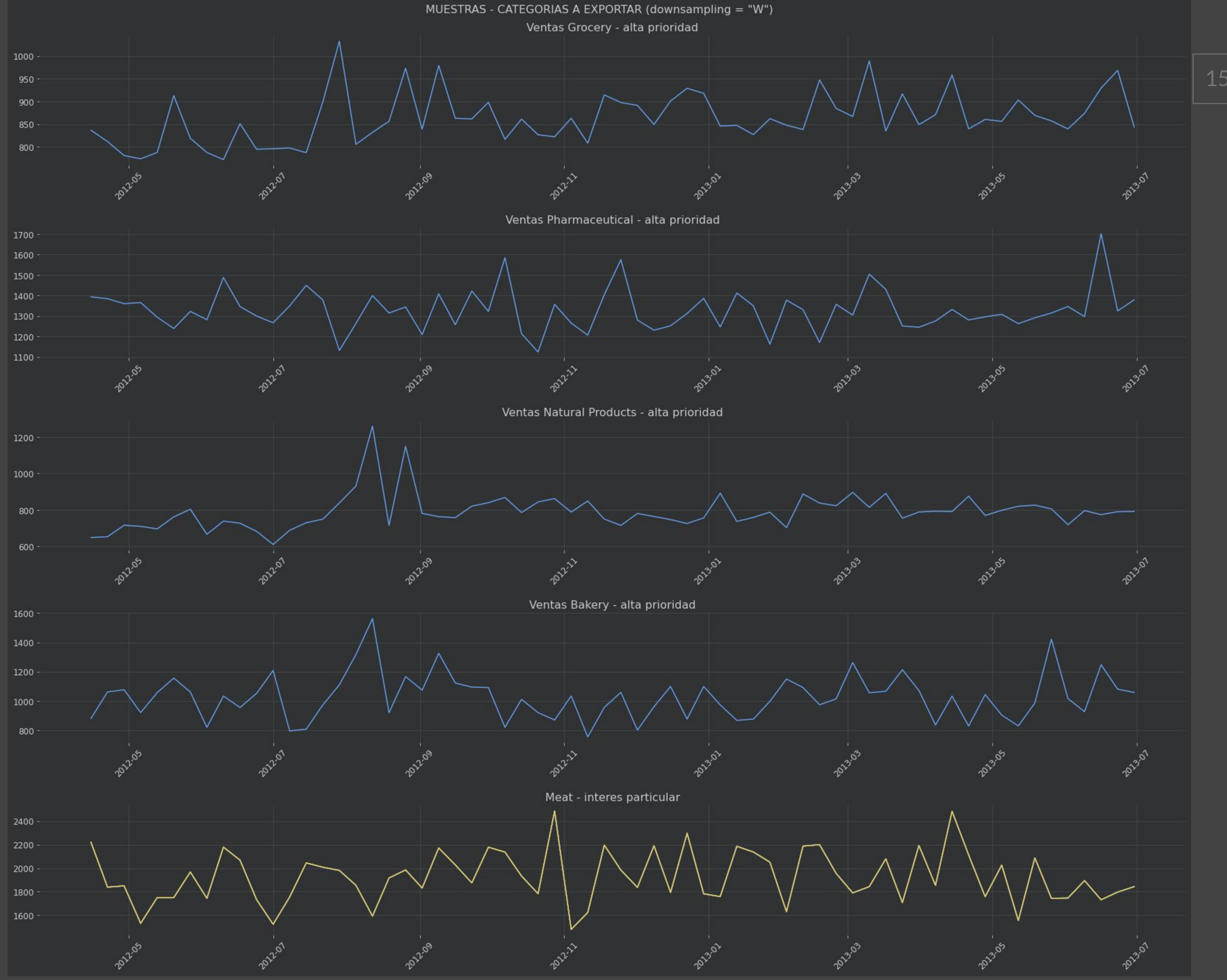
- Almacén
- Farmacia
- Productos Naturales
- Panadería
- Carnes

SmartSeries.py

Tiene la función

apply_dark_mode()

Se obtienen gráficas con fondo oscuro



Machine Learning

TEMAS:

- **Función de evaluación de los modelos sobre set de entrenamiento**
- **Implementación y exportación de modelos predictivos sobre datos modelizados**
- **Función para visualizar las proyecciones de los modelos de forecast**

Introduccion

En esta sección vamos a desarrollar una explicación de la finalidad de los modelos diseñados, hay de dos tipos, modelos para evaluar y modelos para generar pronósticos. Los modelos de evaluación son, descomposición de series temporales, análisis de series estacionarias, y medias móviles, y para pronosticar usamos regresiones lineales, cuadráticas, con transformación logarítmica y estacionalidad, junto con una adaptación del problema para poder implementar XGBoost a los pronósticos.

El potencial de desarrollo se centró en 2 funciones Forecast 1 y Forecast 2:

F_1) Evaluar los modelos mediante forecast en “test set” (seteo manual - umbrales)

F_2) Pronóstico de modelos mediante forecast en “real set”

Forecast 1:

Se definen umbrales manualmente y se obtiene la implementación de los modelos sobre **una categoría** de la selección de muestras.

Forecast 2:

Tiene 2 finalidades:

2.1) Exportar los modelos bimensuales de pronóstico en formato “.pickle”

2.2) Visualización de los pronósticos realizados por los modelos

Forecast 1: regresión lineal

Los umbrales son:

```
############################################################
umbral = 'Bakery' # category to implement forecast
umbral_2 = 60 # test_size for train test split
umbral_3 = 'RMSE' # es para evitar fallas
```

En este caso de las selección de muestras escogemos “Panadería”

Se realiza un train_test_split con test_size = 60

Se entrenan los modelos de regresión lineal evaluar la performance de ellos:

	Modelo	RMSE
0	log_trend_est	66.5422
1	log_trend_est_sq	66.2502
2	log_trend	69.6893
3	log_trend_sq	69.3841
4	lin_trend	69.0055
5	lin_trend_sq	68.9522

```
check_optimized_model(results_reg_log_est, umbral_3)
```

✓ 0.7s

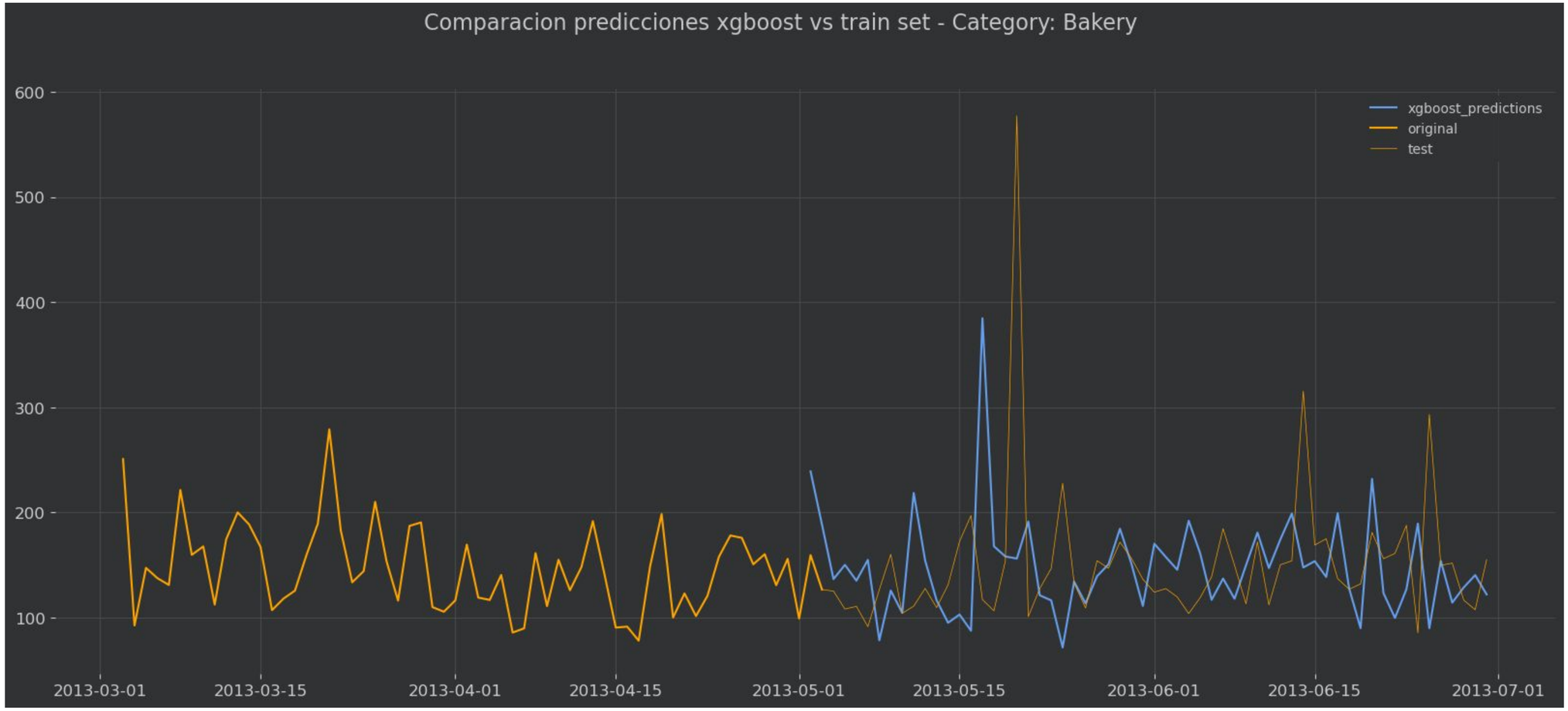
	Modelo	RMSE
1	log_trend_est_sq	66.2502

En base a las predicciones del modelo de regresión que mejor ajusta se procede a realizar un proceso de diferenciación para obtener el residuo de las predicciones y evaluar un test de serie estacionaria.

Forecast 1: xgboost

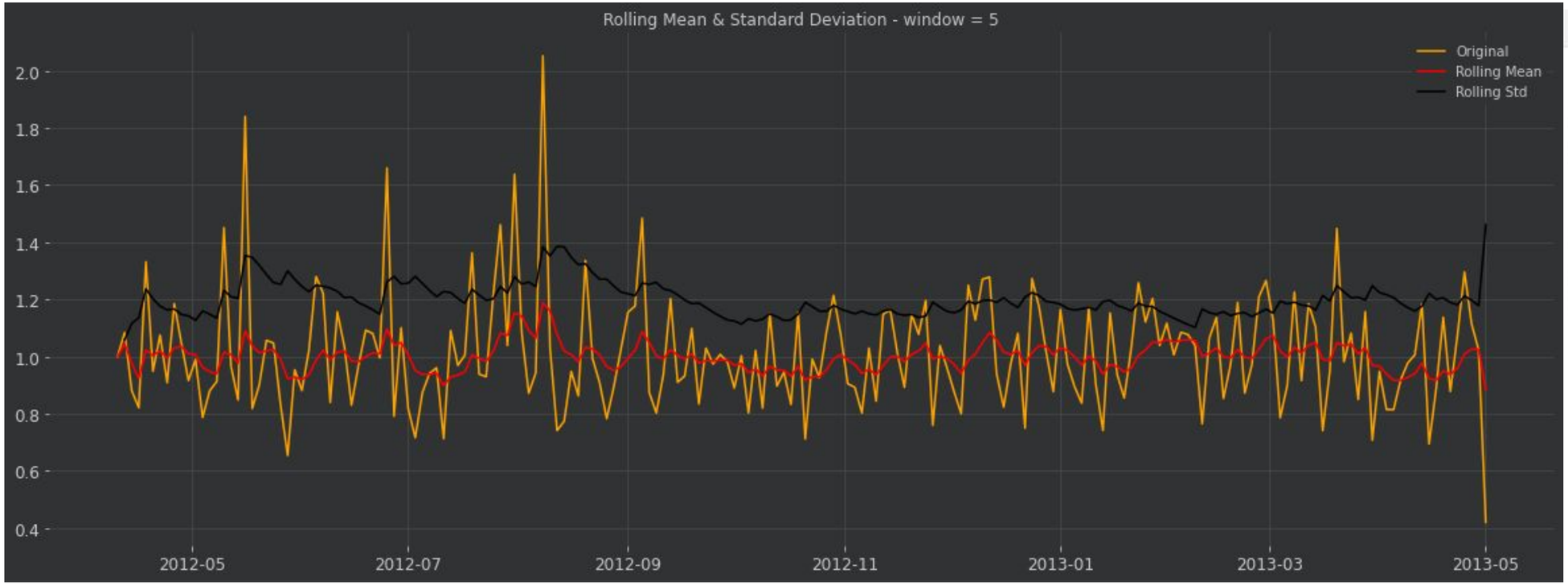
El modelo xgboost aplicado a series temporales no es un modelo que se pueda almacenar entrenado, un modelo “naive” que itera según la cantidad de pasos (umbral_2) que debe proyectar y en cada iteración instancia, entrena y predice un modelo nuevo.

Continuando con el ejemplo de “Bakery” podemos ver el tipo de respuesta que nos ofrece:



Forecast 1: estudio series estacionarias

El código también permite aplicar procesos de diferenciación automatizados dentro de una función y evaluar el valor obtenido por medio de un test de Dickey-Fuller. Luego se procede a optimizar el proceso que haya obtenido menor p-value, de esa forma es posible obtener un valor para la media móvil que mejor ajuste al comportamiento estacionario de la curva:

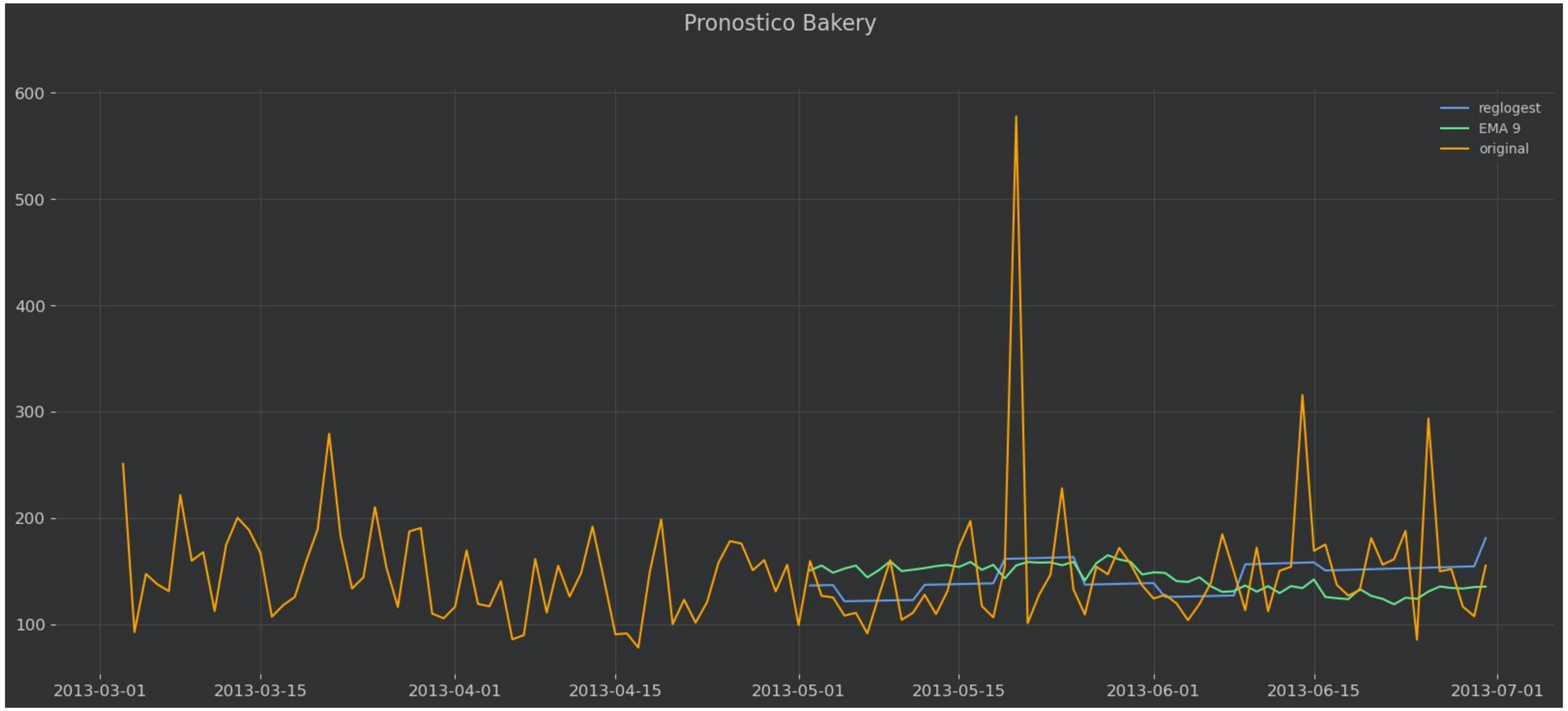


✓ 0.3s

```
        DIFFERENTIATION  PROCESS
Selected Method: EMA
Optimized Params: 5
        DICKEY-FULLER TEST
ADF Statistic: -13.467993036870597
p-value: 3.426302209517956e-25
Critical Values:
        1%: -3.4646940755442612
        5%: -2.8766348847254934
        10%: -2.5748163958763994
```

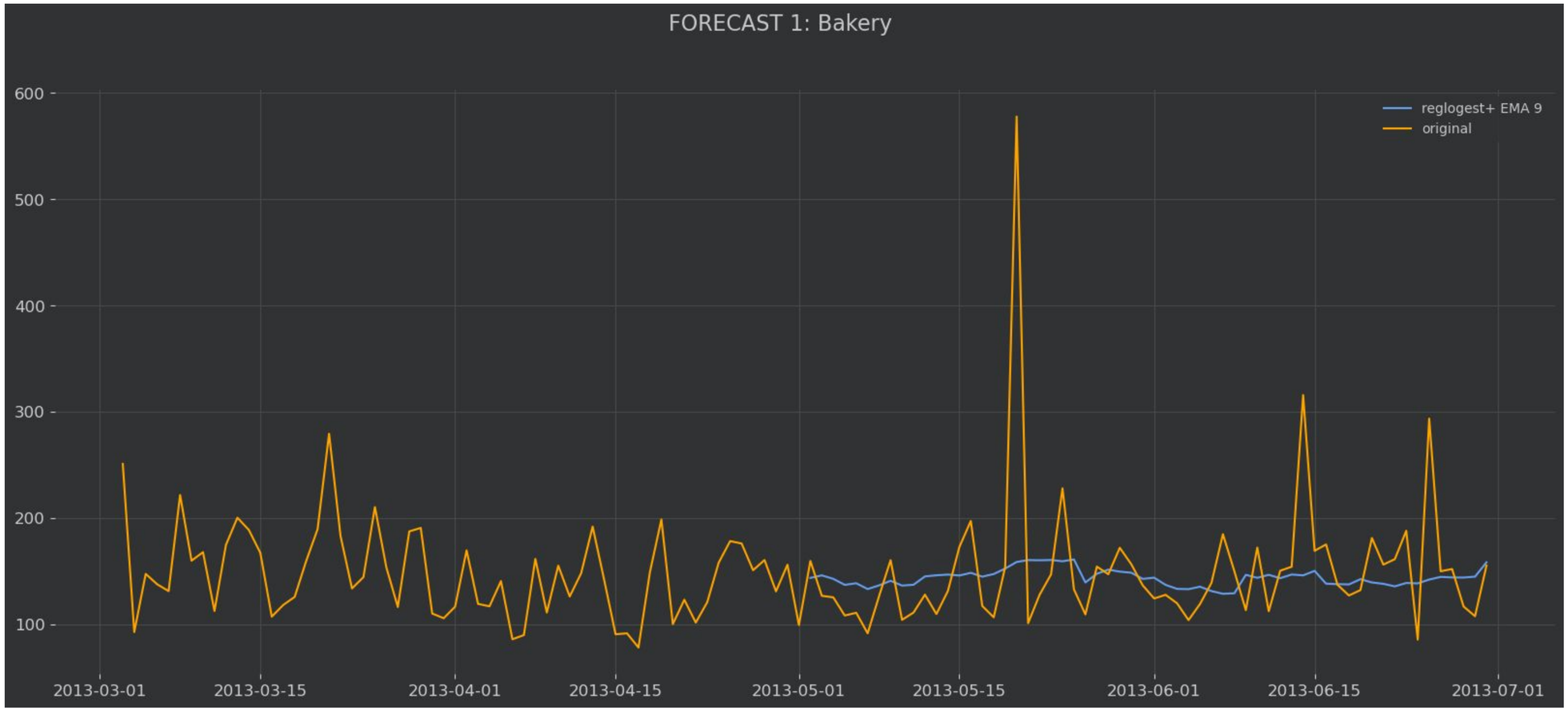
Forecast 1: Evaluación modelo

Recordando que en RMSE de Bakery obtuvimos 66,5 de score en el mejor de los modelos, nuestro modelo xgboost obtuvo RMSE 88, pero lo que hicimos fue sumar la proyección de la regresión, con la EMA optimizada del pronóstico xgboost:



Forecast 1: Evaluación modelo

El resultado fue que el promedio de ambas proyecciones daba un resultado mucho más favorable para ser considerado. El RMSE de reg_log_est(Bakery) + EMA(9) = 10.6796



Forecast 2

La complejidad de esta sección radica en el procesamiento iterado de todas las categorías de la muestra de forma automatizada.

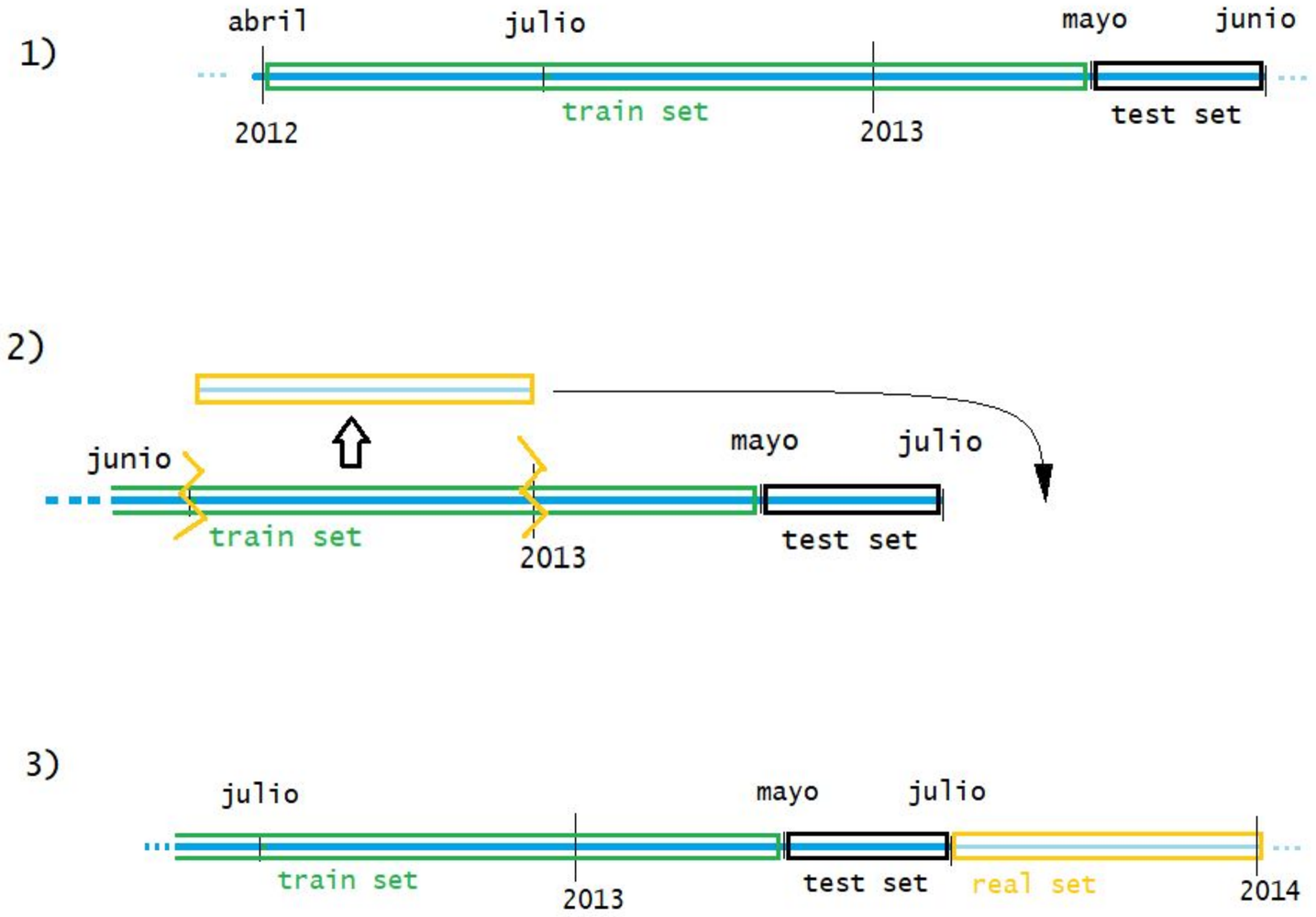
El procesamiento involucra los siguientes pasos:

- tratamiento index
- train test split y almacenamiento de datos para xgboost
- cross validation y almacenamiento de los datos para reg_log_est
- forecast xgboost, y estudio de estacionalidad, optimización del proceso de diferenciación
- desarrollo de modelos de reg_log_est y exportación de los modelos entrenados
- tratamiento de los datos y visualización

Vamos a definir:

1) Tratamiento datos
2) Primera proyección, y modelo de evaluación
3) Segunda proyección y exportación de modelos
4) Visualización de Pronósticos

Forecast 2: Tratamiento de datos



A continuación vamos a explicar el tratamiento de los índices de la serie temporal para modelizar datos a futuro que no tenemos.

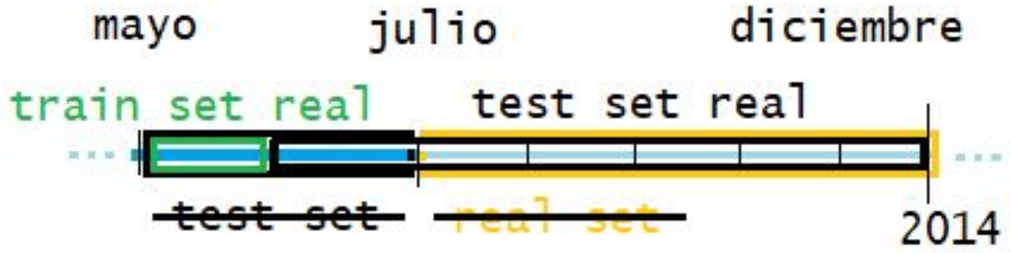
En (1) vemos un train test split normal, el mismo que utilizamos en FORECAST 1.

En (2) recopilamos registros históricos y manipulando los índices lo posicionamos al final de los datos que tenemos.

El resultado final se observa en (3)

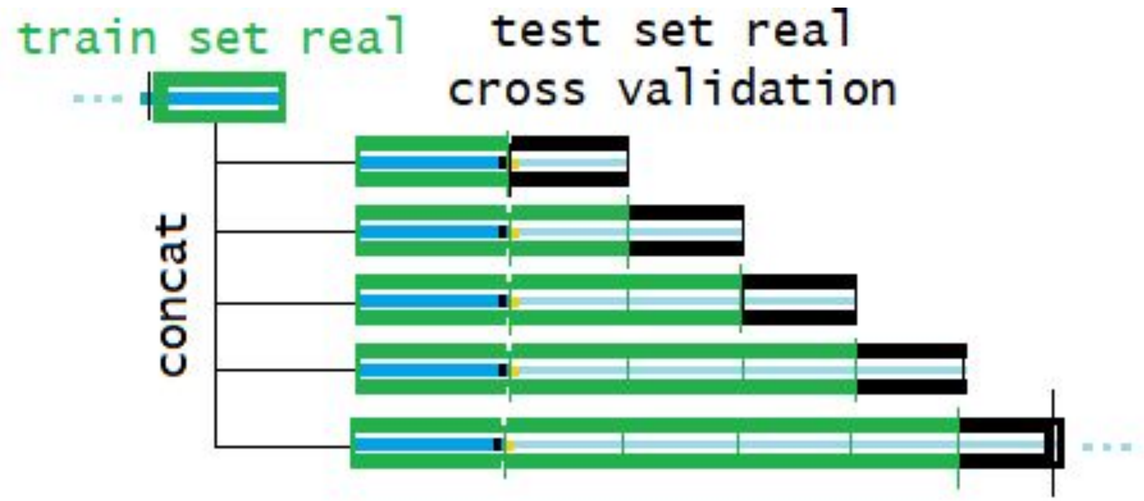
Forecast 2: Tratamiento de datos

4)



En (4) aplicamos nuevamente train test split. El tamaño del test es mayor al del train, porque será luego el set que haremos cross validation.

Una vez realizada la separación en los sucesivos train y test folds del cross validation, concatenamos el train set de (4) con los train set de cross validation.



Forecast 2: Primera proyección, y modelo de evaluación

APP_XGBOOST.py

Tiene la función
guardar_funcion()
Se descarga un documento pickle con la función "walk_foreward_validation()"

```
>> Bucle Forecast + x2 Diferenciacion Estandar - Grocery
        XGBOOST WALK-FOREWARD

100%|██████████| 184/184 [00:33<00:00,  5.56it/s]
```

>>prediccion

```
diferenciacion real + forecast
       DIFFERENTIATION  PROCESS
Selected Method: EMA
Optimized Params: 5
```

>>evaluacion

```
diferenciacion forecast
       DIFFERENTIATION  PROCESS
Selected Method: EMA
Optimized Params: 3
```

>>evaluacion

```
>> Bucle Forecast + x2 Diferenciacion Estandar - Pharmaceutical
        XGBOOST WALK-FOREWARD

100%|██████████| 184/184 [00:35<00:00,  5.12it/s]
```

>>prediccion

```
diferenciacion real + forecast
       DIFFERENTIATION  PROCESS
Selected Method: EMA
Optimized Params: 5
```

>>evaluacion

```
diferenciacion forecast
       DIFFERENTIATION  PROCESS
Selected Method: SMA
Optimized Params: 9
```

>>evaluacion

...

Forecast 2: Segunda proyección y exportación de modelos

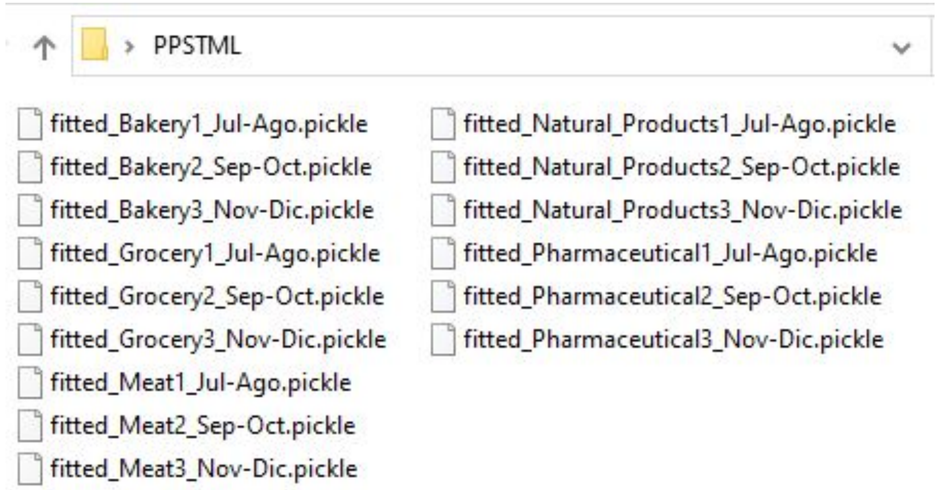
Como vimos anteriormente que aplicamos cross validation, es en el modelo de regresión lineal en donde se aplica.

En el modelo desarrollado modelamos los datos del segundo semestre y realizamos las siguientes divisiones bimensuales:

Año 2013

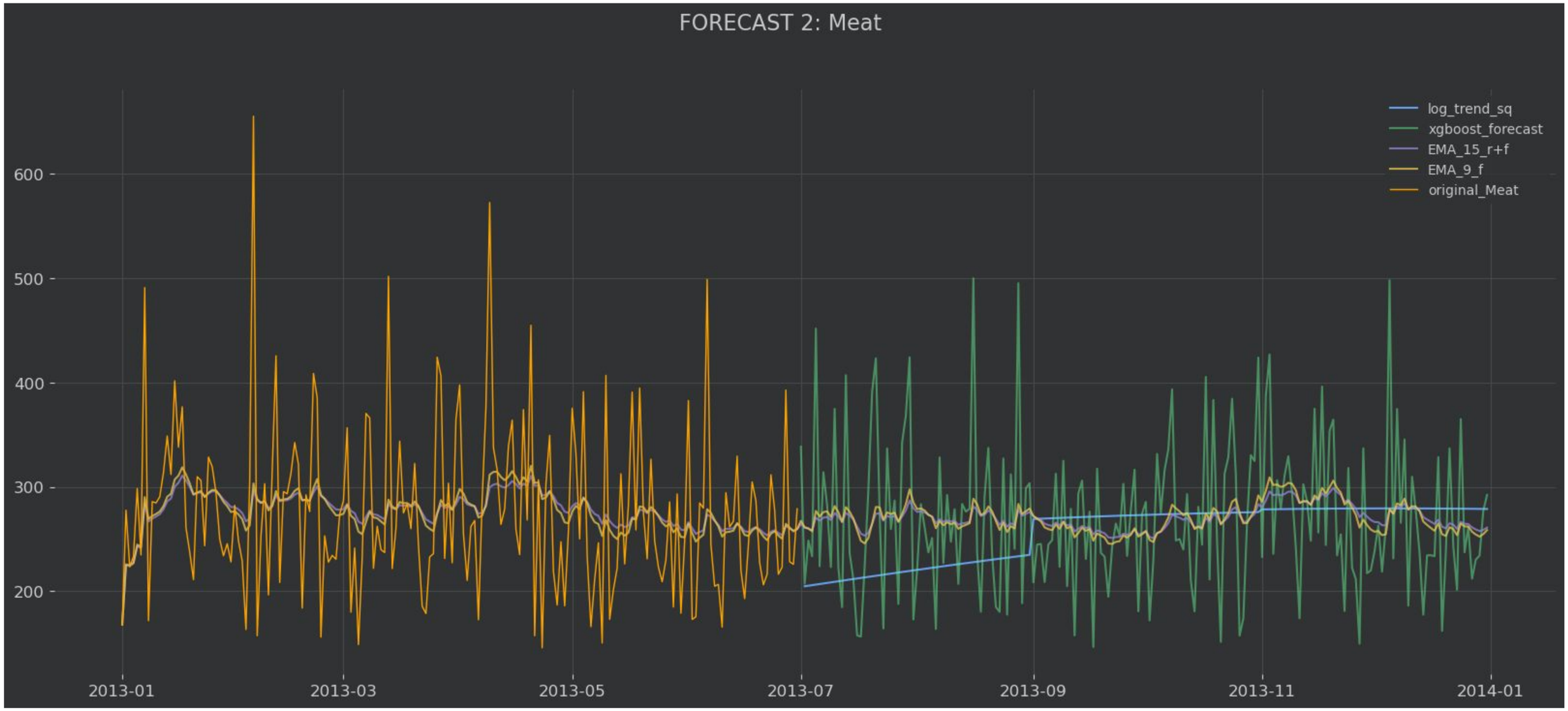
- split 1: train(Enero - Julio) y test(Julio - Septiembre)
- split 2: train(Enero- Septiembre) y test(Septiembre-Noviembre)
- split 3: train(Enero-Noviembre) y teat(Noviembre-Diciembre)

Las regresiones son en total 6 modelos que entrena hasta que selecciona el mejor, y tenemos 5 categorías. Por lo tanto 3 * 6 * 5, son 90 modelos que se instancian, entrenan y predicen, para lograr exportar solo los 15 que mejor pronóstico proponen.



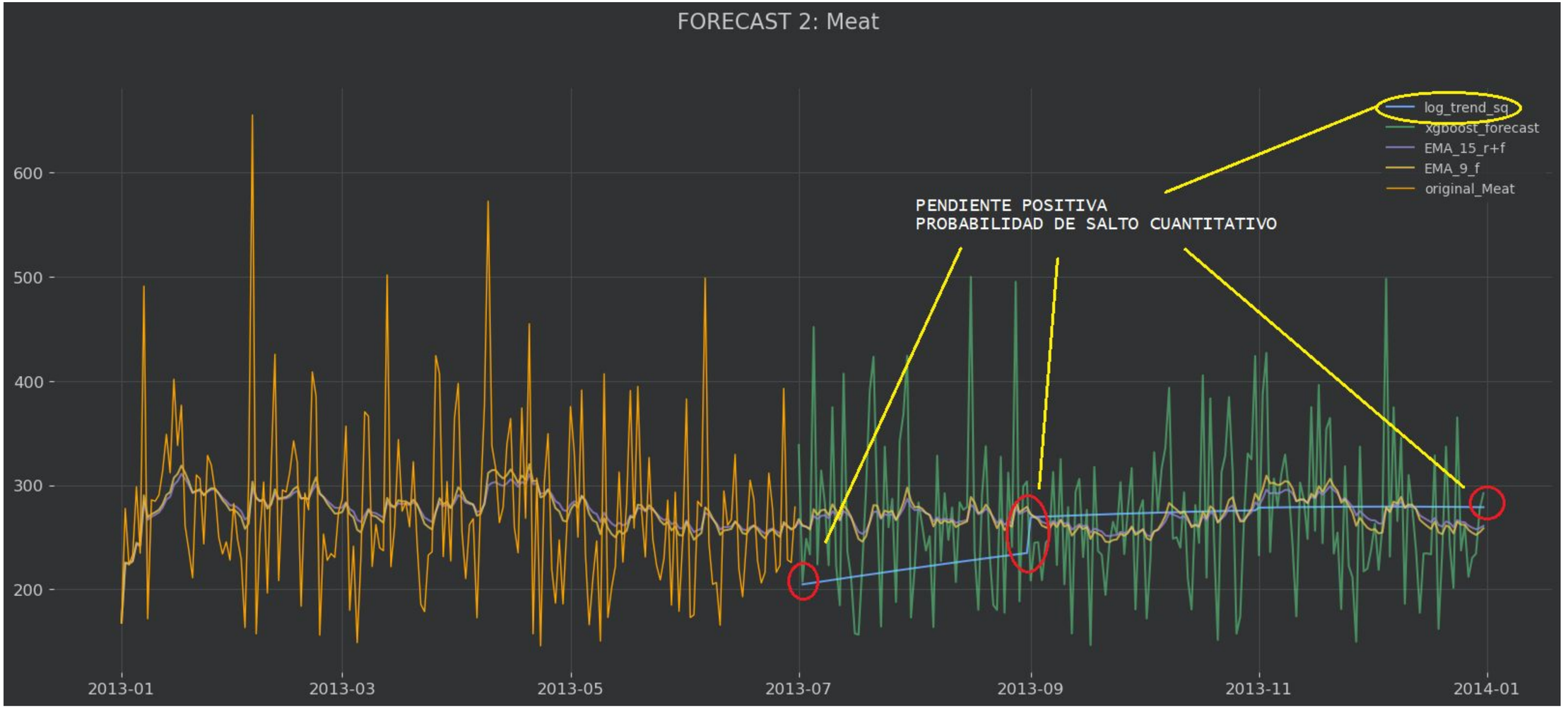
Forecast 2: Visualización de Pronósticos

A continuación vamos a mostrar algunos gráficos para mostrar cómo queda el formato final, y a modo de ejemplo analizaremos la información obtenida por los modelos de forecast.



CONCLUSION

Para demostrar el potencial de desarrollo e innovación que posee nuestro algoritmo vamos a compartir un breve análisis del gráfico visto anteriormente.



CONCLUSION

Podemos contrastar el resultado sobre una categoría si la comparamos con otra.