

Modelo predictivo de demanda para minimizar el desperdicio alimentario

Modelos Avanzados de análisis de datos II - Microproyecto 1

Universidad de los Andes. - Maestría en inteligencia analítica para la toma de decisiones

Néstor Fabián Cholo Acevedo, Jaime Orjuela Viracachá

Resumen

AgroApp es una aplicación diseñada por una start-up que tiene como objetivo conectar el campo con la ciudad de manera directa y sin intermediarios. De esta manera, los agricultores ponen a la disposición de la empresa sus distintos productos y la start-up se encarga de atender las necesidades de demanda distribuyendo los productos hasta los clientes finales en la ciudad. Para optimizar la gestión logística y desarrollar esta labor de manera eficiente, se plantea la siguiente **pregunta de negocio: ¿Qué cantidad de cada producto se debe solicitar para la siguiente semana?**, para lo cual se requiere diseñar e implementar un sistema de gestión y predicción que permita planear con anticipación la demanda esperada. Así se podrá coordinar las actividades de producción con los agricultores y atender la demanda mientras se minimizan las pérdidas por unidades de producto no vendido o estropeado.

Este es un reto altamente relevante para la sostenibilidad de la industria alimentaria, donde gran cantidad de las pérdidas se refieren a grupos alimenticios donde la frescura del producto es un importante criterio para el consumo. El desperdicio de los alimentos se relaciona con el sobre-pedido o el sobre-almacenamiento, lo cual puede a su vez ser consecuencia de dificultades en el pronóstico de la demanda. Este problema es aun más marcado a medida que se sube en la cadena de producción y se aleja de los patrones de comportamiento del consumidor final.

Por lo tanto, AgroApp debe gestionar los inventarios de manera inteligente, disminuyendo la ventana de tiempo entre cosecha y consumo. Para ello, necesita desarrollar modelos de predicción de demanda que apoyen su operación, logrando la disponibilidad de los productos para atender a los clientes del día siguiente, minimizando los desperdicios de alimentos e identificando hábitos de consumo que le permita optimizar sus operaciones logísticas y de negocio.

En el presente trabajo se presentan dos modelos predictivos de demanda para los productos más relevantes con los que trabaja AgroApp, obteniendo una predicción a nivel semanal (una semana adelante). La primera opción evaluada fue usando el paquete `fbprophet` y el segundo mediante la estimación haciendo uso de redes neuronales usando el paquete `keras` con el software `python`. Las dos alternativas muestran valores coherentes frente al histórico registrado, así como gráficas de tendencia que permiten al agricultor conocer estadísticas gráficas fáciles de entender y apropiadas para optimizar su producción.

Keywords: Analítica, modelos de demanda, aprendizaje computacional, toma de decisiones

1 Metodología

En la Figura 1 se muestra la metodología implementada, que inicia con el análisis exploratorio de los datos (productos). Se revisaron dos modelos aplicables a series de tiempo que se evaluaron y compararon, para finalizar con el análisis de resultados y conclusiones.



Figura 1. Metodología

Los modelos de series temporales permiten predecir valores futuros basándose en valores previamente observados, por lo cual se inicia el trabajo, haciendo el análisis exploratorio de los datos para determinar estacionalidad de la variable de total de ventas, así como determinar qué productos son los más adecuados para entrar en los análisis predictivos. Por otro lado, considerando que hay muchas técnicas para el estudio de las series temporales (ARIMA, ARCH, regressive models, neural networks por mencionar sólo algunas), se eligieron dos alternativas de

implementación en python: la primera es prophet, una herramienta desarrollada por Facebook que se liberó al público en 2017 (*Quick Start | Prophet*,) y la segunda mediante redes neuronales usando el paquete keras

1.1 Análisis de exploratorio

Teniendo en cuenta la premisa de minimizar el desperdicio alimentario de productos agrícolas nativos, se contemplaron las siguientes alternativas:

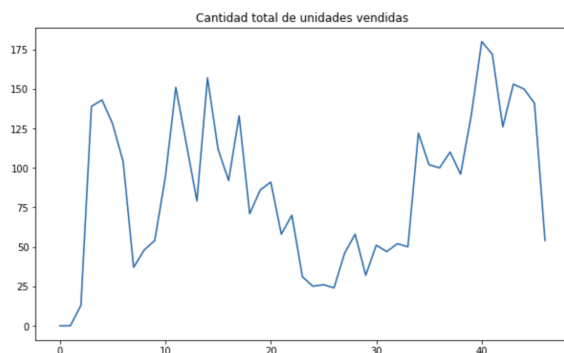
1. Categorizar los productos en tres grupos: Canasta Básica, Canasta Intermedia y Canasta Premium. La categorización se realizó mediante la opinión de expertos y teniendo como referencia *Sistema de información de precios (SIPSA)*,
2. Agrupar los productos por tipo: Frutas, Verduras, Tubérculos y Granos.
3. Calcular la cantidad de unidades semanales promedio vendidas para cada producto, se tomó como punto de decisión aquellos productos que al menos se venden 1 unidad semanal (aplicando redondeo a entero).

CATEGORIA	PRODUCTOS	%	CANASTA	PRODUCTOS	%	VENTAS	PRODUCTOS	%
FRU	56	39,11%	BASICA	35	39,11%	0	69	13,92%
GRA	4	1,21%	FRU	10	18,97%	FRU	36	8,09%
TUB	14	8,89%	GRA	2	0,06%	GRA	2	0,06%
VER	47	50,79%	TUB	7	5,94%	TUB	7	1,29%
Total	121	100,00%	VER	16	14,13%	VER	24	4,48%
			INTERMEDIA	32	21,86%	1	52	86,08%
			FRU	17	9,38%	FRU	20	31,03%
			GRA	2	1,15%	GRA	2	1,15%
			TUB	3	1,76%	TUB	7	7,60%
			VER	10	9,57%	VER	23	46,31%
			PREMIUM	54	39,03%	Total	121	100,00%
			FRU	29	10,76%			
			TUB	4	1,19%			
			VER	21	27,08%			
			Total	121	100,00%			

Figura 2. Distribución de productos

En la Figura 2 se muestran la distribución de productos para cada una de las alternativas evaluadas, en la primera opción si se toman frutas y verduras se tendrían 103 productos. En la segunda opción si se incluía la canasta básica y la premium, se tendrían 89 productos; mientras que la tercera opción resultaba una distribución más equilibrada considerando que los productos elegidos aportan el 86 % del volumen de ventas con el 43 % del inventario actual. Así las cosas, se generó un dataset agrupando las ventas semanales (filas) por producto (columnas) para todas las semanas posibles entre el 30 de octubre de 2016 y el 7 de septiembre de 2017, se encontró faltante de información únicamente para la semana 46 del año 2016, para un total de 47 semanas, 46 con información y una (1) faltante, a la que se le imputaron ventas cero (0).

Posteriormente se aplicó la prueba de Dickey Fuller para el valor total de ventas por semana, con lo cual se encontró que el conjunto agregado de datos es estacional para usarlo como input en el modelo de prophet. Ver figura 3b.



(a) Serie de tiempo ventas totales semana 1 a semana 47

```
from statsmodels.tsa.stattools import adfuller

result = adfuller(serie['Total'])
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %3f' % (key, value))
```

```
ADF Statistic: -3.228528
p-value: 0.018392
Critical Values:
1%: -3.581
5%: -2.927
10%: -2.602
```

(b) Dickey-Fuller test

Figura 3. Análisis de estacionalidad

2 Resultados y análisis resultados

2.1 fbprophet

Prophet es una herramienta de pronóstico basada en modelos aditivos donde las tendencias no lineales son ajustadas mediante un parámetro de estacionalidad que puede ser anual, mensual, **semanal** o diaria y que tiene la posibilidad de incluir efectos para días festivos. Esta herramienta funciona mejor con series de tiempo que tiene una alta estacionalidad, adicionalmente prophet es robusto a las presencia de datos anómalos y faltantes, así como a desplazamientos en la tendencia.

El análisis para cada producto se encuentra en el notebook adjunto a este documento. Se obtuvo una medida de bondad MSE de 1,486 como medida global del modelo. Para el producto de mayor rotación que es el tomate chonto (Figura 4a), se nota una tendencia creciente en ventas y se estima que para la última semana de septiembre se deben pedir 9 libras; mientras que para el producto de menor rotación que es la granadilla (Figura 4b) se nota una tendencia a la baja y se estima que para la última semana de septiembre no debería solicitarse producto al proveedor.

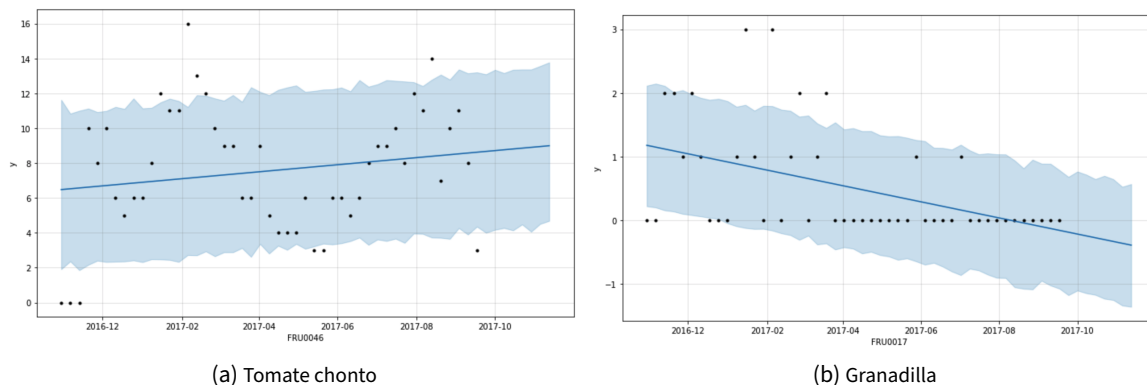


Figura 4. Análisis prophet para el producto de mayor y menor rotación.

2.2 Red Neuronal

Otra alternativa propuesta para el pronóstico de los productos de la segunda semana de septiembre, son las redes neuronales. Una arquitectura sencilla de red neuronal es la FeedForward (FF) mostrado en *Pronóstico de Series Temporales con Redes Neuronales en Python | Aprende Machine Learning*, la cual se implementó utilizando el paquete `keras`, para ello se tomó la serie de cada producto y se transpuso con el fin de pronosticar una semana considerando 24 semanas atrás. Así mismo se escalaron los valores al rango (-1,1). Se estructuró entonces una red neuronal de 24 neuronas a la entrada, una capa oculta con 24 neuronas y una capa de salida de 1 neurona. Como función de activación se utilizó tangente hiperbólica puesto que se utilizaron valores entre -1 y 1. Los valores obtenidos de la red, se reescalan para darle sentido a los resultados.

El análisis para cada producto se encuentra en el notebook adjunto a este documento. Para el producto de mayor rotación que es el tomate chonto (Figura 5a), se obtuvieron pronósticos de 5 libras para la última semana de septiembre y de 6 libras para las dos primeras semanas de octubre; mientras que para el producto de menor rotación que es la granadilla (Figura 5b) la red pronostica que este producto no debería solicitarse al proveedor para la última semana de septiembre.

Comparando las estimaciones de las metodologías propuestas de prophet y redes neuronales, se perciben valores cercanos de estimación en la mayoría de los productos. El valor de RME general para prophet resulta muy favorable para preferir esa metodología por sobre la red neuronal, sin embargo, la enorme flexibilidad de las redes neuronales permitiría probar una gran variedad de modelos ajustados a cada producto que por lo exhaustivo que resulta no fueron posibles de evaluar en el presente trabajo.

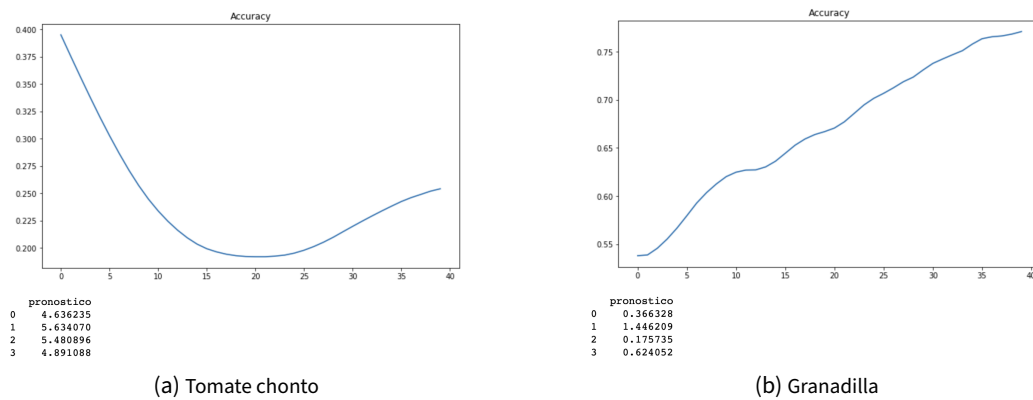


Figura 5. Pronóstico por red neuronal Feed Forward para el producto de mayor y menor rotación.

3 Conclusiones

1. Teniendo en cuenta la pregunta de negocio: ¿Qué cantidad de cada producto se debe solicitar para la próxima semana?, se implementaron dos metodologías de pronóstico, la primera mediante el paquete prophet y la segunda mediante redes neuronales usando el paquete keras.

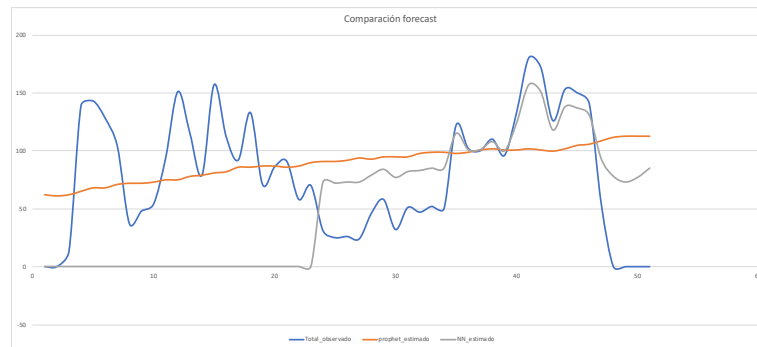


Figura 6. comparación de los valores observados versus los dos modelos propuestos: se pudo determinar que en general las dos metodologías responden correctamente a la tarea de estimación y con pronósticos muy cercanos, aunque la metodología prophet obtuvo mejores medidas de desempeño, mientras que para algunos productos las redes neuronales tuvieron precisión con tendencia descendente

Si bien la red neuronal ofrece mayor flexibilidad de parametrización, resulta más compleja de ajustar dado la cantidad y homogeneidad de productos. Por tanto en términos de implementación y escalabilidad, prophet resulta una alternativa más sencilla de implementar.

2. En la Figura 7 se muestra como el inventario de productos a solicitar disminuye de 121 productos inicialmente ofertados a 52 en el caso de la red neuronal y 43 en el caso de prophet. De acuerdo al registro histórico de lo vendido en la última semana, se presentará a la start-up dos escenarios: el conservador estimado a través de la red neuronal y el optimista estimado a través de prophet.

Fuente	Cantidad productos	Unidades a pedir	Ventas estimadas
Registro historico	33	54	\$101.000
Red Neuronal	52	78	\$172.000
Prophet	43	113	\$217.000

Figura 7. Resumen alternativas

Teniendo en cuenta el estado financiero de la start-up la decisión de cuál de las dos alternativas elegir, estará relacionada al perfil de tolerancia al riesgo y estrategias de crecimiento de la compañía.

3. Se propone entonces como alternativa de crecimiento, diseñar paquetes de productos de alta rotación combinados con los de baja rotación para estimular el consumo de los segundos.
4. Si bien en la fase de exploración de datos, se modeló la demanda total de productos por semana mediante un modelo ARIMA, se decidió no implementar esta metodología para los 52 productos escogidos ya que de acuerdo con la base de datos, los productos tienen un comportamiento de demanda distinto y esto hace dispendioso y poco eficiente el modelado de cada serie de manera individual.

Referencias

Pronóstico de Series Temporales con Redes Neuronales en Python | Aprende Machine Learning. Visitado 13 de octubre de 2020. <https://www.aprendemachinelearning.com/pronostico-de-series-temporales-con-redes-neuronales-en-python/>.

Quick Start | Prophet. Visitado 13 de octubre de 2020. https://facebook.github.io/prophet/docs/quick%7B%5C_%7Dstart.html.

Sistema de información de precios (SIPSA). Visitado 13 de octubre de 2020. <https://www.dane.gov.co/index.php/estadisticas-por-tema/agropecuario/sistema-de-informacion-de-precios-sipsa>.